OPEN
UNIVERSITY OF
CYPRUS
www.ouc.ac.cy

POSTGRADUATE PROGRAMME IN
INFORMATION AND COMMUNICATION SYSTEMS,
SCHOOL OF PURE AND APPLIED SCIENCES

STORY COMPREHENSION USING CROWDSOURCED COMMONSENSE KNOWLEDGE

DOCTOR OF PHILOSOPHY DISSERTATION

Christos T. Rodosthenous

2021

OPEN
UNIVERSITY OF
CYPRUS
www.ouc.ac.cy

POSTGRADUATE PROGRAMME IN
INFORMATION AND COMMUNICATION SYSTEMS,
SCHOOL OF PURE AND APPLIED SCIENCES

STORY COMPREHENSION USING CROWDSOURCED COMMONSENSE KNOWLEDGE

Christos T. Rodosthenous

A Dissertation Submitted to the Open University of Cyprus in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

April 2021

ISBN: 978-9963-695-72-0

VALIDATION PAGE

Doctoral Candidate: Christos T. Rodosthenous
Doctoral Thesis Title: STORY COMPREHENSION USING CROWDSOURCED COMMONSENSE KNOWLEDGE

The present doctoral dissertation was completed in the context of the Doctoral Programme in Information and Communication Systems at the School of Pure and Applied Sciences of the Open University of Cyprus and was successfully defended by the candidate on the 10th of March 2021.

Examination committee:

Chair of the examination committee: Professor Antonis Kakas, University of Cyprus

Supervisor: Associate Professor Loizos Michael, Open University of Cyprus

Committee member: Professor Alessandro Bozzon, Delft University of Technology

Committee member: Associate Professor Jahna Otterbacher, Open University of Cyprus

Committee member: Associate Professor Paolo Torroni, University of Bologna

Professor Antonis Kakas
Faculty of Pure and Applied Sciences
University of Cyprus

Associate Professor Loizos Michael
School of Pure and Applied Sciences
Open University of Cyprus

Chair signature: ....

Supervisor signature: ....

DECLARATION OF DOCTORAL CANDIDATE

The present doctoral dissertation was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy of the Open University of Cyprus. It is a product of original work of my own, unless otherwise stated through references, notes or any other statements.

Christos T. Rodosthenous
April 2021

....
Signature

ABSTRACT (Greek)

Η διατριβη επικεντρώνεται στο πρόβλημα της απόκτησης γνώσης κοινής λογικής και στην εφαρμογή της γνώσης αυτής στην αυτόματη κατανόηση ιστοριών από μηχανές. Την τελευταία δεκαετία ένας σημαντικός αριθμός από ερευνητές και ερευνητικά κέντρα έχουν εντατικοποιήσει τις προσπάθειες τους για συλλογή γνώσης κοινής λογικής, η οποία αποτελεί ένα σημαντικό συστατικό για τη δημιουργία "έξυπνων" μηχανών. Η προσεγγιση που ακολουθούμε εστιάζεται στο ότι η γνώση που είναι κατάλληλη για τη μηχανική κατανόηση ιστοριών μπορει να συλλεγεί μέσω μεθόδων πληθοπορισμού, χρησιμοποιώντας τόσο ενδογενείς όσο και εξωγενείς μεθόδους για συλλογή της γνώσης. Η προτεινόμενη μεθοδολογία έχει ως επίκεντρο τον διαχωρισμό αυτής της εργασίας σε μια σειρά πιο συγκεκριμένων εργασιών, οι οποίες επιτρέπουν στους συμμετέχοντες να εντοπίσουν τη σχετική γνώση, να τη μετατρέψουν σε μορφή που να μπορεί να διαβαστεί από μηχανές και να αξιολογήσουν την εφαρμοσιμότητα της γνώσης στην κατανοηση ιστοριών και πιο συγκεκριμένα στην απάντηση ερωτήσεων. Στην εργασία αυτή προτείνουμε μεθόδους για την απόκτηση και εφαρμογή γνώσης κοινής λογικής για τη μηχανική κατανόηση ιστοριών και χρησιμοποιούμε τεχνικές για την αναπαράσταση, απόκτηση και εξαγωγή συμπερασμάτων από τη χρήση γνώσης κοινής λογικής που έχουν καθιερωθεί από άλλους ερευνητές της περιοχής αυτής.

Αρχίζουμε με ανασκόπηση της βιβλιογραφίας, παρουσιάζοντας την παρούσα κατάστασης στις ερευνητικές περιοχές της μηχανικής κατανόησης ιστοριών, της συλλογής γνώσης κοινής λογικής και της εξέυρεσης κατάλληλων αναπαραστάσεων της γνώσης. Στη βιβλιογραφικη ανασκόπηση παρουσιάζουμε και έναν σημαντικό αριθμό συστημάτων που έχουν αναπτυχθεί τις τελευταίες δεκαετίες, δίνοντας έμφαση σε συστήματα που χρησιμοποιούν μεθόδους πληθοπορισμού για τη συλλογή γνώσης. Στο κείμενο παρουσιάζουμε και την έννοια της υπολογιστικής επιχειρηματολογίας που αποτελεί μια καλή επιλογή για αναπαράσταση της γνώσης.

Επιπρόσθετα, σε όλα τα εργαλεία που παρουσιάζονται σε αυτή τη διατριβή, χρησιμοποιούμε τους μηχανισμούς της υπολογιστικής επιχειρηματολογίας τόσο για αναπαράσταση της γνώσης όσο και για την εξαγωγή συμπερασμάτων. Αρχικά παρουσιάζουμε ένα εργαλείο που υποστηρίζει τους χρήστες στην κωδικοποίηση ιστοριών και στη χειρωνακτική

εισαγωγή κανόνων γνώσης σε μορφή που οι μηχανές μπορούν να διαβάσουν. Το εργαλείο αυτό είναι ένα διαδικτυακό ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που ονομάζεται "Web-STAR" και υποστηρίζει την κωδικοποίηση ιστοριών σε συμβολική μορφή και την εισαγωγή γνώσης κοινής λογικής τόσο από αρχάριους όσο και από εξειδικευμένους χρήστες. Επίσης, το "Web-STAR" παρέχει και μια σειρά από ενσωματωμένα σε αυτό εργαλεία για: 1) μετατροπή ιστοριών από φυσική γλώσσα σε συμβολική, 2) την προσθήκη γνώσης κοινής λογικής μέσω οπτικού εργαλείου που αναπαριστά τη γνώση σε κατευθυνόμενο γράφο, και 3) τη συνεργασία μεταξύ των χρηστών στην κωδικοποίηση ιστοριών. Το αποτέλεσμα της αυτοματοποιημένης διαδικασίας κατανόησης της ιστορίας παρουσιάζεται στους χρήστες τόσο σε μορφή κειμένου όσο και οπτικά σε μορφή χρονικής ροής, όπου οι χρήστες μπορούν να ακολουθήσουν το μοντέλο κατανόησης της ιστορίας και να εντοπίσουν τις διαφοροποιήσεις στη χρονική ροή. Το σύστημα έχει αξιολογηθεί για την ευχρηστία του τόσο από αρχάριους όσο και από εξειδικευμένους χρήστες ακολουθώντας μεθοδολογίες μέτρησης της εμπειρίας του χρήστη. Κατα την αξιολόγηση το σύστημα έλαβε υψηλό βαθμό ευχρηστίας.

Ακολούθως, παρουσιάζουμε ένα καινοτόμο πλαίσιο σχεδιασμού και υλοποίησης εφαρμογών πληθοπορισμού και την πλατφόρμα που δημιουργήσαμε για υλοποίηση των εφαρμογών αυτών (π.χ. παιχνίδια με σκοπό ή εφαρμογές για εκμάθηση γλωσσών) που μπορούν να χρησιμοποιηθούν για τη συλλογή γνώσης κοινής λογικής. Σχεδιάσαμε και εκτελέσαμε δύο πειράματα που εξετάζουν αν οι πλήρως αυτόματες πληθοποριστικές τεχνικές ή οι υβριδικές τεχνικές (αυτές που συνδιάζουν χειρωνακτικές, πληθοποριστικές και αυτόματες μεθόδους απόκτησης γνώσης κοινής λογικής) μπορούν να χρησιμοποιηθούν για την απόκτηση κατάλληλης γνώσης για αυτόματη κατανόηση ιστοριών. Η πρώτη εφαρμογή είναι ένα παιχνίδι με σκοπό, με την ονομασία "Knowledge Coder", που στηρίζεται μόνο σε μεθόδους πληθοπορισμού για την απόκτηση γνώσης από τους παίκτες. Η δευτερη εφαρμογή είναι και αυτή ένα παιχνίδι με σκοπό με την ονομασία "Robot Trainer". Το παιχνίδι αυτό σχεδιάστηκε με τρόπο που να χρησιμοποιεί υβριδική τεχνική για την απόκτηση γνώσης κοινής λογικής, τη γενίκευση της γνώσης και την αξιολόγηση της καταλληλότητας της για απάντηση ερωτήσεων σε ιστορίες που δεν είχε πρόσβαση το σύστημα προηγουμένως. Η γνώση που αποκτήθηκε δοκιμάστηκε για να απαντηθούν ερωτήσεις σε ιστορίες και τα αποτελέσματα δείχνουν ότι η γνώση αυτή είναι χρήσιμη για τον σκοπό αυτό, αφού μπορεί να εφαρμοστεί σε διάφορες γνωστικές περιοχές.

Στο επόμενο στάδιο της έρευνας, προσπαθούμε να αντιμετωπίσουμε το πρόβλημα του εντοπισμού της γεωγραφικής περιοχής που εστιάζει η κάθε ιστορία σε επίπεδο χώρας, δηλαδή τη γεωγραφική τοποθεσία με την οποία σχετίζεται η ιστορία αυτή. Για τον σκοπό αυτό έχουμε αναπτύξει μια εφαρμογή για να συμπεραίνει τη γεωγραφική περιοχή που εστιάζεται μια ιστορία χρησιμοποιώντας γνωσιακές βάσεις δεδομένων που έχουν γνώση που αποκτήθηκε με πληθοποριστικές μεθόδους. Η εφαρμογή αυτή χρησιμεύει στο να απαντήσει την ερώτηση του “πού” μια ιστορία λαμβάνει χώρα. Η εφαρμογή ονομάζεται “GeoMantis”, χρησιμοποιεί γνώση από γνωσιακές βάσεις δεδομένων όπως το ConceptNet και το YAGO και επιστρέφει μια πρόβλεψη για τη χώρα που εστιάζεται η ιστορία. Επίσης, η εφαρμογη περιλαμβάνει και μηχανισμό που επεκτείνει τις υπάρχουσες στρατηγικές που χρησιμοποιούνται με πληθοποριστικές μεθόδους στις οποίες το πλήθος αξιολογεί τη χρησιμότητα των επιχειρημάτων που υποστηρίζουν μια συγκεκριμένη χώρα.

Η διατριβή ολοκληρώνεται με συζήτηση των αποτελεσμάτων των πειραμάτων και της συμβολής των αποτελεσμάτων στην ερευνητική περιοχή της απόκτησης γνώσης κοινής λογικής αλλά και της εφαρμογής της στην αυτόματη κατανόηση ιστοριών.

ABSTRACT

This thesis examines the problem of commonsense knowledge acquisition and the application of this knowledge to automated story understanding. Lately, a number of researchers and institutions focused their efforts to gather commonsense knowledge as an essential component for developing "intelligent" machines. The approach taken is that knowledge appropriate for story understanding can be gathered by sourcing the task to the crowd, using both intrinsic and extrinsic methods for knowledge acquisition. The proposed methodology centers on breaking this task into a sequence of more specific tasks, so that human participants not only identify relevant knowledge, but also convert it into a machine-readable form and evaluate its applicability to story understanding tasks, such as question answering. We propose and investigate methods for the acquisition and application of commonsense knowledge, employing techniques for the representation, reasoning and retrieval of commonsense knowledge established by other researchers in the field.

The work in this thesis begins with the presentation of a literature review on the current state of affairs on automated story understanding, commonsense knowledge acquisition and appropriate representations of the acquired knowledge. A number of systems are presented, focusing on the ones that use human computation or crowdsourcing as a method for acquiring knowledge. The reader is also introduced to computational argumentation which is an appropriate substrate for representing knowledge. Argumentation semantics are used for representing knowledge and reasoning with it in the internal mechanisms of all the developed tools.

We present a tool for helping users to encode a story and to manually add knowledge rules in a way that machines can understand them. This tool is a Web-based Integrated Development Environment called "Web-STAR", that helps both expert and non-expert users in encoding stories in symbolic form and adding background knowledge. The tool also provides a number of embedded utilities for converting natural language stories to symbolic format, visually adding knowledge using a directed graph editor and promoting user collaboration. The output is presented both textually and graphically in a timeline format, where users can follow the comprehension model of a story and track changes in the story timeline. The IDE was evaluated for its ease of use both by expert and non-expert users, following user experience measurement methodologies and it received a high score in its evaluation.

Next we present a novel framework and platform we have developed for implementing crowdsourcing applications (e.g., Games with a Purpose or language learning applications) that can be used by human workers for gathering commonsense knowledge. We designed and executed two experiments that examine whether fully automated or hybrid crowdsourcing techniques, i.e., techniques that benefit from both manually, crowd-contributed and automatic acquisition of knowledge, can be used to gather commonsense knowledge. The first application, a Game With A Purpose (GWAP) called “Knowledge Coder” relied only on crowdsourcing approaches to acquire knowledge. The second application, again a GWAP called “Robot Trainer”, was designed using a hybrid methodology for gathering background knowledge, generalizing it and evaluating its appropriateness in answering questions on unseen stories. The acquired knowledge was tested on story comprehension tasks such as question answering and the results show that the gathered knowledge is useful in answering story questions on new unseen stories, since the gathered knowledge is applicable in different domains.

We also study the problem of inferring the geographic focus of a story at a country level, i.e., the geographic location that the story is related to. We developed an application for inferring the geographic focus of stories using crowdsourced knowledge bases, contributing in understanding the “Where” a story takes place type of question. This application, called “GeoMantis” retrieves knowledge from popular crowdsourced knowledge bases, such as ConceptNet and YAGO and returns a prediction of the country of focus. Furthermore, an expansion of this application was developed to apply a crowdsourced strategy for this task. Crowd-workers evaluated the usefulness of the arguments supporting a specific country on identifying the geographic focus of a document and the evaluated arguments were tested for identifying the geographic focus.

The thesis concludes with a discussion of the outcome of the conducted experiments on the Web-STAR IDE, the GWAPs for acquiring commonsense knowledge and the application of crowdsourced knowledge for geographic focus identification, highlighting the different contributions in the area of commonsense knowledge acquisition and its application in automated story understanding.

ACKNOWLEDGEMENTS

Firstly, I would like to give my sincere gratitude to my advisor, Associate Professor Loizos Michael. It was his guidance and support that led to this work. The ideas shared and discussions were valuable in the completion of this thesis. I appreciate everything I have learned from working with you.

I would like to thank the internal members of my doctoral committee, Professor Antonis Kakas and Associate Professor Jahna Otterbacher, for their interest in this work, their guidance, and feedback during the preparation stage and their support. Special thanks goes to the two external members of the examination committee Professor Alessandro Bozzon and Associate Professor Paolo Torroni who provided constructive feedback on this work.

I would also like to thank the Open University of Cyprus for providing me the opportunity to pursue a PhD in Artificial Intelligence and my colleagues at the Computational Cognition Lab for their support during these years. A huge thank you is owed to Elektra Kypridemou both for her help on parts of this work but also for supporting this project.

I am always grateful to my parents who have supported and encouraged me throughout my studies and my life. I hope that one day, I will be able to offer the same support they offered me to my children.

Last but not least, I am extremely thankful to my family, Georgia, Theodosis and Nikolas, who have tolerated me through this demanding process and have filled my life with happiness.

Part of the work reported in this thesis was supported by funding from the EU's Horizon 2020 Research and Innovation Programme under grant agreements no. 739578 and no. 823783, and from the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Policy.

I would like to dedicate this thesis to my loving wife Georgia and my children Theodosis and Nikolas.

Contents

List of Figures

List of Tables

1

Introduction

> *"Unless we can explain the mind in terms of things that have no thoughts or feelings of their own, we'll only have gone around in a circle."*
>
> – Marvin Minsky, *The Society of Mind (1987)*

Artificial Intelligence (AI) is "the science and engineering of making intelligent machines" (McCarthy, 1959). This is the foundation definition of AI from one of the fathers of the field, McCarthy. For a machine to be "intelligent", it needs to know what we humans know about our world and our surroundings, like "when the sun is up, it is daytime" or "a person needs to stand before he/she can walk". This type of knowledge is called commonsense knowledge and we humans have it naturally.

There is not just one universally acceptable definition of what commonsense knowledge is, but there are rather many that encompass many of its properties. In the work of McCarthy (1989) commonsense knowledge is described as knowledge which includes "the basic facts about events (including actions) and their effects, facts about knowledge and how it is obtained, facts about beliefs and desires. It also includes the basic facts about material objects and their properties". Zang et al. (2013) state that commonsense knowledge is "a tremendous amount and variety of knowledge of default assumptions about the world, which is shared by (possibly a group of) people and seems so fundamental and obvious that it usually does not explicitly appear in people's communications". Hung et al. (2010) state that "commonsense knowledge refers to beliefs or propositions that appear to be obvious to most people, without dependence on any specific esoteric knowledge". These beliefs, don't necessarily need to be true, but rather accepted by a group of people. Michael (2008) presents commonsense knowledge as "anecdotal knowledge that people accumulate through experience, rules of thumb, beliefs that are assumed to be shared by a group of people, statements about the world that are not necessarily always true but that hold sufficiently often so as to make their adoption useful.”. All the above definitions present a clear understanding of the properties commonsense knowledge should hold and emphasize the fact that this knowledge is not always found to be true. In this work, we embrace the view of Michael (2008) for what commonsense knowledge is.

At this point, a curious mind would ask, how much commonsense knowledge does a human have? The amount of knowledge needed for performing commonsense reasoning tasks was investigated and was found that a typical mature person exhibits a functional learned memory content of around a billion bits (Landauer, 1986), based on the rates of learning and forgetting. In the work of Mueller (2006a) a number of estimates of the amount of commonsense knowledge a person has, are depicted along with the different approaches that were used (Moravec, 2000; Turing, 1950). The outcome stresses the fact that it is difficult to have an accurate measurement of the amount of knowledge a mature person has, since we are not able to explain how the human brain actually works.

To develop machines that are able to comprehend, researchers tried to create large repositories of knowledge which can be used for reasoning. There are many examples and projects for building repositories of commonsense knowledge and delivering it to interested parties. The last decades were characterized by the slow progress of science on acquiring and applying commonsense knowledge, concern also shared by Davis and Marcus (2015). Currently, there is an increasing interest on the area of commonsense knowledge bases. The Allen Institute for Artificial Intelligence (AI2) has launched a very promising project (Allen, 2018), which aims at forming a unified knowledge base using knowledge originating from the Institute’s projects (e.g., machine reading and reasoning, natural language and understanding, and computer vision). Based on the institute’s strategy, a number of sub projects were initiated, such as a dataset to test whether or not a machine has common sense (Zellers et al., 2018b), a knowledge graph of everyday commonsense reasoning (Sap et al., 2019), and visual commonsense reasoning (Zellers et al., 2018a).

Another area of interest for AI is that of CHI (Computer-Human Interaction). In the work of Lieberman (2008) the importance of commonsense knowledge in usable AI is stressed out, and more specifically in assuring the adherence of AI interfaces to CHI principles for usable interfaces. This lead to the rise of chatbots and smart assistants, such as Alexa, Siri and Cortana.

There are many examples of machines which are able to exhibit human-like intelligence, such as the IBM Watson system (High, 2012) that managed to win against two of Jeopardy’s greatest champions (Ferrucci et al., 2013), the AlphaGo which is the first computer program to defeat a professional human player (Silver et al., 2016) in the popular Chinese game Go, the AlphaGo Zero (Silver et al., 2017) which is the latest version of the AlphaGo that not only is able to win all human players of Go, but is also capable to win all previous versions of the AlphaGo. Moreover, it is able to learn by simply playing games against itself, starting from completely random play.

Do all of the above achievements give a solution to the hard problems of AI? Even though all these algorithms and systems are able to exhibit some sort of human-like intelligence, even higher in some cases as far as specific tasks are encountered, still none of them is applicable on general tasks. Moreover, these algorithms and systems raise concerns in terms of *accountability*, *responsibility* and *transparency* (Dignum, 2017) and some times they fire the debate on whether humans should rely on such algorithms for their daily tasks without knowing how these systems work and the explainability of their results.

At this point, it is important to highlight the significant role of stories in human thinking. That role was referenced in the work of many researchers, such as Schank (1972), Schank and Abelson (1977), Schank and Riesbeck (1981), and Winston (2012a), in all of which, stories have a central role to play in human thinking. Furthermore, Winston (2012b) argues that "story understanding is the centrally important foundation for all human thinking" and gives a number of examples of stories used in our daily lives, such as fairy tales, history lessons, literature, religious texts, recipes, and specific cases examined in science fields, such as law, medicine, and business. Machines that are able to understand stories have applications in many "intelligent" systems, such as advisory, dialogue, filtering, information retrieval, question answering, and summarization systems (Mueller, 2004). Michael (2013b) argues that "it is natural and desirable to investigate how to build machines that understand stories, both as a means to understand humans themselves, but also as a way to improve human-machine interactions".

1.1 A Short Historical Background on AI

The human envisioning of machines with "Intelligence" goes back to ancient times with stories coming from Greek mythology. The Greek robot of TALOS, which is a creation of Hephaistos, the Greek god of metallurgy, is one of the first mentions of human-shaped machines capable to perform human-like activities (e.g., guarding an island from strangers). Of course, there are other mentions of machines who can "think" (McCorduck, 2004) from Chinese and Hebrew stories.

In the 50's, Alan Turing presented a theory of computation (Turing, 1950) which suggested that a digital machine, i.e., a machine that uses the binary digits of 1 and 0, could simulate human thinking. Turing proposed a definition of "intelligence", by describing a game where a human participant could not distinguish between responses from a machine and a human. A machine that is able to pass such a test could be considered “intelligent”. Even though the Turing test received a good amount of criticism, it is considered as one of the first attempts to formalize machine “intelligence” and introduced the research field of AI.

The actual term **Artificial Intelligence** first appeared at the Dartmouth Summer Research Project on Artificial Intelligence in 1955. This was the seminal event of the AI research field, where the participants Allen Newell (CMU), Herbert Simon (CMU), John McCarthy (MIT), Marvin Minsky (MIT) and Arthur Samuel (IBM) worked on the following proposal:

> “We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.” (McCharty et al., 1955)

In this proposal, notions such as natural language processing, neural networks, theory of computation, abstraction and creativity were discussed and are still valid until today. From that point on, the participants of the workshop and their students were dedicated with addressing the various research problems AI brought, having the ultimate goal of creating machines with human-like intelligence.

Research on AI was concentrated on the design and development of expert systems that were able to perform question answering or summarization on specific domains, such as restaurants, terrorism, economics etc. It was not far after, that researchers understood that in order to have machines that are able to have human-like understanding a huge amount of knowledge about the world is needed, as humans acquire this knowledge since they are born. Moreover, for machines to be able to use our language of communication, it became clear that a way to represent this world knowledge along with a way to represent natural language was needed.

1.1.1 Knowledge Representation and Reasoning

Natural Language is too complex, irregular, diverse and includes a number of philosophical problems of meaning and context (Iwanska and Shapiro, 2000), hence it is very difficult for machines to use it. Since the early days of AI, the need for formal methods to represent knowledge and encode natural language to a more structured form was identified. In the work of McCarthy (1959) logic is used to manually encode commonsense knowledge. Logic-based formalisms, such as First-Order logic (Smullyan, 1968), description logics (Baader et al., 2003), event calculus (Kowalski and Sergot, 1989; Miller and Shanahan, 2002) and situation calculus (Reiter, 1991) where investigated along with more loose representations such as frames and semantic networks which lack formal (logic-based) semantics.

Besides logic-based representations, other approaches were investigated such as neural networks (Haykin, 1999) and Bayesian networks (Pearl, 1988). Representing knowledge using neural networks is inspired by the inception that the human brain works in a very different way than a machine (e.g., a digital computer). This inception recognizes that the brain is a highly complex parallel computer capable to structure its components (neurons) to perform very fast computations (Haykin, 2009). This approach for representing knowledge has a number of limitations since it moves away from the descriptive nature of logic-based representations and makes it very difficult for humans to track the reasoning process.

The way humans reason is also studied in psychology terms. More specifically, the area of Psychology of Reasoning (Wason and Johnson-Laird, 1972), i.e., the study of how people draw conclusions to solve problems and make decisions, presents evidence suggesting that human reasoning is not following strict mathematical or classical logic views when it comes to decision making. Evans (2002) suggests that only few reasoning researchers still follow the line that logic is an appropriate system for the way humans reason.

A more appropriate substrate for human-like reasoning is argumentation (Dung, 1995). Argumentation deals with supporting a certain claim (e.g., a belief) based on some premises and an argument that connects these premises to the claim. A number of researchers (Kakas, 2019; Kakas et al., 2016; Strass et al., 2019) adopt this approach as it is more close to the way humans reason especially when facing conflicting information. In this work, we embrace this view and we employ argumentation both for representing knowledge and for reasoning.

1.1.2 Knowledge Acquisition

From the early days of AI, researchers encountered the problem of acquiring commonsense knowledge to feed their programs and systems. There are three ways to acquire commonsense knowledge: 1) manual encoding of knowledge, 2) automatic extraction and encoding of knowledge from text and 3) crowdsourced knowledge where the task is sourced to the crowd.

The first (manual encoding) is the most difficult approach, since human experts need to encode knowledge in the form of rules or facts using an appropriate representation, a task that takes a lot of time and effort to be completed. In fact, Cyc (Lenat, 1995) is an example of such a project, which is still running since 1984 and researchers are encoding knowledge in symbolic form, aiming to build a knowledge base which will be used in future generations of expert systems.

The second approach (automatic extraction), uses large text corpora available either at the web (unstructured text) or from pre-annotated text, e.g., the Brown Corpus (Kucera and Francis, 1979). A number of natural language processing methods are applied to extract relevant knowledge and represent it to an appropriate format. Examples of such projects include the KNEXT project (Schubert, 2002), the LORE project (Gordon and Schubert, 2011), the Never Ending Language Learner (NELL) (Mitchell et al., 2015), etc. These projects are presented in Chapter 2.

The third approach (crowdsourcing) is the contribution of knowledge from the crowd. This is the case where untrained people contribute knowledge either by directly writing it in natural language or using an implicit crowdsourcing approach, such as a Game With A Purpose (von Ahn and Dabbish, 2008) where contributors play a game and while having fun they contribute knowledge. There are also other type of motives such as monetary rewards (Buhrmester et al., 2011a) or more altruistic ones, e.g., submitting knowledge for scientific purposes, which can be used for knowledge acquisition as well.

We add another distinction of knowledge acquisition methods, similar to that of crowdsourcing, where **implicit** knowledge acquisition relies on knowledge contributors who don't know that they are actually contributing knowledge but they do it as a side task of another process and **explicit** knowledge acquisition where contributors are aware of the task of contributing knowledge.

1.2 Research Problem

In this work, we embark on a journey to address the research problem of the acquisition of commonsense knowledge and its application in an automated story comprehension system. Singh et al. (2004) described the commonsense reasoning problem as one of the most challenging in the field of AI, with many real life applications. Winston (2011) notes that "A team of dedicated first-class engineers can build systems that defeat skilled adults at chess and Jeopardy, but no one can build a system that exhibits the commonsense of a child.". This is the current state of affairs until today for this important problem.

To address this problem, we first need to attack other individual "smaller" problems, such as finding:

- Approaches to acquire large amounts of knowledge

- Suitable representations for the acquired commonsense knowledge
- Methods to reason with the acquired knowledge
- Methods to retrieve suitable knowledge for comprehending
- Methods to apply the acquired commonsense knowledge

In this thesis, we propose methods for the acquisition and application of commonsense knowledge and we employ techniques for the representation, reasoning and retrieval of commonsense knowledge established by other researchers in the field.

Current methods and systems developed are either too slow in acquiring knowledge or too costly. For example the Cyc project has been running since 1984 and until today has managed to gather approximately 1.5 million general concepts and 20 million general rules and assertions involving those concepts. More than 2000 PhD scientist-years have been spend for acquiring these data. Moreover, a large part of this knowledge is "closed" under a proprietary license and only a small part is available through other knowledge bases and can be used for research.

Furthermore, approaches which rely entirely on human computation power, such as crowdsourcing are more scalable but often lead to low quality contributions and are prone to errors. Such approaches mostly acquire knowledge in natural language, which requires more steps to make it machine readable and hence applicable for machine comprehension. On the other hand, strict symbolic logic approaches were tested since the 70s using expert-systems and the majority of the developed systems were abandoned, since it was impossible for users outside of the specific community to uptake the system and use it on their particular paradigm.

Methods which crawl the web or other large corpora are able to automatically extract large amounts of data, but the problem with these approaches is the noise and sometimes misleading data, which result in low quality of learned facts. The Never Ending Language Learner project is an example of such a system which is running 24/7 since 2010 and has managed to gather 50 million beliefs. The system reports a high confidence for only 5% of them.

Machine learning approaches, even though they are not new (first appeared in the 60s), are on the hype after advances in computational power, storage and big data for training them. These approaches perform well when focused on a specific task and trained on large training sets, but fall behind when there is not a large volume of data for training. Moreover, it is almost impossible to explain why a certain conclusion was reached and this is the main reason for criticism and concerns expressed on the inability of these methods to explain their results.

In this thesis we propose user-friendly systems for acquiring knowledge by combining crowdsourcing techniques with knowledge engineering and automated methods to obtain commonsense knowledge suitable for automated story understanding. More specifically we will try to answer the following research questions:

- What is an appropriate representation for commonsense knowledge, and more specifically, how can formal argumentation methods be used for representing and reasoning with commonsense knowledge?
- What type of interfaces can we use to acquire commonsense knowledge from humans?
- How can we evaluate the acquired knowledge in the context of story understanding to demonstrate the usefulness of the acquired knowledge?

1.3 Why is the Problem Interesting

The same research problem puzzled scientists from the early days of AI back in the '70s. Building machines that can understand stories and natural language text in general, requires knowledge that is not always explicitly present in the story text, but it is inferred. One would not expect to find an explicit mention of the text "It is daytime because the sun is up". A large amount of human knowledge, experiences and history is present in textbooks and articles in the web or in print and forms a good source.

A machine that will be able to understand text in natural language is one of the major goals of Artificial Intelligence. Such a machine will be able to exhibit human-like intelligence. Contributions in this research area have a direct impact on machine translation, information retrieval, relation and event extraction, and text summarization.

The importance of the problem and research community's interest on it, is also highlighted by the large amount of conferences on the topic which hold special tracks and workshops. Also, many research organizations are spending a great amount of their budget in research regarding commonsense knowledge acquisition and applicability in many modern tasks.

1.4 Thesis and Contribution

This thesis is focused on the problem of commonsense knowledge acquisition from humans and its application in story understanding. During the last decades, a number of researchers and institutions focused their efforts on gathering commonsense knowledge leading to a revitalizing interest in this area.

We investigate the approach that knowledge appropriate for story understanding can be gathered by using graphical interfaces and games, and by sourcing the task to the crowd or by combining other methods with crowdsourcing. The proposed methodology centers on breaking this task into a sequence of more specific tasks, so that human participants not only identify relevant knowledge, but also convert it into a logic-based format (suitable for automated story understanding) and evaluate its appropriateness.

Furthermore, we demonstrate how argumentation (Baroni et al., 2011; Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008; Dung, 1995) can be used as an appropriate substrate for the development of automated systems that interact with humans (Kakas and Michael, 2016; Michael, 2017, 2019). Argumentation semantics are used for both commonsense knowledge representation and for reasoning using the STory comprehension through ARgumentation (STAR) system (Diakidoy et al., 2015), that supports revision of the comprehension model as new premises are presented to the reader, question answering and knowledge representation in the form of causation, implication and preclusion rules.

We also propose methods for using and evaluating the acquired knowledge on story understanding tasks such as question answering, and especially for questions where their answers are not explicitly found in the story text, but are inferred.

1.4.1 Requirements and Design Considerations

In this doctoral work, we present methods and systems that span the ways in which users can contribute knowledge. For addressing the research questions set for this thesis, we designed systems and graphical interfaces taking under consideration a number of requirements both for the design and their functionality.

For the graphical interfaces, the following high-level requirements and design considerations are set for the developed systems: 1) Should be web-based, 2) Should be accessible by any device, 3) Should be easy to use by non-expert users, 4) Should provide help and guidance to the users for performing their tasks, 5) Should handle input both in natural language and in symbolic language, 6) Should provide easy to understand representations, 7) Should provide mechanisms for automatically evaluating the acquired knowledge, 8) Should present to the user the decision process followed internally for coming to a certain outcome.

In terms of engineering perspective, the developed systems should follow a number of technical design considerations, such as: 1) storing acquired knowledge in a structured format that can be retrieved and processed by the interfaces, e.g., relational databases, 2) exposing their internal functionality through webservices, 3) providing user authentication mechanisms compatible with third-party services, 4) providing a modular architecture for easy expansion, and 5) should be based on open source libraries which do not require licensing costs and hence the systems can be reused by other researchers in the field without the constraint of acquiring costly licenses.

Furthermore, the representation used for the acquired knowledge should be able to support automated story understanding systems and should also support the requirement for systems that are able to explain their behaviour to users.

The proposed methods and systems are evaluated on their ease of use by non-expert users and their ability to answer questions using the acquired knowledge on unseen stories where the answer of the question is not explicitly found in the story text. In the next chapters we provide detailed requirements and evaluation metrics for each of the developed system and method.

1.4.2 Handcrafted Preparation of Knowledge

The first contribution is a visual tool for facilitating users to encode a story and to manually add knowledge rules in a way that machines can understand them. This tool is a web-based integrated development environment (IDE) called **Web-STAR**, that facilitates both expert and non-expert users in encoding stories in symbolic form and adding background knowledge (Rodosthenous and Michael, 2018c). The IDE is built on top of the STAR system (Diakidoy et al., 2015) and provides a number of tools for converting natural language stories to symbolic format, visually adding knowledge using a directed graph editor and collaboration functionality. The output is presented both textually and graphically in a timeline format, where users can follow the comprehension model and track changes to the story timeline as the story unfolds.

The IDE was evaluated for its ease of use by both expert and non-expert users, following user experience measurement methodologies and it received a high score in its evaluation. The IDE is currently used in both a classroom setup and by individual users interested in story understanding.

1.4.3 Crowdsourced Acquisition of Knowledge

Following the first contribution, a more scalable methodology for acquiring knowledge is needed than that of relying only on manually encoding of knowledge rules. This result guided us in the direction of using crowdsourcing with appropriate motives and guidance towards the suitable format needed for further usage of the acquired knowledge. Aiming in that direction, we developed a novel framework and platform (Rodosthenous and Michael, 2018a) for the development of crowdsourcing applications (e.g., Games With a Purpose or language learning applications) that can be used by untrained contributors for gathering commonsense knowledge. We designed and executed two experiments, that examine whether fully or hybrid crowdsourcing techniques, i.e., techniques that benefit from both manually, crowd-contributed and automatic acquisition of knowledge, can be used to gather commonsense knowledge.

Two Games With a Purpose were developed for conducting the experiments: “**Knowledge Coder**” (Rodosthenous and Michael, 2014) and **“Robot Trainer”** (Rodosthenous and Michael, 2016). The first relies only on crowdsourcing approaches to acquire knowledge. The latter uses a hybrid methodology for gathering background knowledge, generalizing it and evaluating its appropriateness in answering questions on unseen stories. The acquired knowledge was tested on story comprehension tasks, such as question answering and the results show that the gathered knowledge is useful in answering story questions on new unseen stories, since the gathered knowledge is applicable in different domains.

1.4.4 Application of Commonsense Knowledge

The third contribution of this thesis is the application of crowdsourced knowledge on inferring the geographic focus of news-stories at a country level where the country of focus is not explicitly mentioned in the story text (Rodosthenous and Michael, 2019). The geographic focus of a story can be defined as the geographic location that the story is related to. For example, the text snippet:

> “*A letter to creditors says Mr Tsipras is prepared to accept most conditions that were on the table before talks collapsed and he called a referendum . . .* ”[1]

is focused on Greece, even though the country is not explicitly mentioned in the text. We developed an application for inferring the geographic focus of stories using crowdsourced knowledge bases, contributing in understanding the “Where” a story takes place type of question. The application, called **“GeoMantis”** (Rodosthenous and Michael, 2018b) retrieves knowledge from popular crowdsourced knowledge bases, such as ConceptNet (Speer and Havasi, 2013) and YAGO (Hoffart et al., 2011; Suchanek et al., 2007, 2008) and returns a prediction of the country of focus. Furthermore, an expansion of the application was developed to apply a crowdsourced strategy for this task (Rodosthenous and Michael, 2021). Crowd-workers evaluated the usefulness of the arguments on identifying the geographic focus of a document and the evaluated arguments were tested for identifying the geographic focus.

[1] http://www.bbc.com/news/

1.4.5 Research Outcome

The outcome of this research was presented in several international fora, peer-reviewed conference proceedings, and high impact journals. Parts of this thesis (ideas, figures, results, sections, chapters and discussions) have appeared previously in the following publications and are also part of this work:

(1) Christos T. Rodosthenous and Loizos Michael. A Crowdsourcing Methodology for Improved Geographic Focus Identification of News-Stories. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pages 680-687, 2021.

(2) Christos T. Rodosthenous. Understanding Stories Using Crowdsourced Commonsense Knowledge. Online Handbook of Argumentation for AI, Volume 1, pp. 27–32, 2020.

(3) Christos T. Rodosthenous and Loizos Michael. A Platform for Commonsense Knowledge Acquisition Using Crowdsourcing. In Katerina Zdravkova, Karën Fort, and Branislav Bédi. Supplementary Proceedings of the enetCollect WG3 & WG5 Meeting 2018, pages 24–25, Leiden, Netherlands, 2018. CEUR.

(4) Christos T. Rodosthenous and Loizos Michael. Web-STAR: A Visual Web-based IDE for a Story Comprehension System. Theory and Practice of Logic Programming, 19(2):317–359, 2019.

(5) Christos Rodosthenous and Loizos Michael. Using Generic Ontologies to Infer the Geographic Focus of Text. In Jaap van den Herik and Ana Paula Rocha. Agents and Artificial Intelligence, pages 223–246, Cham, 2019. Springer International Publishing.

(6) Christos T. Rodosthenous and Loizos Michael. GeoMantis: Inferring the Geographic Focus of Text using Knowledge Bases. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pages 111–121, Madeira, Portugal, 2018. SciTePress.

(7) Christos T. Rodosthenous and Loizos Michael. Inferring the Geographic Focus of Stories Using Crowdsourced Knowledge Bases. Presented at the 1st International Workshop on Cognition and Artificial Intelligence for Human-Centred Design (CAID 2017), Melbourne, Australia, 2017.

(8) Christos T. Rodosthenous and Loizos Michael. Web-STAR: Towards a Visual Web-Based IDE for a Story Comprehension System. In Proceedings of the 2nd International

Workshop on User-Oriented Logic Paradigms (IULP2017), Espoo, Finland, 2017. arXiv.

(9) Christos T. Rodosthenous and Loizos Michael. A Hybrid Approach to Commonsense Knowledge Acquisition. In Proceedings of the 8th European Starting AI Researcher Symposium (STAIRS 2016), pages 111–122, Hague, Netherlands, 2016. IOS Press.

(10) Christos T. Rodosthenous and Loizos Michael. Gathering Background Knowledge for Story Understanding through Crowdsourcing. In Proceedings of the 5th Workshop on Computational Models of Narrative (CMN 2014), volume 41, pages 154–163, Quebec, Canada, 2014. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

(11) Christos T. Rodosthenous and Loizos Michael. Steps Towards Building a Story Understanding Engine. Poster presented at the Doctoral Consortium of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014), Vienna, Austria, 2014.

Additionally to the aforementioned, two datasets were created as an outcome of this thesis. Firstly, a commonsense knowledge rules dataset[2] which includes more than 1500 commonsense verified knowledge rules and preferences between them, all verified using crowd-contributors. Secondly, a crowd-verified dataset[3] of arguments used in identifying the geographic focus of a text document was produced.

The rest of the thesis is structured as follows: In Chapter 2, a comprehensive bibliographic review of the areas of story understanding, crowdsourcing, knowledge bases, knowledge acquisition and argumentation is presented. The connection between these fields is also explored. Chapter 3 presents how argumentation is used as a substrate for knowledge representation and provides examples on suitable frameworks inline with our line of work. Chapter 4 provides insights on our contribution for manual encoding of stories and commonsense knowledge using a visual tool. This contribution includes the Web-STAR IDE, a platform that facilitates the writing of stories in symbolic format along with knowledge for story comprehension and a visual output of the comprehension model. In Chapter 5, we demonstrate a methodology and a platform for acquiring background knowledge using games. We present two Games With a Purpose which were created to help the acquisition and verification of knowledge. In Chapter 6 we present our efforts for using crowd-contributed knowledge for inferring the geographic focus of news-stories and in Chapter 7 an overview of this work is presented along with possible future expansions.

[2] The dataset is available at https://cognition.ouc.ac.cy/robot

[3] The dataset is available at https://cognition.ouc.ac.cy/geomantis

2

Bibliographic Review

> *"Every system that we build will surprise us with new kinds of flaws until those machines become clever enough to conceal their faults from us."*
>
> – Marvin Minsky, *The Emotion Machine (2006)*

In this chapter we provide an extensive literature review on the state-of-affairs of the topics that this thesis deals with, such as story understanding, crowdsourcing and knowledge acquisition. It is important for the reader to understand prior and current work on these subjects so as to identify the contributions of this thesis in the relevant fields. We present a number of systems that were developed aiming in the direction of automated story understanding. These systems were tested using question-answering techniques, textual entailment and geographic focus identification. Furthermore, we highlight initiatives for knowledge acquisition using human contributors, automatic extraction of knowledge from text, and hybrid methods.

2.1 What is a Story or a Narrative

Before getting into the deep of story understanding one should first know what a story is. Currently there are many definitions of what a story or narrative is. Here we present some of these definitions that use events, actions, actors and the results of a narrative:

- **Definition 1** by Ricoeur (1980): "I take temporality to be that structure of existence that reaches language in narrativity, and narrativity to be the language structure that has temporality as its ultimate reference."

- **Definition 2** by Genette et al. (1982): "One will define narrative without difficulty as the representation of an event or of a sequence of events."

- **Definition 3** by Prince (1982): “Narrative is the representation of at least two real or fictive events in a time sequence, neither of which presupposes or entails the other.”
- **Definition 4** by Brooks (1992): “Plot is the principal ordering force of those meanings that we try to wrest from human temporality.”
- **Definition 5** by Prince (2003): “The representation ... of one or more real or fictive events communicated by one, two or several ... narrators ... to one, two or several narratees.”
- **Definition 6** by Abbott (2008): “Narrative is the representation of events, consisting of story and narrative discourse, story is an event or sequence of events (the action), and narrative discourse is those events as represented.”
- **Definition 7** by Bal and van Boheemen (2009): “The transition from one state to another state, caused or experienced by actors.”
- **Definition 8** by Landa and Onega (2014): “The semiotic representation of a sequence of events, meaningfully connected in a temporal and causal way.”

By reading the aforementioned definitions, one can easily understand that a narrative can be any type of document, from a simple text passage to a whole novel, as long as the appropriate properties are present. In this work we use the view of Rick Altman (2008), as reported by Michael (2013b), who argues that “*virtually any situation can be invested with [those] characteristics [necessary to] perform the narrational function*”. This definition encompasses from short text documents such as news-stories to large novels and gives us the flexibility to experiment with various type of stories.

2.2 Story Understanding and Text Comprehension

Research in the hard problem of story understanding goes back to the 70’s and starts with a memo from McCarthy (1990), who discusses the difficulty of having a machine that is able to understand a story from the New York Times. Up until today, there is no machine that can understand such a story at a level near that of a human reader. Several methods and systems have been developed to date, that try to deal with the problem of story understanding and text comprehension. Most of these systems are based on symbolic representation of the story or scripts, i.e., a list of events occurring in a specific domain (Schank and Abelson, 1975) and they follow similar architectures in terms of how stories are encoded symbolically, how the background knowledge is encoded and how the reasoning engine operates.

But what do we mean by story understanding? Story understanding includes the human ability to answer arbitrary questions, find where a story takes place, generate paraphrases and summaries, fill arbitrary templates, make inferences, reason about the story, hypothesize alternative versions of the story, look back over the story, and more (Mueller, 2000). Any system developed for automated story understanding should be able to perform at least one of the aforementioned actions. In the following paragraphs, we present some of the systems developed so far, aiming to tackle the problem of story understanding by machines.

Charniak was one of the first who addressed the problem of automated Story understanding. In 1972, he investigated the process of humans answering questions about children's stories (Charniak, 1972). A model was presented that answers "why" questions by relating the story to real world knowledge. This model was used to generate and answer questions through the story progression. In particular, the presented model was used for answering questions related to children's stories by relating them to real-world background knowledge. The model used an internal representation language for the story and required an expert user to encode it. Finding a proper method for representing background knowledge, was one of the major issues they encountered. Additionally, the selected representation should also fit the comprehension model.

In 1976, Charniak (1977a) attempted to formalize knowledge of "mundane" wall paintings, using a "frame" representation. The author suggested that there is a "deep" understanding of the activity, since the selected representation dictates both the steps to carry out the activity, the way to apply them, and the explanation of why these should be applied. In this work, the author reports that the way we express in natural language, actually reflects the real complexity of the world we live in and our knowledge of it.

Following his latter work, Charniak (1977b) presented a program called **Ms. Malaprop**. This program used encoded knowledge of the "mundane" wall paintings to answer questions on simple stories dealing with painting. The author provided a semantic representation for the stories, the questions, and the answers. The author recognized that even when the program is completed, it will still not be able to address challenges such as search, matching, diagnosis, visual recognition and problem solving.

During the same period, other researchers such as Robert Schank presented several systems and approaches for story understanding. In the work of Schank et al. (1973), a system called **MARGIE** (Memory, Analysis, Response Generation, and Inference on English) was presented. This system reads sentences in natural language, paraphrases them and presents inferences. The authors contributed both a theoretical and a practical application of their methodology, stating that the theory is important for further expansion of the methodology.

Two years later, Schank and Abelson (1975) presented a theoretical system intended to facilitate the use of knowledge in a text comprehension system using scripts. The authors developed an application called **SAM** (Script Applier Mechanism) that uses scripts to make inferences about domains that are known to the program. The program utilizes causal chains (Schank, 1973) for inference generation. The authors noted that in order for a person to understand, knowledge only is not that useful, as it is very difficult to remember all story related information and a mechanism that allows humans to "forget" the not important parts of the story should be in place.

In the work of Cullingford (1978), SAM was used to read news-paper stories from various domains. The system applied world knowledge to summarize, paraphrase and answer questions on each news-story.

Later, the work of Wilensky (1976) on natural language understanding led to the development of **PAM** (Plan Applier Mechanism) (Wilensky, 1978). The way PAM understood stories, was by analyzing the intentions of the story's characters, and relating these intentions to their actions (Wilensky, 1977).

Lehnert (1977) also presented her work on question answering for story understanding, which was motivated by theories of natural language processing based on the nature of the questions posed and tried to classify them according to how humans understand and answer questions. Lehnert developed a program called **QUALM** which reads stories and answers questions on what was read. This program was a successor of SAM and PAM mentioned in the previous paragraphs.

Further applications and programs that were developed by Schank and his academic descendants can be found in the work of Schank and Riesbeck (1981) and Dyer (1983). More specifically, in the work of Dyer (1983), a theory of memory representation, organization, and processing for understanding narratives was presented along with a computer program called **BORIS**. This program reads and answers questions about divorce, legal disputes and personal favors. The system is able to answer questions about facts and events on narratives using various knowledge sources such as goals, plans, scripts, physical objects, settings, interpersonal relationships, social roles, emotional reactions, and empathetic responses.

Dolan (1989) presented his work on a system called **CRAM** which uses and acquires thematic knowledge. The system is able to read a paragraph-long, fable-like story and either give a thematically relevant summary or generate planning advice for a character in the story.

In the work of Norvig (1989) the problem of text inference was addressed, by attempting to extract proper inferences from a text. The approach used is a "loose" one, unlike the ones used by other researchers (e.g., scripts, plans), that are more structured. The method recognizes six generic classes of inference that rely on patterns of connectivity between concepts. Patterns are discovered and inferences are suggested. The author implemented an inferencing algorithm in the **FAUSTUS** (Fact Activated Unified STory Understanding System) system (Norvig, 1987). This program is able to handle a variety of texts and knowledge.

In 1993, Ram proposed a different approach to story understanding, defining understanding as a goal-directed process which requires the identification of questions that originate from the story and questions that their answers exist in the story. Using the proposed approach, he developed an application called **AQUA** (Asking Questions and Understanding Answers) which is a dynamic system for text comprehension, using the story questions to acquire knowledge. Following his latter work, the same author developed an extension to this program called **Meta-AQUA** system (Ram and Cox, 1994). It was implemented as a computer model of an introspective reasoner that learns using multiple strategies during a story understanding task.

Hobbs et al. (1993a) developed an approach to abductive inference, called “weighted abduction”. By using this method, the problem of text comprehension is viewed as a problem of explaining why each sentence is true. There is an implementation of this method in the **TACITUS** (The Abductive Commonsense Inference Text Understanding System) (Hobbs, 1991). This system performs a syntactic analysis of the text and produces a logical form in first-order predicate calculus and it was used to interpret texts ranging from equipment failure reports to terrorist reports. Part of this project was the creation of a large knowledge base for commonsense knowledge (Hobbs and Martin, 1987) that was used for interpreting discourses.

Story understanding was also explored in the work of Shapiro and Rapaport (1995). A system called **SNePS** was developed, allowing experimentation with story understanding. This system used a propositional semantic network (labeled directed graph) to represent knowledge and it provided an inference package, dealing with node-based reasoning, path-based reasoning, and belief revision.

Narayanan (1997) presented a model for real-time inferring of important features or abstract plans and events. The author demonstrated this by interpreting snippets of newspaper stories in the domain of economics.

The complexity of story understanding is further discussed in the work of Domeshek et al. (1999). The authors suggested that an internal representation is needed for natural language understanding, along with a number of criteria that must be met by such a representation.

During that period, work on **Deep Read** (Hirschman et al., 1999) was published. Deep Read was an automated reading comprehension system that accepts stories and answers questions about them. Deep Read’s creators used a corpus to conduct experiments using questions on stories with known answers. The system uses pattern matching techniques, enhanced with automated linguistic processing including stemming, name identification, semantic class identification, and pronoun resolution. The system responds with a correct sentence, i.e., a sentence that contains the answer in 30-40% of the cases.

Mueller (2000) describes in detail the state-of-affairs until the 1999's for in-depth story understanding. In that article, the author identifies the difficulty of the specific task, i.e., of having machines being able to understand stories and discusses some of the major problems in building such systems. Furthermore, he provides pointers to possible solutions, tools and resources for building story understanding systems. Mueller points out that many researchers abandoned their efforts on building machines that understand stories due to the lack of progress and moved to research areas that have a more direct impact and results.

Another approach to story comprehension was attempted with work on **Quarc** (Riloff and Thelen, 2000), a rule-based system that reads a story and finds the sentence that best answers a given question. This system uses reading comprehension tests with questions on who, what, when, where, and why. Quarc (QUestion Answering for Reading Comprehension) uses lexical and semantic heuristics to look for evidence that a sentence contains the answer to a question. The rules used by the system were hand-crafted.

Similar to Deep Read, is the work of Wellner et al. (2006) on **ABC** (Abduction Based Comprehension system). This system reads a text passage and answers test questions with short answer phrases as responses. It uses an abductive inference engine which allows first-order logical representation of relations between 1) entities and events in the text and 2) rules to perform inference over such relations. The system is also able to report on the types of inferences made while reasoning, allowing it to provide insights on where it is not performing well and give indications on where existing knowledge needs update or new knowledge is required. The authors reported an accuracy of 35% using a strict evaluation metric.

More recent attempts include work by Mueller (2007) on a system that models space and time in narratives about restaurants. In particular, Mueller's system converts narrative texts into templates with information on the dialogs happening in a restaurant. Then it uses these templates to construct commonsense reasoning problems and finally, it uses commonsense reasoning and the created commonsense knowledge base to build models of the dining episodes. By using these models, it generates questions and answers to the questions posed. The system was evaluated on stories retrieved from the Web and from Project Gutenberg (https://www.gutenberg.org/). The evaluation showed that the system needs much more work to produce highly accurate models.

The majority of the aforementioned systems are based on a symbolic representation of the story or script. These systems follow similar architectures in terms of how stories are encoded symbolically, how the background knowledge is encoded, and how the reasoning engine operates.

Work on story understanding is also performed in MIT's Computer Science and Artificial Intelligence Laboratory, where researchers developed the **Genesis system** (Winston, 2014, 2015). Genesis, deals with both story understanding and story telling. It models and explores aspects of story understanding using stories drawn from sources ranging from fairy tales to Shakespeare's plays. The system uses the START parser (Katz, 1997) to translate English into a language of relations and events that the system can understand. This system is deployed using the Java WebStart mechanism[1].

Chaturvedi et al. (2017) proposed a model for story comprehension which relies on the sequence of events, the emotional trajectory of the story, and its plot consistency. The model is tested on the Story Cloze Test (Mostafazadeh et al., 2016).

There is also work on story comprehension using Answer Set Programming (ASP), such as the methodology presented by Chabierski et al. (2017) on encoding natural language texts to ASP using Combinatory Categorial Grammars. This method creates a knowledge base which can be combined with commonsense knowledge and queried to return answers. The authors experimented on small datasets with promising results.

In the preliminary work of Kim et al. (2019), the approach that knowledge should be generated in the form of abstract logical schemas is followed. This approach requires a semantic parser to kickstart the system, an inference engine capable to reason using an expressive logical form, and a set of simple schemas that a very young child could plausibly possess, to start with.

Furthermore, work on **CoRg** (Siebert and Stolzenburg, 2019), a system that performs commonsense reasoning and story understanding using a Theorem Prover and machine learning techniques, promises to close the gap between good performance and explanability of the reasoning process. The system makes use of knowledge bases such as ConceptNet and WordNet for its reasoning process.

The current trend in story understanding line of research focuses on neural networks, i.e., complex algorithms which mimic the brain's biological processes, for training systems and models. In the recent SemEval task on "Machine Comprehension Using Commonsense Knowledge" (Ostermann et al., 2018b) 10 out of 11 systems used recurrent neural network (RNN) techniques to encode text, questions and answers and the other team used clustering techniques and scoring word overlap. Only 3 teams used a commonsense knowledge

[1] https://www.java.com/en/download/faq/java_webstart.xml

base in their systems. Moreover, newer approaches on using commonsense for machine comprehension such as the work of Chen et al. (2018) and Liu et al. (2018) use similar techniques with the ones previously mentioned.

In work of Sukhbaatar et al. (2015) an approach using a neural network with a recurrent attention model over a possibly large external memory network (Weston et al., 2015) is presented for answering questions on very simple (toy) stories from the bAbI dataset (Weston et al., 2016). The authors show that a neural network with an explicit memory and a recurrent attention mechanism for reading the memory on diverse tasks from question answering to language modeling can be used to answer questions. To better understand the task, we present the following example story from the bAbI dataset:

> John was in the bedroom.
> Bob was in the office.
> John went to kitchen.
> Bob travelled back home.
> Where is John? **Answer: kitchen**

The answer to the question is found in a subset of the available information whereas the rest of the sentences are distractors. Even though the results of the presented approach are good, the model is performing worse than other models trained with strong supervision and still fails tasks that require deduction and search.

Recent advances in NLP using contextual word embeddings and datasets such as ELMo (Embeddings from Language Models) (Peters et al., 2018) and BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) have given rise to the development of new systems for Question Answering. In fact, the first four positions in the GLUE (The General Language Understanding Evaluation) benchmark are occupied by systems employing BERT. Contextual embeddings work by assigning each word a representation based on its context, which allows the usage of this representation in various contexts, as opposed to non-conceptual embeddings which capture a single representation for each word in a specific context, such as Word2Vec (Mikolov et al., 2013).

In the work of Li et al. (2019) a transferable BERT (TransBERT) three-stage training framework was used, which can transfer both general language knowledge from large-scale unlabeled data and specific kinds of knowledge from various semantically related supervised tasks, for predicting the ending of a story. They reported an accuracy of 91.8% for the Story Cloze Test which is close to human performance.

One of the downsides of using contextual word embeddings is that it made the interpretation of the syntactic and semantic properties learned by their inner representations of the systems more complex (Miaschi and Dell'Orletta, 2020). Moreover, when BERT is used for question answering tasks which do not explicitly mention the correct answer in the text, but the answer is inferred then the accuracy is reduced. In work of Huang et al. (2019), a dataset which was specially crafted to require commonsense knowledge to answer multiple choice questions was used to test BERT against human answers. The results identified a large gap (25.6%) in accuracy between machine comprehension and human comprehension performance.

The readers can better understand the downsides of neural networks by testing them to complete a simple short story. Take for example the story snippet “George picked up the gun. He saw a bird.” and use a NN to complete this. One of the NN suggestions is “George picked up the gun. He saw a bird. He shot at it. His dad yelled at him to shoot it in the head, but George hit it twice in the neck...”. Which is indeed a possible coherent continuation of that story. Next we add a sentence to that story and change it to “George picked up the gun. The gun had no bullets. He saw a bird.”. Response from the NN seems to ignore the fact the George’s gun is empty and responds back “George picked up the gun. The gun had no bullets. He saw a bird. He shot at it. It flew away. George didn’t notice.”.

At this point one can easily observe that work on automated story understanding and machine comprehension makes a turn to the early days of AI where statistical methods where employed. Moreover, it seems that researchers are giving greater attention in building systems that maximize performance, but lack the ability to provide explanations for their decisions and provide evidence on how they reached to a certain conclusion, as humans would be able to provide. There is a lot of criticism that the way these language models work is still far-away from real understanding and the results are just based on probabilities.

Following a different paradigm than that of neural networks for story understanding and strict logical representations, Diakidoy et al. (2014) used an argumentation based approach (Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008) to develop a computational method compatible with psychological evidence (Mercier and Sperber, 2011) of human comprehension. The use of argumentation gives a uniform solution to the problems of frame, ramification, and qualification, as well as the problem of contrapositive reasoning with default information. Argumentation is suggested as a more appropriate substrate for the development of automated systems that interact with humans (Kakas and Michael, 2016; Michael, 2017).

In the work of Diakidoy et al. (2015) and in Chapter 3, Section 3.2.1, the STory comprehnesion though ARgumentation system (STAR) is presented. This is a system that uses argumentation-based semantics, while able to perform reasoning over time which is suitable for story comprehension. The system is able to read a story in symbolic format and by using world knowledge in the form of symbolic inference or causal rules it outputs a comprehension model. This model includes the story facts that hold or not on specific time-points. Furthermore, this system is also able to handle preferences between rules, i.e., when two conflicting rules are activated at a specific timepoint, then the preferred rule is used for the comprehension process and the other is ignored. This also applies to human reasoning, since us humans have more than one rules for the same situation, but we activate only the ones needed for a given situation. This is backed up by reports from psychology on cognitive economy (Diakidoy et al., 2014).

At this point one should note that story understanding is not limited to textual content only. There are systems and work for comprehending other type of content and media, such as video or pictures or comic books, offering both textual content and pictures. In the work of Iyyer et al. (2016), deep neural architectures are tested on cloze-style tasks and the results show that text comprehension and image comprehension alone are not enough for a machine to comprehend a comic book story. In the work of Zellers et al. (2018a), an attempt is made to develop machines that can infer people's actions, goals, and mental states from an image. This task goes beyond simple image recognition tasks, requiring a system that is able to perform commonsense reasoning and justify its answers. The authors named this task "Visual Commonsense Reasoning" and managed to create a dataset that includes 290.000 multiple choice questions and answers derived from 110.000 movie scenes to test a system's ability to answer correctly. The focus of our work presented in this thesis is to story understanding from textual sources and not from images or videos.

In terms of technical skills and expertise needed to use the aforementioned systems, the majority of them use a command line interface (CLI) and require users to prepare input files (e.g., story, background knowledge rules) using external tools. The output of these systems is generally in textual form, which makes it difficult to inspect the resulting model. An exception to this, is the Genesis system, which has a graphical interface and provides a visual way to represent the output, but it is still a stand-alone application that requires installation on the user's device. What is common in all systems, is the requirement for commonsense knowledge. Each system offers its own mechanism to represent this knowledge and retrieve it.

Story Understanding by Answering the "Where" a Story Takes Place Question

The ability to understand a story is not only tested with question answering but also with the ability of the agent, human or machine, to identify certain characteristics of a story, such as the location where the story takes place, the protagonist and the timeline (Bower, 1976). Humans are able to read a text passage or a story and identify where that story takes place, i.e., its geographic focus (Tversky, 1993). Silva et al. (2006) give the following definition:

"*Geographic scope or focus of a document is the region, if it exists, whose readers find it more relevant than average.*". Stories are examples of such texts, that human readers can identify.

Going back to the 90's, there was work (Andogah et al., 2012) in the area of identifying the geographic focus of text, that resulted to the development of several systems. A substantial amount of these systems rely on geoparsers, i.e., systems for extracting places from text (Leidner and Lieberman, 2011; Melo and Martins, 2016), for identifying locations, disambiguating them, and finally for identifying the geographic focus of the text. These systems perform well when documents include place mentions for geoparsers to work, but leave open the case of documents that have none or very few place mentions. It is common for a document to also contain references to geographic locations in the form of historical dates, monuments, ethnicity, typical food, traditional dances and others (Monteiro et al., 2016). These references can be used to infer the geographic focus of a text document and it is the reason why we need machines with commonsense knowledge.

During that decade (90's), the Geo-referenced Information Processing SYstem **GIPSY** (Woodruff and Plaunt, 1994) was created. This system was able to resolve the locations of places in documents related to the region of California. For performing that process, a subset of the US Geological Survey's Geographic Names Information System (GNIS) database was used. GIPSY's document processing pipeline includes three steps. Firstly, the system extracts keywords and phrases from each document according to their spatial relatedness. Each of these phrases are weighted according to a heuristic algorithm. Secondly, the system identifies the spatial locations for the keywords and phrases extracted in the first step using synonyms and hierarchical containment relations. Thirdly, geographic reasoning is applied and after extracting all the possible locations for all the terms and phrases pointing to places in a given document, the final step presents the geospatial footprints as a three-dimensional polyhedron.

In the 00's, the **Web-a-Where** system (Amitay et al., 2004) was introduced, which can identify a place name in a document, disambiguate it, and determine its geographic focus. This system detects mentions of places in a document or a webpage and determines the location each place name refers to. Moreover, it assigns a geographic focus to it by using a similar workflow with the GIPSY system and it also has a specific approach for disambiguating locations for both geo/non-geo and geo/geo ambiguity. When the name of a place is the same with the name of a non-place (e.g., Turkey the country and Turkey the bird), a geo/non-geo ambiguity is identified. When two or more places have the same name (e.g., Athens in Greece and Athens in the USA), a geo/geo ambiguity is identified. Furthermore, the system can assign a geographic focus to a document, even though its location is not explicitly mentioned in it, but it is inferred from other locations. The Web-a-Where system was evaluated using two different pre-annotated datasets. The authors reported that their system detected a geographic focus in 75% of the documents and reported a score of 91% accuracy in detecting the correct country.

Silva et al. (2006), presented a system for automatically identifying the geographic scope of web documents, using an ontology of geographical concepts and a component for extracting geographic information from large collections of web documents. Their approach involves a mechanism for identifying geographic references over the documents and a graph ranking algorithm for assigning geographic scope. Initial evaluation of the system, suggests that this is a viable approach.

Related to this line of research, is the work on **SPIRIT** (Purves et al., 2007), a spatially aware search engine which is capable of accepting spatial queries in the form of <theme> <spatial relationship> <location>. Relevant research is also found in the work of Yu (2016) on how the geographic focus of a named entity can be resolved at a location (e.g. city or country).

Furthermore, work done on a system called **Newstand** (Teitler et al., 2008), monitors RSS feeds from online news sources, retrieves the articles in realtime and then extracts geographic content using a geotagger. These articles are grouped into story clusters and are presented on a map interface, where users can retrieve stories based on both topical significance and geographic region.

An attempt to develop a geo-referencing system was also made within the **MyMose project** framework (Zubizarreta et al., 2009). The developed system, performed a city-level focus identification using dictionary search and a multistage method for assigning a geographic focus to web pages, using several heuristics for toponym disambiguation and a scoring function for focus determination. The authors reported an accuracy of over 70% with a city-level resolution in English and Spanish web pages.

More relevant work, mainly concentrated in using knowledge bases extracted from Wikipedia, is presented in the work of de Alencar and Jr (2011) and Quercini et al. (2010). de Alencar and Jr (2011), presented a strategy for tagging documents with place names according to the geographical context of their textual content by using a topic indexing technique that considers Wikipedia articles as a controlled vocabulary. Quercini et al. (2010), discussed techniques to automatically generate the local lexicon of a location by using the link structure of Wikipedia.

A similar to the Web-a-Where system workflow was used in the **CLIFF-CLAVIN** system (D'Ignazio et al., 2014), which identifies the geographic focus of news stories. This system uses a three step workflow to identify the geographic focus of a text. First, it recognizes toponyms in each story, then it disambiguates each toponym, and finally, it determines the focus using the "most mentioned toponym" strategy. This system relies on "CLAVIN"[2], an open source geoparser that was modified to facilitate the specific needs of news story focus detection. The authors reported an accuracy of 90-95% for detecting the geographic focus when tested on various datasets. This system is freely available under an open source license. It is also integrated in the MediaMeter[3] suite of tools for quantitative text analysis of media coverage.

Moving next, a system called **TEXTOMAP** (Brun et al., 2015), aims to design the geographic window of the text, based on the notion of important toponyms. Toponym selection is based on spatial, linguistic or semantic indicators.

Interesting is also the work on **Mordecai** (Halterman, 2017). This system performs full text geoparsing and infers the country focus of each place name in a document. The system's workflow extracts the place names from a piece of text, resolves them to the correct place, and then returns their coordinates and structured geographic information. This system utilizes a number of natural language processing techniques and neural networks to perform these tasks. A number of newly developed systems, such as GeoTXT, make use of Mordecai to their own pipelines.

Imani et al. (2017), proposed a mechanism that utilizes the named entities for identifying potential sentences containing focus locations and then uses a supervised classification mechanism over sentence embedding to predict the primary focused geographic location. The unavailability of ground truth (i.e., whether words in a sentence constitute a focus or non-focus) suggests a major challenge for training a classifier and an adaptation mechanism is proposed to overcome sampling bias in training data. This mechanism was evaluated against baseline approaches on datasets that contain news articles and showed better results than the other systems tested on the same dataset.

A system called **Newsmap** (Watanabe, 2018), uses a a semi-supervised machine learning classifier to label news stories without human involvement. Furthermore, the system identifies multi-word names to automatically reduce the ambiguity of the geographical traits. The authors evaluated their system's classification accuracy against 5000 human-created news summaries. Results show that the Newsmap system outperforms the geographical information extraction systems in overall accuracy, but authors report that simple keyword matching suffers from ambiguity of place names in countries with ambiguous place names.

One of the most recent developed systems is **GeoTxt** (Karimzadeh et al., 2019). This is a geoparsing system that can be used for identifying and geolocating names of places

[2] https://clavin.bericotechnologies.com/

[3] http://mediameter.org/

in unstructured text. It exploits six named entity recognition systems for its place name recognition process, and utilizes a search engine for the indexing, ranking, and retrieval of toponyms. The system was tested on a dataset of 6,711 manually geo-annotated tweets with each of the six named entity systems to compare results.

In Chapter 6 we use two of these systems (CLIFF-CLAVIN and Mordecai) to compare their performance with a system we developed for identifying the geographic focus of a story. These systems were chosen because they are open source, freely available and actively maintained.

Story Understanding and Explanation

The notion of story understanding is closely related to the notion of explanation. There has been an interesting debate and work on this connection, mostly from the philosophical point of view. In the work of Friedman (1974) an attempt was made to combine the use of narratives with explanation of scientific notions with the purpose to answer why and how questions. Velleman (2003) states that "*A story does more than recount events; it recounts events in a way that renders them intelligible, thus conveying not just information, but also understanding. We might therefore be tempted to describe narrative as a genre of explanation.*". Carroll (2001) describes a narrative as a common form of explanation since it is usual to use narratives to explain how things happened. This is also connected to the causal relations of the events in a narrative. Forster (2010) uses the term "plot" to describe a story that is distinguished by the "why?" question and to separate it from one that is connected with the "and then?" question. The first is a form of explanation since one needs to answer the "why" question that includes a causal link between the story concepts. The work of Roth (1989) includes a discussion on whether narratives provide explanations.

Recent work by Morgan (2017) investigates the role of narratives in the social science case-based research, by creating a productive ordering of the materials within such cases, and on how such ordering functions in relation to "narrative explanation".

In this work, a number of tools were developed, such as the Web-STAR IDE (cf. Chapter 4) which handles the encoding of both causal rules in the background knowledge and questions in the story that provide explanations on the story concepts. Users can take advantage of the debugging options offered by the IDE to get in-depth insights on the inferences made to provide answers to questions and hence lead to the relevant explanation.

2.2.1 Story Understanding Datasets and Corpora

For testing how well a system performs on understanding a story, researchers created pre-annotated datasets. These datasets include stories and questions on each story with their corresponding answer. Most of them are created using crowd-workers and machines and are tested on how well they can answer questions on each of the stories. Following, is a list of the latest available datasets for testing the ability of a system to understand stories, either by answering questions or by textual entailment, i.e., "what" comes next.

The CommonsenseQA Dataset (Talmor et al., 2019) comprises 12,247 questions, aiming to be easily answered by humans without context, requiring commonsense knowledge. These questions were derived from ConceptNet (Speer et al., 2017) using closely-related concepts. They used crowd-workers for adding distractors to each question and asked crowd-workers to formulate the questions in a way that only one of the possible distractor answers can be chosen. This dataset was tested using several models, and the best one achieved a 55.9% accuracy, leaving plenty of space for improvements.

The CoQA Dataset (Reddy et al., 2019) consists of 127,000 questions with answers, extracted from 8,000 conversations about text passages from seven diverse domains. When the dataset was tested using several systems, it achieved an F1 score of 65.4% using the best result from all tested systems, while human performance was measured at 88.8%.

The SWAG Dataset (Zellers et al., 2018b) is one of the newer datasets created from the The Allen Institute for Artificial Intelligence. It includes 113,000 multiple choice questions on a broad spectrum of grounded situations. The authors use Adversarial Filtering, i.e., a method for iteratively training an ensemble of stylistic classifiers and using them to filter the data to address human biases that already exist in many datasets.

The OpenBookQA Dataset (Mihaylov et al., 2018) is a question-answering dataset which consists of 5,957 multiple-choice elementary-level science questions. These questions test the understanding of science facts from a book and the application of these facts to novel situations. For a machine to answer these questions, it needs to have broad understanding of the world, hence commonsense knowledge that is not contained in the book. When neural network approaches were tested on answering questions, they achieved around 50%, whereas crowd-workers achieved 92% accuracy.

The MCScript (Ostermann et al., 2018a) is a dataset which highlights reasoning with commonsense knowledge. It comprises 14,000 multiple-choice questions on 2,100 narrative texts. Most of these questions require knowledge beyond the facts mentioned in the text. When tested with various models, it achieved an accuracy of 72% whereas human performance was measured at 98.2%.

The NarrativeQA Dataset (Kočiský et al., 2017) consists of 1,572 stories from books and movie scripts and also 46,765 human generated questions along with their answers, produced from summaries. For a system to successfully answer these questions, it must understand the narrative, instead of just shallow parse the text.

The ROCStories Corpus (Mostafazadeh et al., 2016) comprises 98,159 five-sentence commonsense stories. The corpus includes causal and temporal commonsense relations between daily events. In this work, the authors also presented the Story Cloze Test, which was used to test a system's ability to understand a story by identifying the correct ending to a four-sentence story.

The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) is a reading comprehension dataset which includes more than 100,000 questions posed by crowd-workers on a set of Wikipedia articles. The answer to each question is a text passage from the corresponding Wikipedia article. Human performance was measured at 86.8%.

The Triangle-COPA (Gordon, 2016) dataset comprises 100 short stories and movies about actions between shapes (triangles, circles, etc.) that are encoded in logic format and used to answer simple questions with two plausible answers (one more plausible than the other). This dataset uses a fixed vocabulary of predicates (122 in total) to avoid having to treat the dataset with natural language processing tools. Furthermore, the dataset is focused on commonsense reasoning and specifically on the type of reasoning people use, based on human psychology. In Chapter 5 we use this dataset both for knowledge acquisition and for testing the acquired knowledge on answering questions.

The Children's Book Test (CBT) (Hill et al., 2015) is a dataset composed of freely available children books from project Gutenberg. Chapters in each of the selected books are used to form questions by enumerating 21 consecutive sentences. In each question, the first 20 sentences form the context, and a word is removed from the 21st sentence, which becomes the query. Benchmarked systems need to choose the answer word between a selection of 10 possible answers that appear in the context sentences and the query.

The MCTest (Richardson et al., 2013) is a dataset composed of 500 fictional stories (660 in total, but the main dataset comprises 500) along with 4 multiple choice questions for each story. The dataset was built using crowd-workers from Amazon Mechanical Turk (Buhrmester et al., 2011a).

Additionally to the above, there are much larger corpora available, that are useful in answering the "where a story takes place" question or identify the category in which each story falls under, such as:

The New York Times Annotated Corpus (NYT) (Sandhaus, 2008) contains in its collection over 1,800,000 articles in English language, written and published by the New

York Times between 1987 and 2007. Most articles are tagged with location metadata by human annotators. The NYT corpus categorization allows a news story to be tagged with more than one locations. This corpus is available under a copyright agreement of the publisher and it requires a license fee to obtain it.

The Reuters Corpus Volume 1 (RCV) (Lewis et al., 2004) includes 810,000, English language news stories that were made available in 2000 by Reuters Ltd. The corpus contains stories from 20/08/1996 to 19/08/1997, tagged with information on where the story is geographically located. Tagging was performed by a combination of automatic categorizing techniques, manual editing, and manual correction. This corpus is free to use, as long as the user agrees to the copyright agreement of the publisher.

The list of datasets and corpora reported in this section is not exhaustive and much more are available, but the above are prevailing in the research for story understanding during the last few years and each of them serves a specific purpose. Interested readers can get a comprehensive report and what is currently available in terms of datasets and copora for commonsense knowledge and reasoning in the work of Storks et al. (2019). In the next chapters of this thesis, some of the mentioned datasets and corpora are used in the experiments and systems we designed.

In the next sections, an overview of the area of crowdsourcing or human computation is presented and is linked with efforts for acquiring commonsense knowledge.

2.3 Crowdsourcing

Crowdsourcing is a term that first appeared in a Wired magazine article (Howe, 2006). In that article, the author defined crowdsourcing as "*the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers.*".

In the work of Brabham (2008), crowdsourcing is defined as an "*online, distributed problem-solving and production model*". Estellés-Arolas and González-Ladrón-de Guevara (2012), provide a more integrated definition taking under consideration the definitions provided by other authors. Their proposed definition is the following: "*Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of* *the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowd-sourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken.*". This definition was verified three years later after a repetition of their study (Estellés-Arolas et al., 2015). A shorter definition is presented in the work of Wang et al. (2013) as "*a strategy that combines the effort of the public to solve a problem or produce a resource.*".

In all definitions, the role of information technology and the internet as a medium is present and is stressed out as an important factor for successful crowdsourcing projects (Doan et al., 2011). Through the years, a number of projects emerged using crowdsourcing, such as crowdvoting (Kirkels, Yvonne E. M. and Post, 2013), crowdsolving, crowdfunding, and microwork.

Various attempts were made to categorize crowdsourcing approaches. According to Geiger and Schader (2014), crowdsourcing approaches can be distinguished according to (i) whether they seek homogeneous vs. heterogeneous contributions and (ii) whether they seek a non- emergent vs. an emergent value from these contributions. Geiger et al. (2011) proposed a classification scheme which concentrates exclusively on the organizational perspective. The resulting metrics are: preselection of contributors, accessibility of peer contributions, aggregation of contributions, and remuneration for contributions. Other taxonomies suggest the distinguish of implicit vs explicit approaches (Doan et al., 2011). Implicit crowdsourcing refers to approaches where users do not necessarily know they are contributing. Explicit crowdsourcing, on the other hand, refers to approaches where users are willingly contributing to create an output that is of common interest for a large number of persons.

Examples of implicit crowdsourcing include Games With A Purpose (GWAPs) (von Ahn and Dabbish, 2008), where players contribute while having fun, and the CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenge to distinguish a human from a machine user (von Ahn et al., 2003). In the case of explicit crowdsourcing, the most widespread example is the Wikipedia where users knowingly submit their articles for the creation of a worldwide encyclopedia. In this category, one can also include paid workers platforms, such as Amazon Mechanical Turk (Buhrmester et al., 2011b), Figure Eight (formerly known as Crowdflower) (Van Pelt and Sorokin, 2012), and Microworkers.com (Nguyen, 2014), where users contribute and get money in return for their work. Requesters (people who have a task that want to be completed) can hire crowd-workers who will complete this task for a certain amount of money. There are a number of advantages of using this incentive, such as the speed of task completion, the selection of workers and cost (there are times that it is more cost-effective to pay for a task to be completed instead of designing a system from scratch). In the work of Chittilappilly et al. (2016) a survey of the various crowdsourcing methods is presented along with pros and cons of each method.

2.3.1 Games With A Purpose

One of the approaches used in implicit crowdsourcing is the use of games for having people contributing to tasks while having fun playing a game. The term **Game With A Purpose (GWAP)** was first coined by Luis Von Ahn in 2014.

GWAP (von Ahn and Dabbish, 2008) is a genre of crowdsourcing and is best described by existing applications such as the ESP game (von Ahn and Dabbish, 2004) and Verbosity (von Ahn et al., 2006a). The purpose of the ESP game was to label images. Two human players are presented with the same image and they try to add the same label to that image. This game was acquired by Google Inc. in 2006 to enhance the company's image labeling technology. An extension to the ESP game is the Peekaboom game (von Ahn et al., 2006b), where players associate a label with a region of an image.

The verbosity game (von Ahn et al., 2006a) is a Taboo like game where two players are selected at random and one is chosen as the "Narrator" while the other is the "Guesser". The "Narrator" is presented with a secret word and tries to make the "Guesser" to find that word by typing hints in the form of sentence templates. The purpose of the game is to gather commonsense knowledge.

Since 2006, a plethora of GWAPS emerged aiming to address several tasks, such as knowledge acquisition, biological processes, medical processes, natural language processing, lexicography and language learning (Lafourcade et al., 2015). Games developed in these fields include, but are in no way limited to Foldit (Cooper et al., 2010), Phylo (Kawrykow et al., 2012), , Phrase Detectives (Poesio et al., 2013), Duolingo (von Ahn, 2013), ZombiLingo (Fort et al., 2014) and many others. We dedicate a section for GWAPs aiming to gather knowledge, since knowledge acquisition is one of the main goals of this work.

GWAPs require a motivation for players to use them. This motivation is "fun" (von Ahn and Dabbish, 2004). Studies showed that knowledge workers contribute more when they are having fun doing so (Law and von Ahn, 2011). GWAPs can be categorized according to the following three templates, according to von Ahn and Dabbish (2008):

- output-agreement games
- inversion-problem games and

- input-agreement games

An **output-agreement game** requires that all players agree on the resulted output of a given task. All players are presented with the same input and are urged to create an output similar to the one of their opponents, without of course knowing what that is. Game designers should provide adequate instructions to players to guide them on that direction while playing the game.

An **inversion-problem game** requires that players are divided into two groups; the "describer" and the "guessers". The "describers" are introduced with a specific input and are required to produce an output that will be sent to the "guessers". The "guessers" must use this output to try and create the initial input.

In an **input-agreement game**, players are presented with the same or different inputs (without players knowing that) and are required to create outputs describing the inputs. Players are then called to decide if the inputs given to them and their partners are the same or not. Each player can only see the output of their partner and not the input.

For the games to be attractive, it is also important to stress out the use of gamification techniques (Morschheuser et al., 2016), such as game-style graphics, high-scores, in-game competitions, etc. Many times these are ignored as the purpose of the game is a different one, but the same design proncipIes used by the game industry should also apply to GWAPs if the developers aim for a game that can retain its players.

2.4 Commonsense Knowledge Acquisition

The importance of commonsense knowledge in story comprehension is highlighted in all systems developed so far and researchers are striving to find ways to retrieve this knowledge, represent it in a suitable format and find ways to reason with it. Commonsense knowledge can be found in 3 forms: factual knowledge, ontological knowledge, and rule based knowledge Zang et al. (2013). Factual knowledge is knowledge that describes facts about an entity, e.g., Cyprus `isA` Country or Donald Trump `isPresidentOf` the USA. Ontological knowledge is knowledge that describes a set of concepts within a domain and the relationship between these concepts, e.g., red wine `isATypeOf` wine. Rule based knowledge is knowledge that describes how the world works, e.g., "If a person sits in a room then this person is in the room".

Various methods and systems were developed to acquire commonsense knowledge. These methods include: 1) the handcrafting of knowledge where human experts try to encode knowledge, 2) the automatic extraction of knowledge from large corpora or the web, and 3) the use of crowdsourcing.

The aforementioned methods require that knowledge is represented in a way that can be reused. Certain researchers in the field (Haase, 1996) claim that an appropriate knowledge representation is based on general axiomatic formulations of different facets of the commonsense world; others claim that symbolic representations (Schank and Abelson, 1977) or concrete rules (Hobbs et al., 1993b) are the right way to represent commonsense knowledge, and yet others claim that routine behavioral activity that operates using purely procedural representations is the appropriate format (Agre and Chapman, 1987). A hybrid approach is proposed in the work of Singh (2002), where commonsense knowledge is allowed to be represented in a variety of formats. This diversity makes it more likely to gather appropriate commonsense knowledge for whatever commonsense problem one is faced with at the moment.

A more recent approach suggests that argumentation is an appropriate substrate for representing knowledge. Reports from psychology (Diakidoy et al., 2014) state that "inference generation is a task-oriented process that follows the principle of cognitive economy enforced by a limited-resource cognitive system". Humans understand a story by integrating story related knowledge with commonsense knowledge. This is due to the fact that humans have limited cognitive resources which leads to the activation of only a small restricted subset of the available commonsense knowledge (Gerrig, 2005). Moreover, humans do not have a single commonsense knowledge rule for each situation. They are more likely to have a series of rules that might be conflicting and at a given time only some of these commonsense knowledge rules are activated and the rest are ignored. The notion of commonsense knowledge rule preferences is introduced to describe this process.

In terms of commonsense knowledge acquisition, there are several approaches and systems that were built to deal with the problem of commonsense knowledge acquisition. Most of them employ various techniques for gathering factual commonsense knowledge and only few of them deal with the problem of commonsense knowledge acquisition in the form of rules. The majority of these systems use knowledge engineers as a source of knowledge and use symbolic languages (e.g., CycL, First Order Logic notation etc.) for representing acquired rules.

2.4.1 Knowledge Acquisition Systems and Knowledge Bases

Our review of knowledge acquisitions systems and knowledge bases starts with the most promising and long running approach so far, the **Cyc** Platform (Lenat and Guha, 1989). Cyc started as an effort to manually encode commonsense knowledge rules and currently delivers a whole suite of tools, such as: (i) The Cyc Knowledge Base which includes an ontology of approximately 1.5 million general concepts, 20 million general rules and assertions involving those concepts, (ii) tools to perform reasoning, and (iii) tools to query and access knowledge and software tools for creating, modifying, testing, and deploying Cyc's applications (Lenat, 2019). There used to be a free version of Cyc (ResearchCyc) which is currently not available and the knowledge base is only delivered through a paid license. This project has been active since 1989 and just to have a measure of difficulty of hand coded knowledge, more than 2000 Ph.D. scientist-years of effort were required so far for encoding knowledge. The major advantage of this approach, is that high-quality knowledge rules are produced.

Witbrock et al. (2005) proposed a system built on top of Cyc to extract commonsense knowledge rules using machine learning techniques to ground facts. As the authors state, these rules are not guaranteed to be correct, so a review and validation process is needed. Even though this is still a manual process, it is much easier to review a commonsense knowledge rule than to create it from scratch.

WordNet (Fellbaum, 2010) is another example of a manually created knowledge base. Work on WordNet started in 1986 and resulted in a large semantic network where words that are synonyms are grouped into unordered sets called synsets. In its current version (WordNet 3.1) it contains 117,000 synsets, along with their semantic relations to other words (e.g., antonymy, hyponymy, hypernymy, entailment). Work on WordNet inspired also the development of similar resources for verbs, i.e., VerbNet

Work on **ThoughtTreasure** by Mueller (1998) also got attention. This knowledge base includes 25,000 concepts and 50,000 assertions such as `[isA soda drink]` (a soda is a drink). Moreover, it contains 100 scripts, i.e., representations of typical activities and 29 grids, i.e., setup of objects in common places such as kitchens and bedrooms. ThoughtTreasure was released in 2000 as an open source project and it is available at GitHub[4]. It is alarming that after this work, Mueller expressed his concerns about the future of symbolic AI. By reviewing the current state-of-affairs we can see that indeed, researchers drew their attention on statistical methods and moved away from symbolic formats that require much work on manual coding of rules. Nevertheless, there are researchers that focused their attention in automatic methods for extracting knowledge from large texts, such as corpora or the web. The following are just some examples of such attempts:

Starting with work on the **KNEXT** and **LORE** (Gordon and Schubert, 2011, 2010; Schubert, 2002) projects, the Penn Treebank corpus was used to extract general probabilistic knowledge, i.e., relationships implied to be possible in the world. According to the project's website[5], KNEXT includes 73,701,863 unique factoids extracted from sources like Wikipedia,

[4] https://github.com/eriktmueller/thoughttreasure

[5] http://www.cs.rochester.edu/research/knext/browse/

the Brown Corpus and other sources and are represented in symbolic format. The continuation of the KNEXT project is called LORE[6].

There is also work on **KnowItAll** (Etzioni et al., 2005), which proposes an automatic way for extracting facts from large collections from the Web using an unsupervised, domain-independent, and scalable manner. The system is able to extract facts such as names of cities or names of politicians. This system managed to gather 50,000 class instances.

Furthermore, the **TEXTRUNNER** (Banko et al., 2007) system is presented, which is an implementation of the Open Information Extraction (OIE) paradigm. In this paradigm, the system extracts relation triples from a single data-driven pass over a corpus without requiring any human input. Developers of TEXTRUNNER compared it to KnowItAll system, matching its recall metric and achieving better precision.

Sharma and Forbus (2010) investigated the usage of **Plausible Inference Patterns (PIP)** on fully grounded queries aiming in improving the performance of question answering systems. The authors presented a number of examples extracted from the ReasearchCyc knowledge base. By examining these examples, one can observe that a highly trained knowledge engineer is required to create such rules and this process can not be scaled enough to gather a substantial amount of commonsense knowledge rules.

In the work of Michael (2013a), the notion of “WebSense” is introduced, i.e., knowledge found in the Web, for building a system which can crawl the web for knowledge, parse identified text snippets, learn rules, reason with acquired knowledge, and generate text as answer to queries. The author argues that machines today are capable to perform such a pipeline.

Larger scale projects include the **Microsoft Concept Graph** (Wang et al., 2015) and the **Never Ending Language Learner (NELL)** (Mitchell et al., 2015). The **Microsoft Concept Graph** is a continuation of the Probase project (Wu et al., 2012) which extracted 2.7 million concepts (e.g., animals, books, etc.) automatically from a corpus of 1.68 billion web pages. Factual knowledge is captured in the form of `isA` relationships between concepts.

The Never Ending Language Learner (NELL) is a project that started in 2010 with the ambition to create a machine that is able to “Read the Web”. It was feeded with an initial ontology and categories and since that date it is crawling the web to extract factual knowledge. So far, more than 50 million candidate beliefs have been retrieved, and 2,810,379 (5.62%) of these beliefs were marked with high confidence. Sometime after the initial launch of the project, the ability for humans to verify these beliefs was added so as to facilitate the learning process of the machine.

[6] http://cs.rochester.edu/research/lore/

2.4.2 Knowledge Acquisition Using Crowdsourcing

There is a number of attempts that use crowdsourcing and GWAPs in particular to acquire commonsense knowledge. A literature review was conducted that revealed a number of games developed to date. The first GWAP developed is **Verbosity** (von Ahn et al., 2006a), which was presented in the previous section.

Next, Lieberman et al. (2007) presented **Common Consensus** game, which aims in collecting commonsense knowledge from peoples' everyday goals. This is a web based GWAP that is based on an American TV game show called Family Feud, where players had to answer questions based on templates, to extract goals. According to the authors, the game was launched for a test run with few players and the amount of unique answers retrieved were approximately 550. The extracted commonsense knowledge goals are in natural language.

Orkin and Roy (2007) presented the **Restaurant Game**, where player actions and behavior in a virtual restaurant world are recorded, encoded, and visualized on a plan network using 5000 gameplay sessions by 7504 players. More specifically, a virtual restaurant environment was created where players interacted with other players and these interactions led to the gathering of high quality data that reflect typical human behavior and language.

Vickrey et al. (2008) presented three online games for collecting semantic relations between words (e.g., hypernym/hyponym relationships). These games were based on Scattergories$^{\text{TM}}$ and Taboo$^{\text{TM}}$ real-life games. The first two games, **Categorilla** and **Categodzilla** were inspired by Scattergories$^{\text{TM}}$, in which players are asked to type words or phrases which fit specific categories (e.g., "Things that fly" or "Types of fish"). The third game called **Free association** is based on Taboo$^{\text{TM}}$ and players are asked to type words related to a specific word, without using certain "Taboo words", just like normal taboo games.

In the work of Siorpaes and Hepp (2008a), a game framework called **OntoGame** (Siorpaes and Hepp, 2008b) was presented. In this work the authors give a detailed description of the game platform, along with design choices they made and provide examples of games developed using this platform. OntoPronto is one of these games that aims to build domain ontologies from Wikipedia articles by matching these articles with classes in the Proton[7] ontology. Proton is a high-level ontology which can be used as a basis for modelling various tasks in different domains.

Other attempts include the **Rapport** and the **Virtual Pet** games (Kuo et al., 2009) which focus on social interactions between players. The Rapport game is based on user collaboration through a social media platform by using actions like questions, votes, etc. The Virtual Pet game is deployed in a popular bulletin board and players perform actions like feeding a virtual pet and teaching it common sense, aiming in getting more commonsense points.

[7] http://proton.semanticweb.org

Contributions are stored in natural language and according to the authors, in a six-month period the Rapport game managed to gather 14,000 statements and the Virtual Pet game gathered 511,734 statements.

Herdagdelen and Baroni (2010) presented their work on the **Concept Game**. This is an application that was available on Facebook and combined commonsense harvesting by text mining and a GWAP. This GWAP uses commonsense facts mined from corpora and asks players in a simple slot-machine-like game to validate them, i.e., "they make sense on not". The game was tested by 25 players and gathered approximately 5,000 responses.

In the same year, Markotschi and Johanna (2010) presented the **GuessWhat?!** GWAP, where a player is presented with a partial description of a concept and is asked to type the name of an object which matches the description. The game also presents other players responses and asks the player to evaluate them. Two test groups of 5 players each performed an initial evaluation of the game and two different test groups of 6 players performed a final evaluation of the game. In total, all players contributed 59 class expressions.

Thaler et al. (2011) described their work on the ontology alignment process, where researchers need to match, merge, integrate ontologies, and to interlink RDF data sets. The Resource description framework (RDF) is used to describe web resources. They approached the problem by developing a GWAP called **SpotTheLink**, that was built on top of the OntoGame framework. SpotTheLink's purpose was the alignment of the DBpedia (Lehmann et al., 2015) ontology with the Proton ontology. The authors reported that 16 players matched 32 of 246 DBpedia concepts to the Proton ontology. In their evaluation, they stated that almost all players, answered that they would not play the game again in its current form.

Hees et al. (2011), developed a web-game prototype called **BetterRelations** for ranking triples by importance. In this game, two players are presented with a topic (originating from a Linked Data knowledge base) and two terms (facts in symbolic form related to the topic). For a period of 18 days, 359 players initiated 1,041 games and contributed 4,700 matches.

Waitelonis et al. (2011), presented the **WhoKnows?** GWAP which focuses on identifying inconsistencies in Linked Data and score properties to rank them for sophisticated semantic search scenarios. In total 165 users played the game for 781 times and contributed on 4,051 distinct triples in 18,488 rounds. 13,404 of these rounds were answered correctly.

Wolf et al. (2011), presented a quiz game, similar to Jeopardy called **RISQ!**. In this game, questions were automatically generated from linked object data facts and players were asked to provide evaluations. The evaluation of the game was conducted by 118 players who contributed on 6,484 questions, 3,678 of which were answered correctly.

Celino et al. (2012), developed **UrbanMatch**, a GWAP for mobile devices for matching POI (points of interest) with their relevant photos. The game was downloaded 54 times and users played 781 levels in total.

In the work of Scharl et al. (2012) and Sabou et al. (2013), a Facebook-based GWAP called **Climate Quiz** is presented, aiming to capture knowledge and create metadata through collaborative ontology building in the domain of climate change. Players can engage in two types of challenges, i.e, to select the correct relation between two environmental concepts, and to answer climate-related questions. A total of 648 players tried the game and contributed 19,896 ontology relations and 3,871 quiz answers in a period of 7 months. What is also interesting in this work, is the comparison of a paid-crowdsourcing approach with the GWAP approach for acquiring knowledge. The authors argue that the game-based approach is the most popular but a shift is noticed towards payed-crowdsourcing platforms (Sabou et al., 2013).

In the work of Cambria et al. (2015), a game engine for commonsense knowledge acquisition called **GECKA** is presented. The platform acquires commonsense knowledge while game designers use it to create games with a purpose.

One of the latest works on developing a large-scale GWAP is that of Otani et al. (2016), on a Facebook-based quiz game that collected over 150,000 unique commonsense facts by gathering the data of more than 70,000 players over eight months.

There are also attempts that use hybrid approaches, such as the one of Herdağdelen and Baroni (2012) where a slot-machine GWAP was designed which gathers verification from players on commonsense knowledge facts. The facts are gathered using a text miner which harvests candidate commonsense facts from corpora. Then, a simple slot-machine GWAP presents these candidate facts to the players for verification by playing. The authors claim that "*this combined architecture is able to produce significantly better commonsense facts than the state-of-the-art text miner alone*".

There are also crowdsourcing approaches for knowledge acquisition using crowd-workers that are paid to perform specific tasks. We have already discussed in the beginning of Section 2.3 some of the popular platforms available and one can easily see (cf. examples in Section 2.2.1) that these platforms are used for developing large datasets for story understanding such as the ones presented in the following section.

2.5 Crowdsourced Knowledge Bases and Ontologies

A large amount of commonsense knowledge is stored in knowledge databases in the form of facts. The sources of these knowledge bases include crowd-workers, game players, volunteers, and contributors in general. As presented in the previous section, currently there are many knowledge bases and ontologies available that form an ecosystem of Linked Data[8] providing both humans and machines with a structured form of commonsense knowledge that can be used for reasoning.

In terms of commonsense knowledge, a number of ontologies and knowledge bases are used for experimenting with story understanding systems, such as ConceptNet (Speer et al., 2017) and YAGO (Hoffart et al., 2011; Suchanek et al., 2007, 2008) which are generated mostly by crowdsourcing approaches and include generic factual knowledge for persons, countries, objects and other everyday items and notions used. These knowledge bases do not hold "absolute" knowledge on a specific topic, but rather hold broader knowledge on various topics. A brief overview of these knowledge bases is presented in the following paragraphs, trying to give an introduction to the reader on the content of the next chapters of this thesis, where we use some of these knowledge bases for experimentation on knowledge acquisition and reasoning.

2.5.1 ConceptNet

ConceptNet (Speer et al., 2017) is a freely-available semantic network that contains data from a number of sources such as crowdsourcing projects, Games With A Purpose (GWAPs) (von Ahn and Dabbish, 2008), online dictionaries, and manually coded rules from the ReaserchCyc project. In ConceptNet, data are stored in the form of edges. An edge is the basic unit of knowledge in ConceptNet and contains a relation between two nodes (or terms). Nodes represent words or short natural language phrases. In Figure 2.1 a depiction of the ConceptNet linking between the terms "ConceptNet" and other terms is presented.

ConceptNet is currently in version 5.7 (released in April 2019), holding approximately 34 million edges and 37 relations, such as "AtLocation", "isA", "PartOf", "Causes" etc. The following are examples of edges available in ConceptNet: <monkey> <isA> <primate>, <war> <RelatedTo> <battle>. ConceptNet data can be retrieved by using its Application Program Interface (API). There were also attempts to represented its data in an RDF format (Najmi et al., 2016).

In the work of Ohlsson et al. (2013) ConceptNet's version 4 ability to answer IQ questions using simple test-answering algorithms was evaluated and the results showed that the system has the verbal IQ of an average four-year-old child. Nevertheless, this is a narrow comparison, using only a question-answering task and this is also acknowledged by the ConceptNet team[9].

[8] http://linkeddata.org/

[9] https://github.com/commonsense/conceptnet5/wiki/FAQ

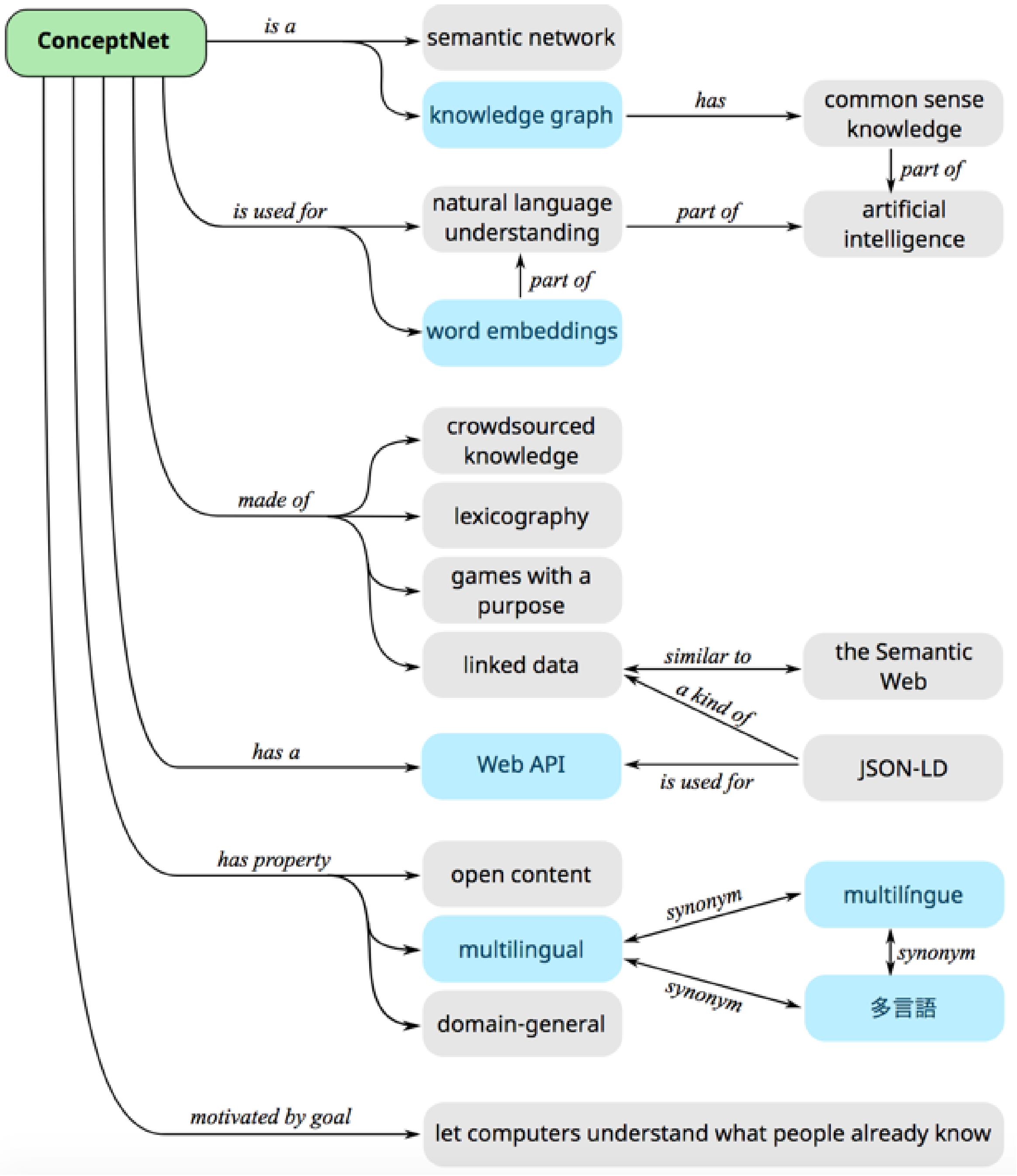


Figure 2.1 A ConceptNet 5 graph structure example with linking the term “ConceptNet” to other terms. Source: http://conceptnet.io/ (Retrieved on 22/08/2019).

2.5.2 DBPedia

DBPedia (Auer et al., 2007) is a project that tries to extract the vast multilingual content from WikiPedia into structured knowledge which in turn can be queried in many sophisticated ways, allowing access to information besides the usual full-text-search. DBPedia also links to other datasets on the Web. In its current state, it includes 103 million Resource Description Framework (RDF)[10] triples.

Users can retrieve data from DBPedia using SPARQL queries (Quilitz and Leser, 2008) and filter the results. Data are multilingual and many other projects use knowledge from DBPedia to expand their existing data.

2.5.3 YAGO

YAGO (Yet Another Great Ontology) is a knowledge base which holds semantic knowledge and is built from sources such as Wikipedia, WordNet (Fellbaum, 2010) and GeoNames[11]. More specifically, information from each Wikipedia page is extracted using the categories, redirects and infoboxes available in each page. A number of relations are also available between facts that are described in detail in the work of Hoffart et al. (Hoffart et al., 2011). Currently, YAGO released its 3rd version, consisting of 150 million facts about 9,8 million entities (like persons, organizations, cities, etc.). Facts in YAGO were evaluated by humans, reporting an accuracy of 95%.

YAGO includes both semantic and technical oriented relations between entities. Semantic relations include: "`wasBornOnDate`", "`locatedIn`" and "`hasPopulation`" and technical relations include: "`hasWikipediaAnchorText`" and "`hasCitationTitle`".

Moreover, YAGO has a number of spatial relations that place an object in a specific location (i.e., country, city, administrative region, etc.) and temporal relations that place an object in time. Relations such as: "`wasBornIn`", "`diedIn`" and "`worksAt`" place an entity of type `Person` at a location. Relations such as "`wasBornOnDate`" and "`diedOnDate`" specify the timeframe of a fact e.g., <`Barack_Obama`> <`wasBornOnDate`> <`1961-08-04`>.

The YAGO knowledge base can be downloaded or queried online using the SPARQL endpoint[12].

[10] https://www.w3.org/TR/REC-rdf-syntax/

[11] http://www.geonames.org

[12] https://linkeddata1.calcul.u-psud.fr/sparql

2.5.4 ATOMIC

ATOMIC is an atlas of everyday commonsense reasoning (Sap et al., 2019). It is one of the recently published datasets which offers 877,000 textual descriptions of inferential knowledge, i.e., if-then relations with variables (e.g., "if X pays Y a compliment, then Y will likely return the compliment"). The dataset consists of 9 if-then relation types. The creators of ATOMIC experimented using neural models to acquire simple commonsense capabilities and reason about previously unseen events.

2.6 Discussion and Approach Followed

A careful study of the bibliography shows that most of the story understanding systems developed since the 70's, are focused on a specific domain or subject area such as terrorism, painting, dinning in restaurants, etc., and require specific background knowledge based on the respective topic. The story comprehension level of the majority of these systems is also limited to the basic events covered in each story and the key actors involved.

In recent years, the shift towards statistical approaches and neural networks is obvious, leading to systems that perform very well on a specific task, but lack the ability to explain their decisions the way humans can. Moreover, systems which rely on neural networks are costly in terms of processing power and are too brittle, meaning that when used on domains outside of those which they were trained on, do not perform well.

Only few systems for story understanding that use symbolic representations are still available and are actively maintained by their developers. The ones that are available, use platforms that are most of the time outdated and difficult to be maintained which increases the difficulty of being used by users outside of the specific community. A basic requirement for designing a story understanding system is that of explaining its reasoning process in a way that humans are able to comprehend it. This will become more clear as readers move to Chapter 4 where we present a web platform for story understanding.

In terms of commonsense knowledge acquisition, hand coding of knowledge looks the most promising in terms of quality, but it is slow and resource intensive. Methods for automatic extraction of knowledge such as the ones presented in Section 2.4.1 produce large amounts of factual knowledge, but with a lot of noise and inconsistencies (Kuo and Hsu, 2011). Crowdsourcing approaches are more scalable and give much bigger datasets, but are bounded by low quality contributions and costly development of systems. A major concern of using crowdsourcing is that of the quality of the workers and hence their contributed work. The need to deploy mechanisms for quality control is stressed in the work of Nguyen et al. (2017) and should be taken into account when designing systems for acquiring commonsense knowledge. Moreover, current work on Games With A Purpose for acquiring knowledge is focused on factual knowledge and little to none is oriented towards acquiring commonsense knowledge in the form of rules which leaves space for systems aiming towards that direction. Existing work in GWAPs for knowledge acquisition rely only on player evaluation of the knowledge which does not guarantee its usefulness for automated tasks. In Chapter 5 of this work we present a methodology and implementations of games that address this problem by offering a methodology for evaluating acquired knowledge.

Another factor one should consider while acquiring commonsense knowledge, is that of the cost. In the work of Paulheim (2018) this is discussed, showing that the cost of manually creating knowledge (e.g., the example of Cyc) is between $2 and $6, and that the cost of automatically gathering knowledge is cheaper, i.e., 1 cent to 15 cents per statement. Of course the latter does not take into account the cost of infrastructure (computing power) and software licensing fees (if they exist). Crowdsourcing solutions such as GWAPs also include costs that need to be considered (Chamberlain et al., 2017), including the cost to develop the GWAP, the cost to have someone start to play a game, the cost to acquire a completely annotated item and the average cost to get a player to provide a useful judgment.

What is also important from reviewing relevant approaches, both for story understanding and knowledge acquisition is the need to have an efficient pipeline to acquire knowledge, represent it in an appropriate format, retrieve knowledge, reason with the story and finally present the comprehension model or perform any other story understanding task. Both manual and crowdsourced approaches can be used towards that purpose but our hypothesis is that the exploitation of a hybrid model, i.e., a model which uses both humans and machines towards gathering such knowledge, is more suitable for the task of knowledge acquisition.

During the literature review we have identified systems that move beyond question answering for testing story understanding and use methods such as textual entailment and geographic focus identification. In Chapter 6 we dive deeper in this line of research and we use systems presented in Section 2.2 to compare them with our approach.

In terms of representation, a number of approaches have been tested so far by different researchers, including various logic representations such as event calculus (Kowalski and Sergot, 1989; Miller and Shanahan, 2002) and situation calculus (Reiter, 1991), episodic logic (Schubert and Hwang, 1989) and ontologies. There are also argumentation based representations which are more similar on how humans reason.

Arguing is a prevalent human ability and as such, it is one of the many intelligent tasks humans can perform. If a machine is able to argue, then this machine may probably hold some sort of human intelligence. In the next Chapter, we provide an in depth view of the computational argumentation field and its connections to story understanding, hence our work.

3

Using Argumentation for Knowledge Representation and Reasoning

"For every belief comes either through syllogism or from induction"
– Aristotle - The Organon, *Prior analytics II, Part 23*

In this Chapter, we provide the foundation for the Chapters to follow in terms of representing commonsense knowledge suitable for story understanding. We provide an introduction to argumentation and how argumentation is used for story understanding.

Argumentation is "the action or process of reasoning systematically in support of an idea, action, or theory." (Merriam-Webster Online, 2009). Its historical roots can be traced back to ancient Greeks and more specifically to the writings of Aristotle on proof and persuasion and Plato's dialectics. In his first book (Prior Analytics I), Aristotle describes a syllogism as "an argument (λόγος) in which, certain things being posited, something other than what was laid down results by necessity because these things are so." (Barnes, 1995; Read, 2016). What Aristotle used in his syllogisms, are arguments for supporting the conclusions that they draw. Complex arguments can be built from simpler, basic arguments (Kakas et al., 2016).

The dialectic method (διαλεκτική) supported by Plato (The Republic, 348b) is the discourse between two or more people holding different points of view about a subject but wishing to establish the truth through reasoned arguments.

In the work of Toulmin (2003) a presentation of the general structure of arguments from a philosophical perspective is given. The author stated that "regardless of substantive context, argument could be seen as the offering of a claim together with answers to certain characteristic questions, but that standards for judging the adequacy of arguments are variable from one argument field to another".

According to the Oxford English Dictionary, a logical argument (or argument) is "a process of creating a new statement from one or more existing statements. An argument proceeds from a set of premises to a conclusion, ..., via a procedure called logical inference". An argument can be used as a basis for discussion or reasoning and that is also the reason that the study of argumentation is historically motivated by an interest in the improvement of discourse (Van Eemeren et al., 2015).

In a more recent work, that of Bench-Capon and Dunne (2007), argumentation is defined as "the study of processes concerned with how assertions are proposed, discussed, and resolved in the context of issues upon which several diverging opinions may be held.".

Argumentation is used in our every day life in conversations where we employ arguments for or against a position. It can be used for a range of tasks that include negotiation such as legal reasoning, medical decisions, decision support, social networks and many other applications. The scientific area of argumentation is an interdisciplinary one and it is linked to psychology, philosophy, formal logic and linguistics. At this point we present a short dialogue, which is used as an example of an argumentation process:

`Person A:` I want a raise.
`Person B:` What makes you think that you are worthy of a raise?
`Person A:` I have been working for you for 10 years, I am collaborating well with everyone at the office and according to my last evaluation I was in the top 5 employees of this company.
`Person B:` Yes, but a month ago you had a fight with John at HR.
`Person A:` ...

In the above dialog one can identify the premises (P^i), i.e., the statements that provide reason or support for the conclusion, and the conclusion (C), i.e., a statement in an argument that indicates what an agent is trying to convince another agent. There can be only one conclusion in a single argument, in oppose to premises that can be one or more in a single argument.

The outcome of the argumentation process is the acceptance or not of the conclusion based on the presented premises.

P^1: Work on a company for many years.
P^2: Collaborate well with co-workers.
P^3: Worker has a very good evaluation.
P^4: When you fight with other people at the office you do not collaborate well with co-workers. (implicit)
`C:` Worker gets a raise.

3.1 An Introduction to Argumentation

Argumentation is an appropriate substrate for knowledge representation and reasoning since it fits the human reasoning process, by being able to handle incomplete, inconsistent, and evolving knowledge and it can be used for modeling reasoning types such as persuation, decision making and deliberation (Kakas et al., 2016). There are experimental studies which show that human reasoning is not following the classical logic paradigms (Byrne, 1989; Wason, 1968). More specifically there are empirical evidence that humans are able to do "Modus Ponens" but not "Modus Tollens" (Storring, 1908) and evidence that humans do not reason in a "possible model" but in an "intended model" (Johnson-Laird and Steedman, 1978). Recent work by Saldanha and Kakas (2019) suggests that Argumentation is a cognitively compatible approach for human reasoning and more specifically, for human syllogistic reasoning. Argumentation can be used to reason with incomplete or imperfect information, which involves the formulation of arguments concerning a specific claim. The first step in representing an argumentation scenario is to find a way to represent arguments and the relationships between them.

In literature, there are several methods to identify the steps needed to construct an argumentation model, but we choose to present the one proposed by Atkinson et al. (2017) which is a more generic one and comprises five central building layers: the structural layer where the argument is formed including its internal structure, the relational layer that deals with how arguments are linked to each other (e.g., attack and support relationships), the dialogical layer which handles the rules of exchanging arguments among agents, the assessment layer which deduces the result of the argumentation process (e.g., which arguments are accepted or not) and the rhetorical layer that includes the believability and impact of arguments from the perspective of the audience, use of threats and rewards, appropriateness of advocates, and values of the audience. In short, the argumentation process follows these five steps (Amgoud et al., 2008):

- Construction of arguments,
- Definition of interactions between arguments,
- Valuation of each argument,
- Selection of the most acceptable arguments,
- Conclusion.

In this chapter we present an overview of the area of computational argumentation without resolving to details or proofs, since these are extensively presented in the relevant papers where the work was originally presented. Readers should note that definitions are presented from the original work and only some modifications are made to unify the presentation in this work. We also give examples of how argumentation is used as an appropriate layer to represent knowledge for the rest of this thesis (cf. Chapters 4, 5, 6).

3.1.1 Abstract Argumentation Framework

We first give an overview of the "Abstract Argumentation framework" proposed by Dung (1995). This is a framework that focuses on selecting acceptable (justified) arguments. Arguments are formulated together with an attack relation between them and are handled as an abstract entity, ignoring its internal structure and focusing on the relations between them (e.g., attack relations). Also, possible conflicts between the arguments are resolved on the semantical level. To understand how this framework works we need to examine some of its key definitions.

Definition 1 *An **argumentation framework** is a pair $AF =< AR, ATT >$, where AR is a set of arguments and $ATT \subseteq AR \times AR$ is a binary relation on AR representing an attack relationship between arguments.*

Example 1 *Consider the following argumentation framework $AF =< AR, ATT >$ where $AR = \{a,b,c,d\}$ and $ATT = \{(a,c),(b,c),(c,d)\}$. There exist 4 arguments where a attacks c, b attacks c, c attacks d. This can be represented in a directed graph (cf. Figure 3.1).*

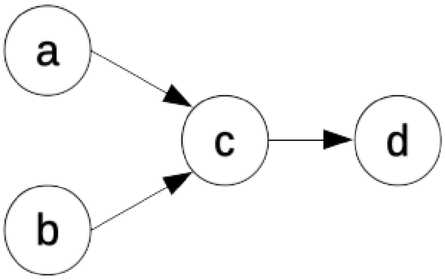


Figure 3.1 A directed graph representation of the abstract argumentation framework described in example 1. Arguments are depicted in circles and arrows represent the attack relation between them.

Definition 2 *$S \subseteq AR$ is **conflict free** if and only if no two arguments in S attack each other.*

Example 2 *The following are conflict free sets of AR*
cfs(AR)=$\{\emptyset,\{a\},\{b\},\{c\},\{d\},\{a,b\},\{a,d\},\{b,d\},\{a,b,d\}\}$.

Dung (1995) provides a number of semantics, called extensions, that allow for a rational agent to decide if an argument can be accepted or not, or if arguments can be accepted together. These extensions include the acceptability of a set of arguments and their admissibility. For a set of arguments to be admissible, they first need to be conflict-free, i.e., no two arguments in the set attack each other.

Definition 3 *An argument a is acceptable with respect to a set of arguments S if every argument* $\in$ *S that is attacked, is defended by another argument* $\in$ *S, i.e., if b attacks a then b is attacked by some argument* $\in$ *S.*

Definition 4 *S is **admissible** if it is conflict-free and all its members are acceptable w.r.t. S.*

Example 3 *The following sets are admissible in AR,*
adm(AR)=$\{\emptyset,\{a\},\{b\},\cancel{\{c}\}\ ,\cancel{\{d\}},\{a,b\},\{a,d\},\{b,d\},\{a,b,d\}\}$. *c is not admissible since it is attacked by a and b and no argument in S attacks a and b. d is not admissible since it is attacked by c and no argument in S attacks c.*

Furthermore, argument acceptability semantics include the preferred extension, the complete extension, the stable extension and the grounded extension.

Definition 5 *A set S* $\subseteq$ *AR is **preferred** in AF, if S is admissible in AF and for each T* $\subseteq$ *AR admissible in T,* $S \nsubseteq T$.

Example 4 *The Preferred extension of AR is the set* $\{a,b,d\}$

There are cases that preferred extensions are not unique and cases where only one preferred extension exists, i.e., the empty set. If an argument is a member of every preferred extension, then this argument is sceptically accepted. If an argument is a member of at least one preferred extension, then this argument is credulously accepted. An argument that is sceptically accepted, is also credulously accepted.

Definition 6 *The **grounded extension** of* $AF =< AR, ATT >$ *is given by the least fixpoint of the operator (the characteristic function)* $F : 2^{AR} \rightarrow 2^{AR}$ *which is defined as* $F(S) = \{a | a \textit{ is acceptable w.r.t. } S\}$

What is important to note, is that the **grounded extension is always unique**.

Example 5 *The grounded extension of AR is the set* $\{a,b,d\}$

Definition 7 *A conflict-free set of arguments S is called a* ***stable extension*** *iff S attacks each argument which does not belong to S.*

Example 6 *The stable extension of AR is the set* $\{a,b,d\}$

Definition 8 *A conflict-free set of arguments S is called a* ***complete extension*** *iff it is admissible and every argument acceptable w.r.t. S is in S.*

Example 7 *The complete extension of AR is the set* $\{a,b,d\}$

Example 8 *Lets also consider this example* $AF =< AR,ATT >$ *where* $AR = \{a,b,c,d\}$ *and* $ATT = \{(a,b),(b,b),(b,c),(c,d),(d,c)\}$. *There exist 4 arguments where a attacks b, b attacks itself and c, c attacks d and d attacks c. This can be represented in a directed graph (cf. Figure 3.2).*

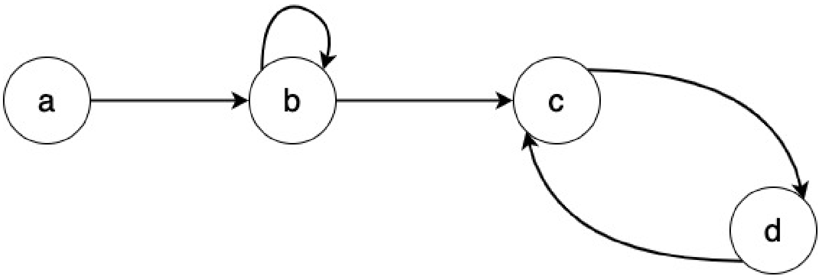


Figure 3.2 A directed graph representation of the argumentation framework in Example 8.

The conflict free sets of AR are: cfs(AR)=$\{\emptyset,\{a\},\{c\},\{d\},\{a,c\},\{a,d\}\}$.
From these, the following sets are admissible in AR: adm(AR)=$\{\emptyset,\{a\},$~~$\{c\}$~~$,\{d\},\{a,c\},\{a,d\}\}$. The preferred extensions of AR are the sets pref(AR)=$\{\{a,c\},\{a,d\}\}$. The grounded extension of AR is the set $\{a\}$, the complete extensions of AR are the sets compl(AR)= $\{\{a\},\{a,c\},\{a,d\}\}$ and the stable extensions are the sets stab(AR)= $\{\{a,c\},\{a,d\}\}$.

A number of systems were developed for calculating the extensions of an argumentation framework, such as ASPARTIX (Egly et al., 2008), ConArg (Bistarelli et al., 2016) and Dung-O-Matic[1]. In the work of Charwat et al. (2015) these are presented in detail. Additionally, there are tools for teaching the argumentation semantics, like ArgTeach (Schulz and Dumitrache, 2016) where the labelling semantics (cf. Section 3.1.2) of abstract argumentation frameworks are presented using an interactive web interface.

[1] https://arg-tech.org/index.php/projects/dung-o-matic/

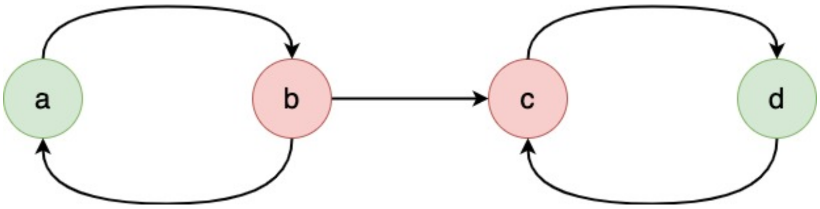


Figure 3.3 An example of a labelling. Arguments a and b are IN (depicted in green) and b and c are out (depicted in red). This labelling is also stable, semi-stable and preferred, but not grounded.

3.1.2 Argumentation Semantics With Labellings

Besides the extensions semantics, there is also the option to deduce the winning arguments of an argumentation framework by using status assignments or labellings (Baroni et al., 2011; Caminada, 2006; Jakobovits and Vermeir, 1999; Verheij, 1996). This is a more expressive method than the extensions method to deduce the acceptance of the arguments. Arguments are labelled as **IN** (accepted argument), **OUT** (rejected argument), or **UNDEC** (undecided argument) using the following labelling rules:

- Label(a)=IN, iff all its attackers are labelled OUT
- Label(a)=OUT, iff at least one of its attackers is labelled IN
- Label(a)=UNDEC, iff at least one of its attackers is labelled UNDEC and none of its attackers is labelled IN

Then the argumentation semantics presented above are used in the following manner:

- **grounded** if it has a minimal set of IN arguments among all complete labellings
- **preferred** if it has a maximal set of IN arguments among all complete labellings
- **semi-stable** if it has a minimal set of UNDEC arguments among all complete labellings
- **stable** if it has no UNDEC arguments

In Figure 3.3 an example of a labelling is depicted, using green circles as IN arguments and red as OUT arguments.

Apart from Dung's abstract argumentation framework, there are a number of frameworks that consider other forms of attacks in arguments, such as abstract bipolar framework, where the notion of support is introduced (Cayrol and Lagasquie-Schiex, 2005), the Abstract Dialectical Frameworks (ADF) (Brewka and Woltran, 2010), a generalization of Dung's framework, where possible relations between arguments include attack, support, and conditional relations, Value-based Argumentation (Bench-Capon, 2003) where arguments are mapped with a value and weighted argumentation where attacks are associated with a weight, indicating the relative strength of the attack (Dunne et al., 2011).

3.1.3 Weighted Abstract Argumentation Framework

In abstract argumentation there is always the possibility to have the empty set as a solution to the acceptability semantics. Even though this is an accepted result, there are cases where we are willing to ignore some of the attacks on the arguments up to a certain threshold so that we get a result set. This is where weighted abstract argumentation (WAF) finds ground, by adding strengths on attacks. In the work of Dunne et al. (2011) the semantics are presented in detail and in the next paragraphs we present the basic notions of this framework.

First, we present the definition of a weighted argumentation framework:

Definition 9 *A* ***weighted argumentation framework*** *is a triple $AF = < AR, ATT, w >$, where AR is a set of arguments, $ATT \subseteq AR \times AR$ is a binary relation on AR representing an attack relationship between arguments and $w : ATT \rightarrow R \geq 0$ is a function assigning real valued weights to attacks.*

Weights can take many meanings such as measures of votes in support of attacks; measures of the inconsistency of argument-pairs; weights as rankings of different types of attack and weights as human evaluations of an attack relation. In the work of Dunne et al. (2011), the first three examples are describes and in Chapter 6 of this work we provide an example of how this was used to enhance the geographic focus identification of a story.

The original Dung semantics are relaxed from the usual notion of conflict-free sets of arguments. In a set S some inconsistencies are tolerated, as long as the sum of the weights of attacks between the arguments of S do not exceed a given inconsistency budget. The set S is admissible in the same manner as with standard Dung semantics and that leads in the same definitions for the rest of the extensions, i.e., stable, grounded and preferred.

Coste-Marquis et al. (2012) propose a different approach to the weight aggregation, and instead of summation of weights they suggest other methods. Moreover, they show how weights can strengthen the usual notion of defence, leading to new concepts of extensions.

There is work on ConArg (Bistarelli and Santini, 2011) which implements the weighted argumentation semantics. ConArg is a tool which relies on constraint programming and is able to handle both AAF semantics and WAF semantics. To add an argumentation graph you need to encode the graph in a special notation. For example, the notation presented in Figure

```
arg(c1).
arg(c2).
arg(a).
arg(b).
arg(c).
arg(d).

att(c1,c2):-1.
att(c2,c1):-1.
att(b,c1):-1.
att(a,c2):-1.
att(c,a):-1.
att(d,a):-2.
```

Figure 3.4 ConArg notation for inserting and drawing a WAF graph. `arg` denotes the arguments and `att` the attacks between two the arguments. Next to each attack there is an integer which denotes the strength of that attack. The highest the value, the stronger the attack.

3.4 will result in drawing the graph in the left site of Figure 3.5. In WAFs we can relax the acceptance semantics by either allowing an internal conflict inside the extensions satisfying a given semantics, or by relaxing defence taking into account the difference between the two weights of attacks (aggregated per attacker) and defence. There are two parameters which influence new semantics: α is the amount of internal conflict that can be tolerated, while γ represents how much defence can be relaxed (Bistarelli et al., 2016). Going back to our example, by setting a value for $\alpha = 1$ and $\gamma = 0$ we receive two sets under the complete, stable and preferred semantics, depicted in the right side of Figure 3.5.

3.2 An Approach to Story Understanding Using Argumentation

Stories and narratives, are special cases of text that have a number of properties. These properties were presented in Chapter 2 and provide a challenge for both representing knowledge and for reasoning with it. Previous work of Diakidoy et al. (2014) suggests that Argumentation can be used for both purposes, as it can overcome the frame, ramification, and qualification problems required by a story comprehension system, as well as the problem of contrapositive reasoning with default information. It is also suitable for constructing and revising compre-

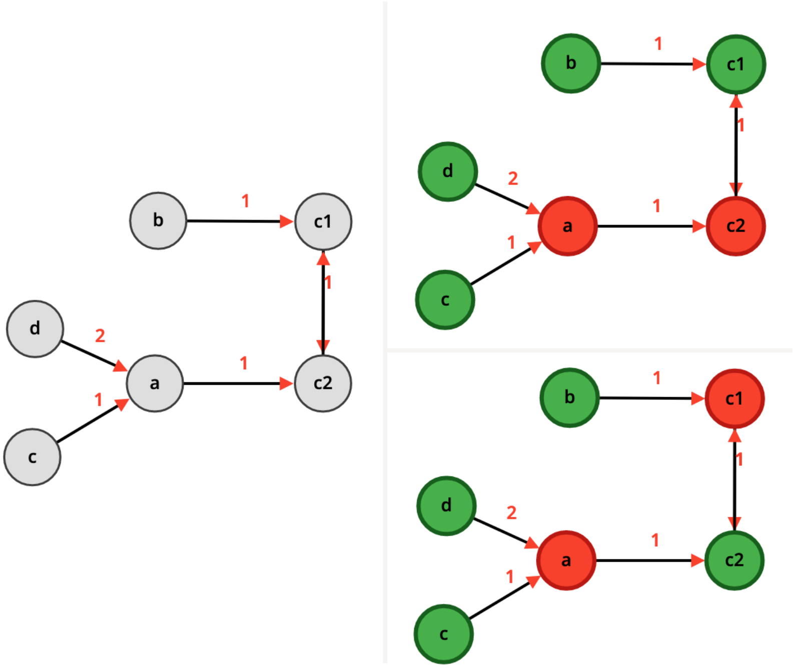


Figure 3.5 An example of a WAF. On the left side the argumentation graph is depicted and on the right side, the two sets of the computed acceptability semantics under the complete, stable and preferred extensions.

hension models using its grounded semantics of admissibility and acceptability of arguments and by combining story specific information with commonsense knowledge.

Argumentation was also proposed as a suitable methodology by Bex and Verheij (2010), depicting how stories and arguments can be used in the context of reasoning with evidence in criminal cases. More specifically, the authors presented how argumentation schemes and story schemes form the most relevant forms of commonsense knowledge in the context of reasoning with evidence.

In the work of Bex and Bench-Capon (2014), a presentation of how argumentation is used to explain how stories can themselves be seen as arguments, using a value based argumentation framework. The authors used the biblical parable of the Good Samaritan to present this approach.

```
Bob called Mary on the phone.

Was Mary embarrassed?
Was the phone ringing?

She did not want to answer the phone.
Bob had asked her for a favor.
She had agreed to do the favor.

Was the phone ringing?

She answered the phone.
She apologized to Bob.

Was Mary embarrassed?
```

Figure 3.6 A short story in natural language with interspersed questions.

3.2.1 The STAR System: An Argumentation-based Reasoning Engine

The STAR: STory comprehension through ARgumentation (Diakidoy et al., 2014, 2015) system adopts the view that comprehension requires the drawing of inferences about states and events that are not explicitly described in the story text (Mueller, 2003) through the use of background world knowledge and commonsense reasoning (Mueller, 2015). Retaining the view that stories and background knowledge are symbolically represented, the STAR system abandons classical logic as the underlying semantics for knowledge, and adopts argumentation (Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008) as a more appropriate substrate for the development of automated systems that interact with humans (Kakas and Michael, 2016; Michael, 2017).

The STAR system is based on the well-established argumentation theory in Artificial Intelligence (Baroni et al., 2011; Bench-Capon and Dunne, 2007), uniformly applied to reason about actions and change in the presence of default background knowledge (Diakidoy et al., 2015). The STAR system follows guidelines from the psychology of comprehension, both for its representation language and for its computational mechanisms for building and revising a comprehension model as the story unfolds.

In terms of its underlying infrastructure, the STAR system is written in SWI-Prolog (Wielemaker et al., 2012). Upon the setting up of the Prolog environment and the invocation of the system, a user-selected domain file is loaded and processed. We present the syntax and semantics of the STAR system through the example story in Figure 3.6.

The example story is interspersed with questions. These are not meant to be parts of the story, but are questions directed towards the reader of the story. Whenever a sequence of questions is encountered, the reader is expected to provide answers to the questions based on the information given in the story in all the preceding story lines. The story then continues until a new sequence of questions is encountered, and so on. Each such part of the story is effectively a *scene* or a reading *session*, and each session is associated with the questions that need to be answered based on the information provided in that and all preceding story sessions. Although the reader goes through the story in the linear fashion in which the story is represented, the story time need not be linear, and can jump back and forth between different time periods. Questions are assumed to refer to a story time-point following the one at which the last story session left off.

As the story unfolds, answers to questions might change either because the same question is asked at a different point in the story time-line, or because the story information leads the reader to revise their *comprehension model* of what they infer (based on their background knowledge and the given story information) to be the case in the story world. The two questions in the example have their answers changed as a reader progresses from the top to the bottom of the story, with the question "Was the phone ringing?" changing because of the first of the aforementioned reasons, and the question "Was Mary embarrassed?" changing because of the second of the aforementioned reasons.

Having explained how a reader may comprehend our example story, we hasten to note that the STAR system *does not* process stories in natural language — in fact, processing stories in natural language is one of the main features of the web-based IDE that is presented in Chapter 4. Instead, the STAR system expects the story statements, their partitioning into sessions, and their association with questions to be provided in a certain symbolic language. All these elements constitute the first part of the domain file that the STAR system loads once it is invoked. A possible representation (although by no means the only one) of our example story is given in Figure 3.7.

Each of the story statements is of the form `s(N) :: Literal at Time-Point`, where `N` is a non-negative integer representing the session of that statement; session `0` is a special session that includes typing information only. A literal `Literal` is either a concept `Concept` or its negation `-Concept` (i.e., the symbol for negation is "-"), where a concept `Concept` is a predicate name along with associated variables or constants for the predicate's arguments. The representation of our example story clearly shows its non-linear time-line.

Following the story statements are the question statements of the form `q(N) ?? Literal at Time-Point; Literal at Time-Point; ...`, where `N` is a non-negative integer representing the number of the question and ";" separates the possible answers to that question;

```
session(s(0),[],all).
session(s(1),[q(1),q(2)],all).
session(s(2),[q(3)],all).
session(s(3),[q(4)],all).

s(0) :: is_favor(favor1) at always.
s(0) :: is_person(bob) at always.
s(0) :: is_person(mary) at always.
s(0) :: is_phone(phone1) at always.

s(1) :: call(bob, mary, phone1) at 6.
s(2) :: -do_want(mary,answer(phone1)) at 12.
s(2) :: have_ask(bob, mary, favor1) at 2.
s(2) :: have_agreed(mary,do(favor1)) at 4.
s(3) :: answer(mary, phone1) at 16.
s(3) :: apologize(mary, bob) at 18.

q(1) ?? is_embarrassed(mary) at 8.
q(2) ?? is_ringing(phone1) at 10.
q(3) ?? is_ringing(phone1) at 14.
q(4) ?? is_embarrassed(mary) at 20.
```

Figure 3.7 A possible representation of the example story depicted in Figure 3.6 to the STAR syntax.

although the notation is meant to represent multiple-choice questions, in effect the STAR system treats each of the choices as a true/false question. Which questions are associated with which sessions is given by the session statements.

Given the story and question representation in Figure 3.7, the STAR system aims to produce a comprehension model of the story, through which it will subsequently attempt to answer the posed questions. Much like human readers, the STAR system invokes background knowledge about the story world to infer what else holds beyond what is explicitly stated in the story. This background knowledge is also represented in a logic-based language, and constitutes the second part of the domain file. For our example story, and in a manner consistent with our chosen symbolic representation of that story, a possible representation of (some of) the background knowledge relevant for the story is given in Figure 3.8.

The presented representation includes four type of statements[2]: a list of concepts that are marked as *fluents*, indicating that their truth value persists across the story time-line;

[2] The STAR syntax allows additional types of statements and expressivity, which we do not present here for simplicity.

```
fluents([
   do_want(_,_),
   is_embarrassed(_),
   carried_out(_),
   has_asked_for(_,_,_),
   has_agreed_to(_,_),
   is_ringing(_)
]).

c(01) :: have_ask(P1,P2,S) causes has_asked_for(P1,P2,S).
c(02) :: have_agreed(P2,do(S)) causes has_agreed_to(P2,S).

p(11) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), apologize(P2,P1)
         implies -carried_out(S).

c(21) :: have_agreed(P2,do(S)), -carried_out(S) causes is_embarrassed(P2).

c(31) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), -carried_out(S),
         call(P1,P2,D), is_phone(D) causes -do_want(P2,answer(D)).

c(41) :: is_person(P1),is_person(P2),call(P1,P2,D),is_phone(D) causes
         is_ringing(D).

c(42) :: is_person(P1),answer(P1,D),is_phone(D) causes -is_ringing(D).

c(42) >> c(41).
```

Figure 3.8 A possible representation of the background knowledge for comprehending the story depicted in Figure 3.6, assuming its encoding in Figure 3.7

.

rules prefixed by the symbols `c(N)` and `p(N)` to indicate that they are, respectively, causal or property rules, and priorities » indicating relative strength between conflicting rules.

The main part of a rule is of the form `Body causes / implies Head.`, where `Body` is either the tautology `true`, or a comma-separated list of literals, and `Head` is a single literal. Each rule, then, expresses an implication from the premises in its body to the conclusion in its head. The difference between the causal and property rules lies in their treatment of time. Property rules are meant to capture dependencies between the properties of entities, and refer to any single point in the story time-line: whenever the body holds, the head also holds at that same time-point. On the other hand, causal rules are meant to express how things change over time, and capture dependencies between consecutive time points: whenever the body holds, the head holds at the following time-point. Further, when the head literal of a causal rule is inferred, this inference *causes* the persistence of the truth-value of that literal from earlier time points to stop in case the persisted truth-value conflicts with the inference. In effect, the fluent list expresses implicitly a third type of persistence rules, along with the implicit lower priority compared to all conflicting causal rules. Additional priorities between causal and/or property rules are expressed explicitly.

We will not discuss in detail the intuitive interpretation of the rules in the background knowledge for our example story, other than to say that they roughly capture the knowledge that: if you apologize to someone that has asked you to do something, to which you have agreed, then it is because you have not carried out that something; having agreed to do something that you have not carried out causes embarrassment, and further causes not wanting to answer the phone when the call is from the person that has asked you for that something; a call causes the phone to start ringing, and answering the phone causes the ringing to stop.

With all the aforementioned logic-based information in a domain file, the STAR system proceeds to construct a comprehension model of the story. A comprehension model can be thought as a partial mapping from timed concepts to truth-values, essentially indicating when each concept is true, false, or unknown. In computing these truth-values, one takes into account both the information given explicitly in the story, but also draws inferences through the background knowledge. The STAR system adopts a particular argumentation-based approach to how inferences are drawn. Roughly, it combines story statements with rules to build a proof of the entailment of a literal. Rules are used both in the forward direction (i.e., via modus ponens) and in the backward direction (i.e., via modus tolens). Since different combinations of story statements and rules might lead to contradictory inferences, each constructed proof is viewed as an argument in support of an inference, and conflicts between arguments are resolved by lifting the priority relation between rules to an attacking relation between arguments (Rahwan and Simari, 2009).

Once the grounded extension is computed after each session (with the arguments that are relevant given the premises of the story that far), the STAR system outputs the computed comprehension model, as in Figure 3.9 for our example story.

The output presents the comprehension model (literals that are true at each time-point), with parts of it marked in triangular parentheses to indicate that those come directly from the story and not from inferences. Following the comprehension model, each of the questions (of the session being processed) are presented along with all their choices for answers, and for each answer the system responds on whether it is accepted, rejected, or possible,

```
===================================
>>> Reading story up to scene s(3)
===================================
>>> Universal argument...
>>> Acceptable argument...

>>> Comprehension model:

0: -carried_out(favor1) < is_favor(favor1)> < is_person(bob)>
   < is_person(mary)> < is_phone(phone1)> < is_ringing(ringing1)>
   < is_ringing(ringing2)>

1: -carried_out(favor1) < is_favor(favor1)> < is_person(bob)>
   < is_person(mary)> < is_phone(phone1)> < is_ringing(ringing1)>
   < is_ringing(ringing2)>

2: -carried_out(favor1) < is_favor(favor1)> < is_person(bob)>
   < is_person(mary)> < is_phone(phone1)> < is_ringing(ringing1)>
   < is_ringing(ringing2)> < have_ask(bob,mary,favor1)>

...

19: -carried_out(favor1) is_embarrassed(mary) < is_favor(favor1)>
    < is_person(bob)> < is_person(mary)> < is_phone(phone1)>
    -is_ringing(phone1) < is_ringing(ringing1)> < is_ringing(ringing2)>
    -do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
    -call(bob,bob,phone1) -call(bob,mary,phone1) -call(mary,bob,phone1)
    -call(mary,mary,phone1) has_asked_for(bob,mary,favor1)

20: -carried_out(favor1) is_embarrassed(mary) < is_favor(favor1)>
    < is_person(bob)> < is_person(mary)> < is_phone(phone1)>
    -is_ringing(phone1) < is_ringing(ringing1)> < is_ringing(ringing2)>
    -do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
    has_asked_for(bob,mary,favor1)

>>> Answering question q(4):
+ accepted choice: ,[is_embarrassed(mary)at 20]

>>> Finished reading the story!
```

Figure 3.9 Part of the output of the STAR system for the story depicted in Figure 3.6, as encoded in Figure 3.7 and with the associated background knowledge presented in Figure 3.8.

depending on whether the answer appears affirmatively, appears negatively, or is absent in the comprehension model.

Additionally to the above, the user may also select to present only part of the comprehension model, or have the system present the arguments that it used to support the inferences that led to the comprehension model. Roughly, the user may request to see: the "universal argument" showing all rules that are activated by the story (in all extensions) without regards to conflicts, the "acceptable argument" showing those activated rules that are accepted (in the grounded extension) after the argumentation semantics resolve all conflicts, and details on which rules are qualified ("attacked") by other rules to help the user understand why certain rules did not end up in the acceptable argument. Since the comprehension model is revised from session to session, the user may also see which rules become obsolete and are retracted across sessions (i.e., part of the grounded extension for the preceding but not the current session), and which new rules come into play and are used to elaborate the comprehension model (i.e., part of the grounded extension for the current but not the preceding session). Finally, the user may choose to see how much time the STAR system spends in each part of its computation, and to decide whether the relevant part of the story will be shown along with each session.

3.2.2 STAR Internal Mechanics and Argumentation Semantics

The STAR system adopts a structured rule-based argumentation framework in the spirit of the ASPIC+ framework (Modgil and Prakken, 2014). In the ASPIC+ framework the conflicts between the arguments are resolved with explicit preferences, and arguments are built with both strict and deductive inference rules, whose premises guarantee their conclusion, and defeasible rules, whose premises only create a presumption in favour of their conclusion.

The STAR system use combinations of premises from the story with defeasible rules from the background knowledge to form a proof tree in support of some inference; this tree corresponds to an argument. In this section we depict the argumentation semantics of STAR, as these were initially presented in the work of Diakidoy et al. (2014):

Definition 10 *A story is a triple $S =< N,W,\prec>$ where N is a narrative, W is the world knowledge needed for understanding the narrative, and $\prec$ a priority relation.*

The STAR system uses the notion of an argument-rule, i.e., $arg(H,B)@T^h \overset{\mathrm{d}}{\rightarrow} (C,T)$ where:

- $arg(H,B)$ is a unit-argument comprising H is a fluent or action literal and B is a set of such. Unit-arguments capture the relation between concepts in the language, i.e., if the body B holds, then we have some evidence that the head H holds.

- T^h is the time-point at which the head of the unit-argument head is applied, and
- (C,T) is the conclusion that follows from its application, where C is a fluent or an action literal, and T the time-point at which the literal is inferred to hold.

By taking the story representation SR = $< N, W, \prec >$, the corresponding abstract argumentation framework $AAF =< A^{SR}, Att^{SR} >$.

Definition 11 *A timed literal (C, T) is a* ***supported*** *conclusion of a set A of argument-rules if an observation $OBS(C,T) \in N$, or if (C,T) is the conclusion of an argument-rule in A. A set of argument-rules A is* ***story-grounded*** *if it can be totally ordered so that every (L,T) in the premise of any argument-rule in A is a supported conclusion of the set of argument-rules that precede the aforementioned argument-rule in the chosen ordering of A.*

Definition 12 *An argument in A^{SR} is any story-grounded set of argument-rules. (C,T) is an inference of A if it is a supported conclusion of A.*

Definition 13 *Consider two argument-rules $p_1 = arg_1(H_1,B_1)@T_1^h \xrightarrow{d_1} (C_1,T_1)$ and $p_2 = arg_2(H_2,B_2)@T_2^h \xrightarrow{d_2} (C_2,T_2)$. Then:*

- *p_1 and p_2 are in* ***direct conflict*** *if $C_1 = \neg C_2$, $T_1 = T_2$*
- *p_1 and p_2 are in* ***indirect conflict*** *if $H_1 = \neg H_2$, $T_1^h = T_2^h$.*

Definition 14 *Consider two argument-rules $p_1 = arg_1(H_1,B_1)@T_1^h \xrightarrow{d_1} (C_1,T_1)$ and $p_2 = arg_2(H_2,B_2)@T_2^h \xrightarrow{d_2} (C_2,T_2)$. Then:*

- *p_1 (endogenously) qualifies p_2 if $arg_2(H_2,B_2) \not\succ arg_1(H_1,B_1)$, and either p_1 and p_2 are in direct conflict, or they are in indirect conflict and $d_2 = F$, $d_1 = B$.*
- *If $arg_1(H_1,B_1) \succ arg_2(H_2,B_2)$, then p_1 strongly qualifies p_2; otherwise, p_1 weakly qualifies p_2.*
- *The story (exogenously) qualifies p_2 if $OBS(\neg C,T_2) \in N$.*

The following definition explains the attacking relation between arguments.

Definition 15 *An argument A_1 attacks an argument A_2, and thus $(A_1,A_2) \in Att^{SR}$, if an argument-rule p_1 in A_1 strongly qualifies an argument-rule p_2 in A_2, or p_1 weakly qualifies p_2 and there is no argument-rule p_1' in A_1 that is strongly qualified by an argument-rule p_2' in A_2. Furthermore, the empty argument attacks an argument A_2, and thus $(\emptyset,A_2) \in Att^{SR}$, if the story qualifies an argument-rule in A_2.*

Definition 16 *Given a story SR and the corresponding abstract argumentation framework $AAF =< A^{SR}, Att^{SR} >$, a set of arguments $\Delta \subseteq A^{SR}$ is a comprehension model of SR if Δ is a subset of the (unique) grounded extension of $< A^{SR}, Att^{SR} >$*

Unlike in the ABA framework (Toni, 2014) the premises are assumed to be indefeasible, and (exogenously) attack any argument that supports a contrary inference. Also unlike in the ABA framework, arguments (endogenously) attack each other on the rules they use, not on their premises. An attack comes from the last / head / top rule in the proof tree of an argument, and is directed towards any (possibly internal) rule in the proof tree of another argument. As long as the former rule is not less preferred than the latter rule, the attack is present. The semantics of the attack relation implies, in particular, that a pair of arguments can attack each other. With this attack relation, the STAR system proceeds to compute the grounded extension of the resulting argumentation framework, and offers this unique extension as the comprehension model of the story. Beyond consulting the relevant work for more details (Diakidoy et al., 2014), the interested reader may wish to also consult a more recent work (Michael, 2017), where a similar in spirit argumentation semantics is discussed, without the nuances of temporal reasoning and contrapositive reasoning, and where a case is also made for the learnability of this type of arguments.

3.3 Discussion

In this Chapter we give a short introduction to computational argumentation methods and how these can provide the base for representing knowledge suitable for story understanding. Furthermore, we demonstrate the STAR system which is used extensively in our research presented in Chapters 4 and 5 both for representing knowledge using the argument-rule form and for reasoning.

Work on argumentation and especially abstract argumentation frameworks is also utilized by a system we developed for identifying the geographic focus of stories called GeoMantis. The system and a series of experiments are presented in detail in Chapter 6.

Computational argumentation found many application areas, such as law, personal assistants, decision making and many others where there is not a single level of truth, but rather many conflicting opinions that need to be considered before taking a decision. Story understanding is such a paradigm, where readers coming from different background and experiences perceive a story in different ways.

Moreover, computational argumentation is able to provide explanations to the user on the specific system outcome. The symbolic language used is also human-readable and users can track the reasoning process (e.g., activated rules, accepted arguments, conflicting arguments).

Using Argumentation for Knowledge Representation and Reasoning

The STAR system exposes its internal mechanics using several methods presented in detail in Section 3.2.1. Argumentation is inherently transparent in explaining the process and results of reasoning (Fan and Toni, 2014).

Related Publications:

(1) Christos T. Rodosthenous. Understanding Stories Using Crowdsourced Commonsense Knowledge. Online Handbook of Argumentation for AI, Volume 1, pp. 27–32, 2020.

(2) Christos T. Rodosthenous and Loizos Michael. Web-STAR: A Visual Web-based IDE for a Story Comprehension System. Theory and Practice of Logic Programming, 19(2):317–359, 2019.

4

A Web-Based IDE to Facilitate the Handcrafted Preparation of Knowledge for Story Understanding

> *"I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."*
>
> – Alan Turing, *1950*

4.1 Introduction

Knowledge acquisition requires the human ability to explicitly write what is "in your head" in a way that a third-party can understand it. This is not a trivial task. In cases where the third-party is a machine, this task gets even harder, as machines are not capable to understand natural language as we humans do. Automated text comprehension and story understanding (Mueller, 2006b), were the topics of interest by many researchers across a diverse set of fields, including computer science, artificial intelligence, logic programming, psychology, language learning, narratology, and law. Due to the varying interests each of these fields have on story understanding and their varying skills, it is difficult to have a system that addresses everyone's needs.

We have presented a number of systems in Chapter 2 for story comprehension that rely on a symbolic representation of knowledge. Moreover, we have presented the STAR system which specifically addresses story understanding through argumentation in Chapter 3, Section 3.2.1. The users of these systems can be distinguished in two groups: expert users (e.g., computer scientists, logic programmers, and AI experts), who might be more interested in developing systems for story understanding and are able to encode and read stories and background knowledge in a machine-readable format; non-expert users (e.g., psychologists, language experts, narrators), who might be primarily interested in utilizing existing systems for story understanding, and may prefer to write stories in natural language, to examine the comprehension process and perform experiments, without caring that much about the internal encodings and representations that are used by the automated systems. The distinction that we make between experts and non-experts is not meant to be absolute. Junior computer science students might fit better in the non-expert category, and language experts might be considered experts for the particular task of translating a story into a logic-based representation, even if they lack the skills to handle other parts of a story comprehension system. In any case, the diversity and heterogeneity that exists in terms of expertise in the use of automated story comprehension systems suggests the need for systems with a simple and intuitive interface that allows expert and non-expert users to input and reason with chosen stories, and to trace and debug the comprehension process.

In this chapter, we present our work on building a platform for facilitating both story understanding and knowledge acquisition. We utilize the STAR system (cf. Chapter 3, Section 3.2.1) and build a platform on top of it to endow the end system with the aforementioned characteristics. We present the design and development of the **Web-STAR** platform built on top of the STAR system. The platform includes a web-based integrated development environment (IDE) that presents a personalized environment for each user with tools for writing, comprehending, and debugging stories, while visualizing the output of the comprehension process. The IDE also delivers a community-building tool, where people can share stories, comment, and reuse other community-created stories. Under the same umbrella, a web service is also made available for integrating other systems with the Web-STAR platform. The successful use of web-based IDEs (cf. Section 4.2) both for research and teaching in logic-based systems, fueled our work for the development of a web-based IDE for the STAR system.

Web-STAR allows both expert and non-expert users to write stories and encode them in the internal STAR syntax, offering a number of features. Non-expert users can take advantage of the following features: 1) the automatic conversion of a story from natural language to the STAR syntax, 2) the encoding of background knowledge using a visual representation based on directed graphs, 3) the automatic conversion of the graph to the STAR syntax and vice versa. Non-expert users also benefit from the visual representation of the system output in a time-line format. Expert users benefit from the feature-rich IDE, which allows the preparation of a story in the STAR syntax using a state-of-the-art source code editor, the reasoner debugging options, and the raw output. All users benefit from the collaboration options available and the story repository.

In the following sections, we present the current state of affairs on web-based IDEs that are used in logic-based systems, followed by a presentation of the STAR system as the underlying engine of the Web-STAR platform. Next, the Web-STAR platform is presented with details of the various features that it offers, along with scenarios on how these features can be used. The platform's usability is then evaluated and discussed, and new features and additions to the Web-STAR platform are presented as part of our ongoing work on the platform.

Work on web-based IDEs that are geared towards imperative and declarative (logic-based) languages is presented. Currently, little work has been done to enhance story understanding systems with functionality present in an IDE, and more specifically in a web-based IDE. The lack of a visual online environment makes it harder for non-experts to setup and use these systems without prior programming knowledge and explicit knowledge of the specific system's internal mechanisms and representation. Furthermore, the majority of these systems rely on external tools (text editors) for editing the source code and lack basic functionality that an IDE can easily provide (e.g., code folding, syntax highlighting).

4.2 Web-based IDEs

Web-based IDEs are systems available through a web browser with no reliance on specific hardware or software stack and are agnostic to the Operating System. Some of these are now considered as mainstream IDEs for developing applications, such as AWS Cloud9 (Amazon Web Services Inc. or its affiliates., 2019) delivered by Amazon and tightly integrated to its cloud services, Codiad (Safranski, 2017), ICEcoder (ICEcoder Ltd, 2017), Codeanywhere (Codeanywhere Inc., 2017) and Eclipse Che (Eclipse Foundation, 2017).

These web-based IDEs allow users to write code in an online source code editor using the programming language of their choice. They also provide code folding, code highlighting, and auto-complete functionality, built in their source code editors. Moreover, some of them provide online code execution functionality with access to an underlying virtual machine and thus access to the shell. There are also other smaller web-based IDEs such as JSFiddle[1], used for testing and showcasing user-created and collaborational HTML, CSS and JavaScript code, and the very popular Jupyter notebooks (Kluyver et al., 2016) used for running python code in a browser.

[1] https://jsfiddle.net/

Web-based IDEs are also used in the logic programming domain. There are only a few systems developed to address this need, like SWISH (SWI-Prolog for Sharing) (Wielemaker et al., 2015), IDP Web-IDE (Dasseville and Janssens, 2015), and Answer Set Programming (ASP) specific IDEs and tools, like the system presented in the work of Marcopoulos et al. (2017).

SWISH is a web front-end for SWI-Prolog, and is used to run small Prolog programs for demonstration, experimentation, and education. The platform offers collaborative tools for users to share programs with others, and a chat functionality. An instantiation of the system was also used to build the SWISH DataLab system (Bogaard et al., 2017), which is oriented towards data analysis. Additionally, there is also a web-based implementation (Wielemaker et al., 2019) of the computer language LPS (Logic-based Production System) (Kowalski and Sadri, 2016) built as an extension of SWISH. This system aims at supporting the teaching of computing and logic in educational domains.

The SWISH design is geared towards the educational domain, allowing learners of the Prolog language to easily access code examples and execute them without the need to install SWI-Prolog locally. However, it exposes only a limited subset of the SWI-Prolog language, and it is not recommended for large and real-world applications.

The IDP Web-IDE is an online front-end for Imperative Declarative Programming (IDP), a Knowledge Base System for the FO(·) language. FO(·) is an extension of first-order logic (FO) with types, aggregates, inductive definitions, bounded arithmetic and partial functions (Denecker and Ternovska, 2008). The Web-IDE allows users to open a chapter from the online tutorial and start testing example programs. There are options for collaborative work and visualization functionality for some of the program outputs.

In the work of Marcopoulos et al. (2017), an online system is presented with a cloud file system and a simple interface, which allows users to write logic programs in the SPARC language (Balai et al., 2013) and perform several tasks over the programs. The authors aim to use this system to teach Answer Set Programming to undergraduate university students and high school students.

4.3 Web-STAR: A web-based IDE on Top of STAR

Following the successful paradigm of many other projects that moved to an online environment, and aiming towards increasing the usage of the STAR system from non-expert users, we developed a web-based IDE for STAR. This IDE incorporates all the functionality of the STAR system in a structured web environment with the addition of visualization, automation, and collaboration tools that help users prepare and process their stories.

Moreover, the IDE employs a number of social features for user collaboration, like public code sharing and posting of stories, both in natural and symbolic language, to a public repository. In addition, users can work together using a state-of-the-art collaboration component which allows screen sharing, text and voice chat, and presenter following functionality. In short, work on Web-STAR includes:

- A Web-IDE that does not require setup, it is OS agnostic, and offers modern IDE functionalities.
- A platform for collaboration and educational support.
- A platform for integrating story comprehension functionality to other systems.
- A modular architecture that facilitates the addition of new components and functionality.

Figure 4.1 depicts the architectural diagram of the Web-STAR platform that comprises the Web-STAR IDE, the web services, the STAR system engine, the public repository, and the databases for storing related information. In the next paragraphs, a presentation of the web-based IDE is shown with details of the workspace layout, the components, and the functionality of the IDE. The Web-STAR IDE is available online at http://cognition.ouc.ac.cy/webstar/ and it is accessible from any device.

4.3.1 Getting Started With the Web-STAR Interface

To start using the Web-STAR IDE, a user creates an account and activates the personal workspace. Currently, both local and remote authentication options are available. The local authentication method uses the integrated storage facilities of the platform. The remote authentication method uses the OAuth2 protocol (https://oauth.net/2/), offered by third parties like Facebook, Google, Github, etc. Other authentication methods are supported as long as the appropriate plugin is available.

After the authentication process is completed, the user is redirected to the Web-STAR IDE environment where both the source code editors and the visual editors are present.

4.3.2 The IDE Environment and Workspace

Users are presented with the workspace (cf. Figure 4.2), which is divided into three distinct areas: 1) the story writing area, 2) the background knowledge writing area, and 3) the story comprehension output area. This design was chosen to give users a clear understanding of

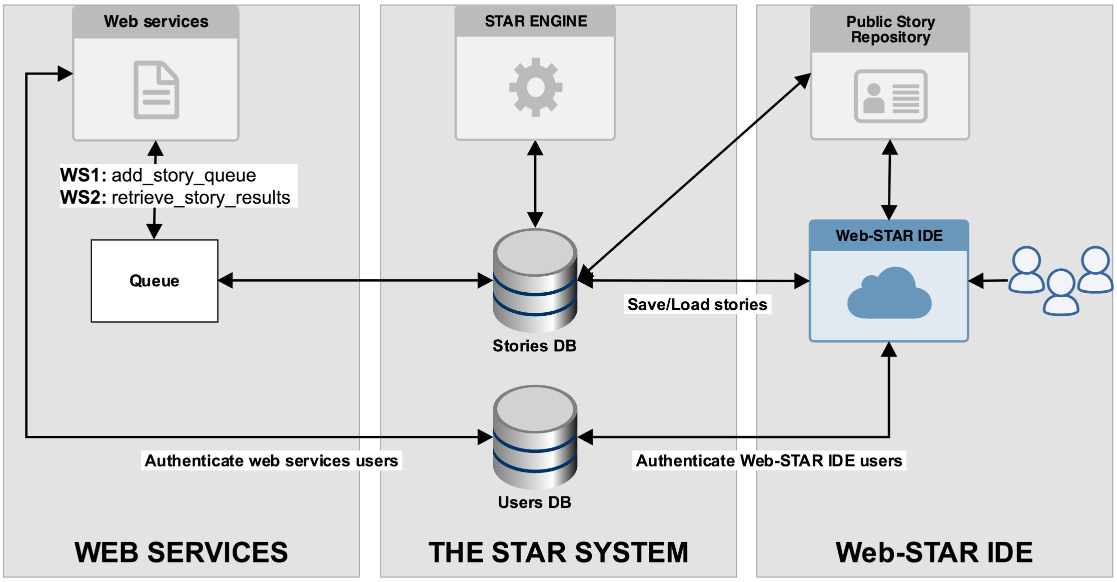


Figure 4.1 The Web-STAR platform architecture with its three core components: the Web-STAR IDE, the STAR system, and the web services infrastructure. The diagram also presents the authentication mechanism, the storage functionality, and the web services provided.

the workflow of the story comprehension process, and to enable users to hide the areas which are not needed, aiming to avoid information overload.

The workspace (cf. Figure 4.2) is also divided into two columns. The left column is for the tasks that do not require users to have prior knowledge and experience in using the STAR system and the right column is for more seasoned users who have prior knowledge of the STAR system semantics and experience in encoding stories using the STAR system. This modular layout allows users to choose the mode they want to use while preparing their stories. More specifically, the web interface comprises three view modes:

- **Simple**: Users write a story in natural language, add background knowledge using the visual editor, and view the visual representation of the story comprehension model. This mode is ideal for users that are new to story understanding systems and want to have an overview of the capabilities and processes of encoding a story.

- **Advanced**: Users write a story in the STAR syntax, encode the background knowledge in the source code editor, and are presented with raw output from the STAR reasoning engine. This mode is ideal for users with prior knowledge in encoding stories in the STAR syntax, and for users who wish to enter pre-encoded stories as done in the standalone version of the STAR system.

- **Mixed**: Users write a story using any of the above options and convert from one mode to the other. For example, users can encode the background knowledge using the source code editor and then convert it to the visual format where they can make further changes. This mode is ideal for users that are learning the system and feel more confident in using the visual components and viewing the conversion. Moreover, this mode is used for teaching, since educators can present examples of encoding the story in visual format and then present the corresponding STAR syntax.

Additionally, users are also able to set the active area in any of the above view modes, i.e., to display only the background knowledge area, the story input area, or the story comprehension output area. The design of the workarea is fully customizable, allowing users to maximize the part of the IDE they are currently working on and minimize the areas that are not needed at that time. Whenever a mode is chosen, the relevant area is resized to maximize the view to the screen size of the device used.

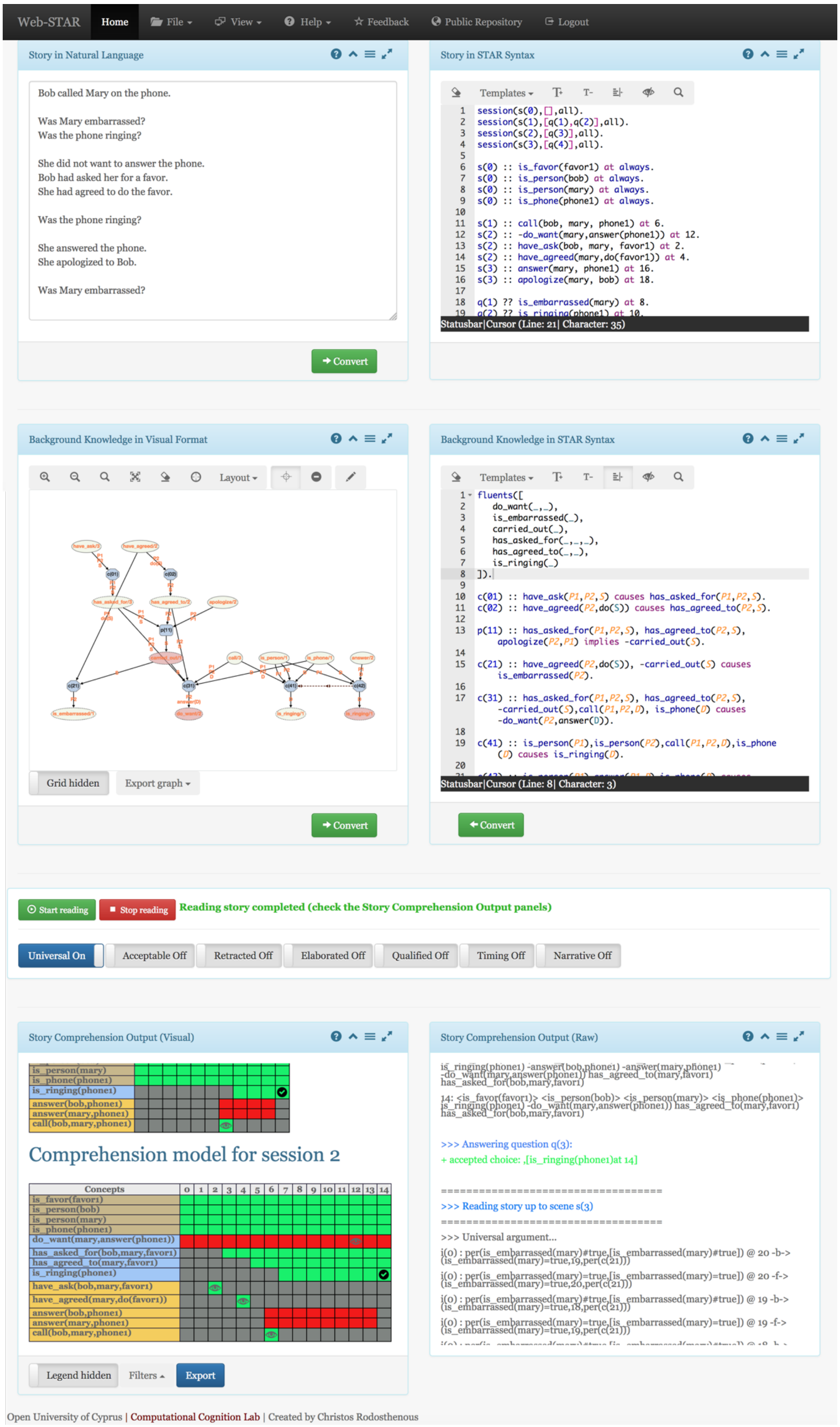
4.3.3 The Story Workarea

Users can start using the system by creating a story from scratch, either in natural language or in the STAR syntax, or even by loading an existing story. More specifically, users can write their code in the source code editor or load it from an external file previously created for the standalone STAR system, load an example file, or load a story file from the public repository. Non-experts can benefit from the example stories and the user-contributed stories in the public repository.

Currently, the source code editor (cf. Figure 4.3) allows syntax highlighting using a STAR syntax highlighter file that inherits the Prolog's syntax template and is expanded with the STAR semantics. Furthermore, line numbering and code wrapping are also available to users along with the extensive "search & replace'" capability for finding text in large stories.

The Web-STAR IDE has a comprehensive list of menu options that enable users to load example story files, study them and edit them. Users have a personal workspace for saving their newly created stories and a public workspace for loading other users' stories. A story that is saved in the personal workspace can only be accessed by its creator, whereas a story stored in the public space is visible to everyone. Options for importing code stored locally on the user's personal device and exporting stories to a file for local processing are also available. This functionality allows a user to use the standalone version of the STAR system to process the story.

When a user loads a story, like the example story presented in Chapter 3, Section 3.2.1, the source code editor immediately identifies and highlights the STAR semantics (variables,



Bob called Mary on the phone.

Was Mary embarrassed?
Was the phone ringing?

She did not want to answer the phone.
Bob had asked her for a favor.
She had agreed to do the favor.

Was the phone ringing?

She answered the phone.
She apologized to Bob.

Was Mary embarrassed?

```
session(s(0),[],all).
session(s(1),[q(1),q(2)],all).
session(s(2),[q(3)],all).
session(s(3),[q(4)],all).

s(0) :: is_favor(favor1) at always.
s(0) :: is_person(bob) at always.
s(0) :: is_person(mary) at always.
s(0) :: is_phone(phone1) at always.

s(1) :: call(bob, mary, phone1) at 6.
s(2) :: -do_want(mary,answer(phone1)) at 12.
s(2) :: have_ask(bob, mary, favor1) at 2.
s(2) :: have_agreed(mary,do(favor1)) at 4.
s(3) :: answer(mary, phone1) at 16.
s(3) :: apologize(mary, bob) at 18.

q(1) ?? is_embarrassed(mary) at 8.
q(2) ?? is_ringing(phone1) at 10.
```

Statusbar|Cursor (Line: 21| Character: 35)

Background Knowledge in Visual Format
Layout
Grid hidden
Export graph
➜ Convert
Background Knowledge in STAR Syntax
Templates
T+
T-

```
fluents([
   do_want(_,_),
   is_embarrassed(_),
   carried_out(_),
   has_asked_for(_,_,_),
   has_agreed_to(_,_),
   is_ringing(_)
]).

c(01) :: have_ask(P1,P2,S) causes has_asked_for(P1,P2,S).
c(02) :: have_agreed(P2,do(S)) causes has_agreed_to(P2,S).

p(11) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S),
    apologize(P2,P1) implies -carried_out(S).

c(21) :: have_agreed(P2,do(S)), -carried_out(S) causes
    is_embarrassed(P2).

c(31) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S),
    -carried_out(S),call(P1,P2,D), is_phone(D) causes
    -do_want(P2,answer(D)).

c(41) :: is_person(P1),is_person(P2),call(P1,P2,D),is_phone
    (D) causes is_ringing(D).
```

Statusbar|Cursor (Line: 8| Character: 3)

Reading story completed (check the Story Comprehension Output panels)

is_person(mary)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_phone(phone1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_ringing(phone1)	⬛	⬛	⬛	⬛	⬛	⬛	⬛	🟩	🟩	🟩	✓
answer(bob,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	🟥	🟥	🟥	🟥	⬛
answer(mary,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	🟥	🟥	🟥	🟥	⬛
call(bob,mary,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	👁	⬛	⬛	⬛	⬛

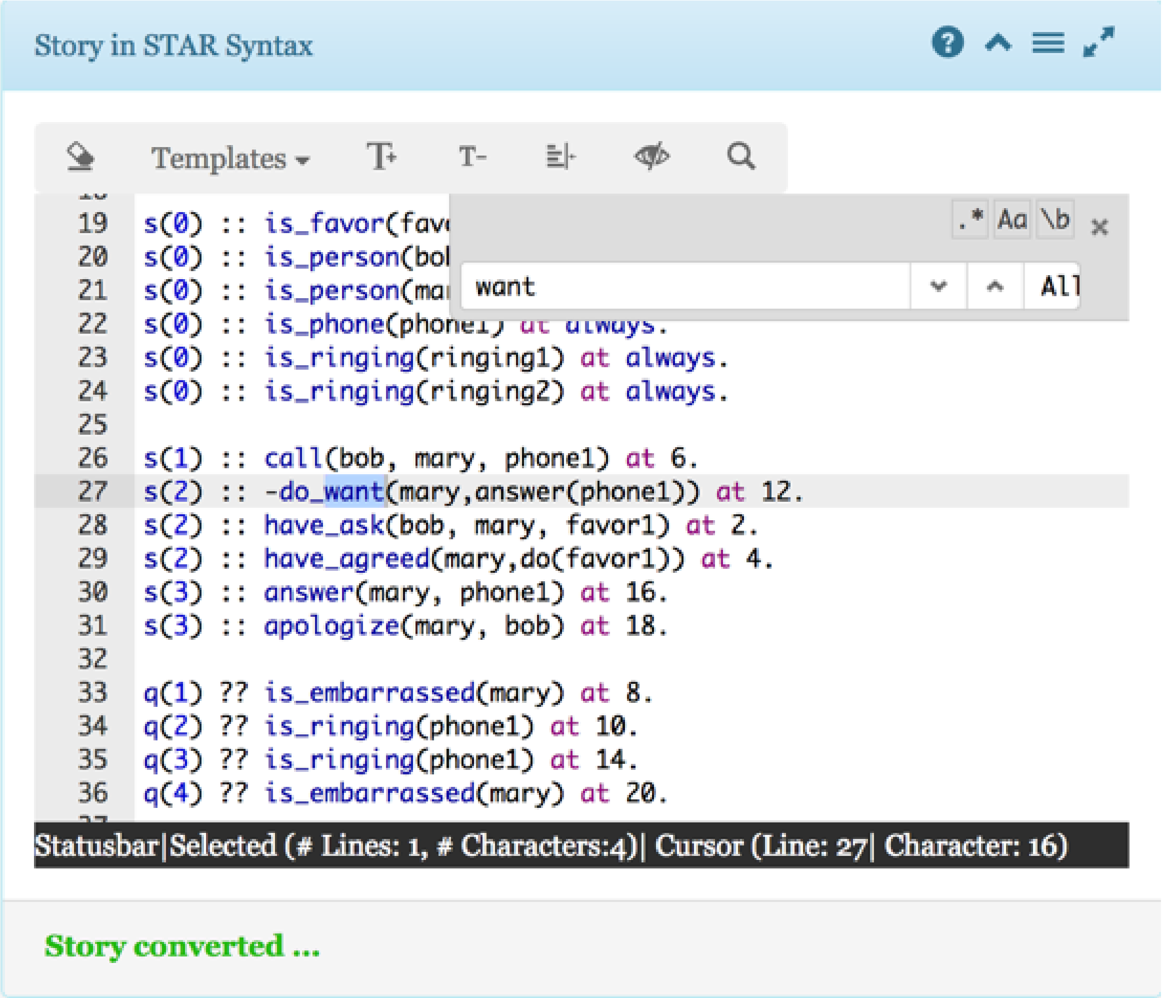
Comprehension model for session 2

Concepts	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
is_favor(favor1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_person(bob)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_person(mary)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_phone(phone1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
do_want(mary,answer(phone1))	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	👁	🟥	🟥
has_asked_for(bob,mary,favor1)	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
has_agreed_to(mary,favor1)	⬛	⬛	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_ringing(phone1)	⬛	⬛	⬛	⬛	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟩	🟩	🟩	✓
have_ask(bob,mary,favor1)	⬛	⬛	👁	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
have_agreed(mary,do(favor1))	⬛	⬛	⬛	⬛	👁	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
answer(bob,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	⬛
answer(mary,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	⬛
call(bob,mary,phone1)	⬛	⬛	⬛	⬛	⬛	⬛	👁	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛

```
is_ringing(phone1) -answer(bob,phone1) -answer(mary,phone1)
-do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
has_asked_for(bob,mary,favor1)

14: <is_favor(favor1)> <is_person(bob)> <is_person(mary)> <is_phone(phone1)>
is_ringing(phone1) -do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
has_asked_for(bob,mary,favor1)


>>> Answering question q(3):
+ accepted choice: ,[is_ringing(phone1)at 14]


======================================
>>> Reading story up to scene s(3)
======================================
>>> Universal argument...
i(0) : per(is_embarrassed(mary)#true,[is_embarrassed(mary)#true]) @ 20 -b->
(is_embarrassed(mary)=true,19,per(c(21)))
i(0) : per(is_embarrassed(mary)=true,[is_embarrassed(mary)=true]) @ 20 -f->
(is_embarrassed(mary)=true,20,per(c(21)))
i(0) : per(is_embarrassed(mary)#true,[is_embarrassed(mary)#true]) @ 19 -b->
(is_embarrassed(mary)=true,18,per(c(21)))
i(0) : per(is_embarrassed(mary)=true,[is_embarrassed(mary)=true]) @ 19 -f->
(is_embarrassed(mary)=true,19,per(c(21)))
```

Open University of Cyprus | Computational Cognition Lab | Created by Christos Rodosthenous

Figure 4.2 A screenshot of the Web-STAR IDE layout. The workarea is divided into two columns: the left column (Simple mode) and the right column (Advanced mode); and three rows: the story area, the background knowledge area, and the story comprehension output area.



```
s(0) :: is_favor(fav
s(0) :: is_person(bo
s(0) :: is_person(ma
s(0) :: is_phone(phone1) at always.
s(0) :: is_ringing(ringing1) at always.
s(0) :: is_ringing(ringing2) at always.

s(1) :: call(bob, mary, phone1) at 6.
s(2) :: -do_want(mary,answer(phone1)) at 12.
s(2) :: have_ask(bob, mary, favor1) at 2.
s(2) :: have_agreed(mary,do(favor1)) at 4.
s(3) :: answer(mary, phone1) at 16.
s(3) :: apologize(mary, bob) at 18.

q(1) ?? is_embarrassed(mary) at 8.
q(2) ?? is_ringing(phone1) at 10.
q(3) ?? is_ringing(phone1) at 14.
q(4) ?? is_embarrassed(mary) at 20.
```

Statusbar|Selected (# Lines: 1, # Characters:4)| Cursor (Line: 27| Character: 16)

Figure 4.3 A screenshot of the source code editor, depicting the line numbering, syntax highlighting and line highlighting functionality. Above the source code editor resides the toolbar menu with the "search & replace" functionality window open. At the bottom of the editor resides the statusbar with information on the selected line and character.

rules, operators) and makes it easier for the user to read the encoded story (cf. Figure 4.3). After studying the file, the user can move to the questions part and can add one or more questions by choosing the "question template'" from the menu. The question template (presented in Chapter 3, Section 3.2.1) is added and the user can add the predicates and time-points at which the question is posed. When changes are made to the example file, the user can save it to the personal workspace using the corresponding menu option.

Natural Language to the STAR Syntax Converter

One of the innovations available to the Web-STAR user is the automated component[2] to convert a story from natural language to the STAR syntax. This is a real-time process, where the story and the questions are written in natural language, and the system processes them using a Natural Language Processing (NLP) system and a custom-built parser that maps processed words and phrases to predicates with their arguments.

More specifically, each sentence is processed using the Stanford CoreNLP (Manning et al., 2014) system for NER (Named Entity Recognition), part-of-speech, lemmas (canonical or base form of the word), basic dependencies, and coreference resolution.

First, the component automatically identifies the sessions or scenes of the story. Sessions are added when a series of statements are followed by a question, or a group of questions. There is always a base session "**Session S(0)**'" where all the constants are represented. For each group of questions, an additional session is created (e.g., the story in Chapter 3, Section 3.2.1, Figure 3.7).

Next, the nouns in each sentence and the named entity types (location, person, organization, money, percent, date, time) are identified to create the concepts that represent constant types. For each named entity, a statement of the form "`is_<EntityType>(<entity>) at always.`" is added to the base session of the story. Personal pronouns are also identified as a `Person` entity. The following is an example of this: `is_person(personX) at always.`, where X is an integer, representing the number of entities with the same name. When a coreference is found, the identified person name is used in the concept.

Predicates are created using the Stanford basic dependencies for extracting textual relations from the text. More specifically, for each sentence, the lemmatized "`ROOT`" is used as the predicate name and the lemmatized text from types "nsubjpass, dobj, nmod:poss, xcomp" is used to create the predicate arguments in lower case.

When a word is characterized with the "aux" or "compound:prt" types, then the predicate name is expanded with a "_" and the new word is added in front of the predicate name. When a word has a "neg" type, then the negation symbol "-" is added in front of the predicate name. Words with types "amod, case, cop, auxpass, aux and compound:prt" that are dependent of the "ROOT" are also appended to the predicate name. For words with types "aux, aux:pass and cop" when the lemma "be" is identified, it is converted to "is".

For adding time-points to the story statements, we start at time-point 2 and form two lists. First a list with statements in past perfect is formed starting from time-point 2 and increasing by two for each statement. Then a list with statements starting from the maximum time-point

[2] The API for this component was developed by Adamos Koumis, a colleague at the Computational Cognition Lab. Web-STAR utilizes this API to convert a story in natural language to the STAR syntax.

of the first list with an increment of two is formed. These two lists are joined and form the story statements (cf. Chapter 3, Figure 3.7).

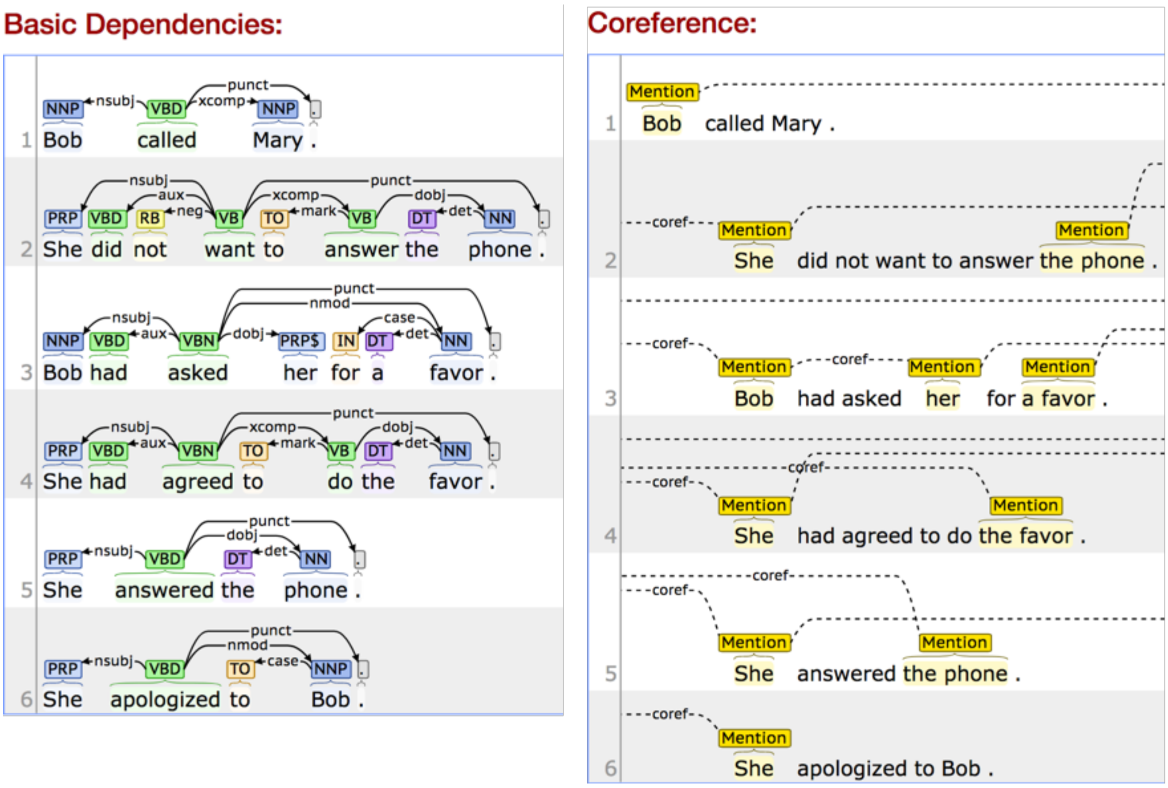
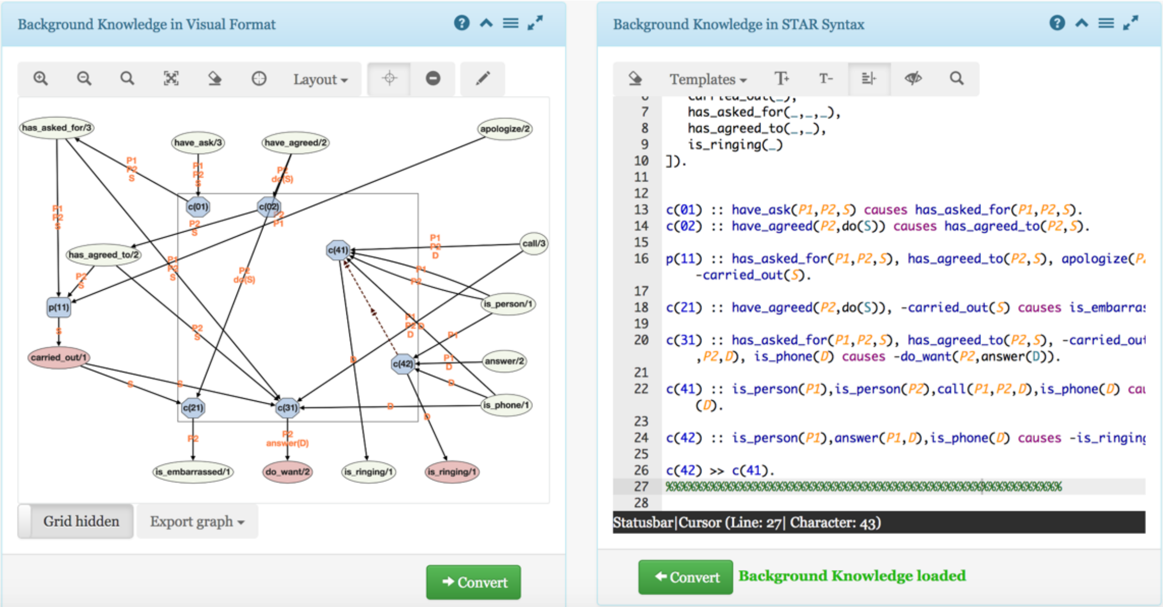

Basic Dependencies:
1 Bob called Mary .
NNP VBD NNP
nsubj xcomp punct
2 She did not want to answer the phone .
PRP VBD RB VB TO VB DT NN
nsubj aux neg xcomp mark dobj det punct
3 Bob had asked her for a favor .
NNP VBD VBN PRP$ IN DT NN
nsubj aux dobj nmod case det punct
4 She had agreed to do the favor .
PRP VBD VBN TO VB DT NN
nsubj aux xcomp mark dobj det punct
5 She answered the phone .
PRP VBD DT NN
nsubj dobj det punct
6 She apologized to Bob .
PRP VBD TO NNP
nsubj nmod case punct
Coreference:
1 Bob called Mary .
2 She did not want to answer the phone .
3 Bob had asked her for a favor .
4 She had agreed to do the favor .
5 She answered the phone .
6 She apologized to Bob .
Mention
coref

Figure 4.4 The output of the CoreNLP processing for the example story in Chapter 3, Figure 3.6. On the left side, the basic dependencies are presented in graphical form and on the right side the coreferences are depicted.

To better understand the conversion process, we take the example of the story in Chapter 3, Figure 3.6, its representation in the STAR syntax (cf. Chapter 3, Figure 3.7) and the output from the Stanford CoreNLP depicted in Figure 4.4. In particular, for the sentence: "She had agreed to do the favor" (sentence 4 in Figure 4.4) we take the ROOT ("agreed") and the dependent words "She","do" and form the predicate `agree (She,do)`. Next, the "ROOT" is connected with an "aux" type with the word "had" and the predicate name is updated accordingly `have_agreed(she,do)`. The word "She" refers to "mary" (using the coreference parsing) and "do" is connected with an "xcomp" relation with the `ROOT`, so "do" will form a new predicate `do(favor1)` and the final concept will become `have_agreed(mary,do(favor1))`.



```
    carried_out(_),
    has_asked_for(_,_,_),
    has_agreed_to(_,_),
    is_ringing(_)
]).


c(01) :: have_ask(P1,P2,S) causes has_asked_for(P1,P2,S).
c(02) :: have_agreed(P2,do(S)) causes has_agreed_to(P2,S).

p(11) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), apologize(P
     -carried_out(S).

c(21) :: have_agreed(P2,do(S)), -carried_out(S) causes is_embarra

c(31) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), -carried_ou
     ,P2,D), is_phone(D) causes -do_want(P2,answer(D)).

c(41) :: is_person(P1),is_person(P2),call(P1,P2,D),is_phone(D) ca
     (D).

c(42) :: is_person(P1),answer(P1,D),is_phone(D) causes -is_ringin

c(42) >> c(41).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Statusbar|Cursor (Line: 27| Character: 43)

Figure 4.5 The background knowledge workarea of the IDE. The visual editor is depicted on the left side of the screenshot, and the source code editor on the right side. There are conversion buttons from one form to the other at the bottom of each panel.

4.3.4 The Background Knowledge Workarea

The next step in preparing the story is the encoding of the background knowledge. This can be done either by using the source code editor or the visual editor (cf. the left side of Figure 4.5). The visual editor uses a directed graph to represent rules. Users are able to see how rules build on each other (i.e., rules whose body literals are the head literals of other rules) and better understand the reasoning process. Moreover, users can focus on specific rules and literals and understand their role in forming the comprehension model of the story.

In particular, each rule is represented with a blue-colored node of octagonal shape for causal rules, and of a rounded orthogonal shape for property rules. Literals are represented with nodes of a cyclical shape, and are green or red to indicate that the literal is, respectively, positive or negative. Literals with a directed edge towards a rule node represent body literals for that rule, and the single literal with a directed edge from a rule node represents the head literal for that rule.

Each node is labeled with the rule's or the predicate's name. For literals, the name is created using the predicate's name and the arity of the predicate (e.g., the predicate "have ask" with three arguments is represented as `have_ask/3`). Arguments are represented with labels on the edges connecting the literal nodes with the rule node (cf. orange labels on edges in Figure 4.6).

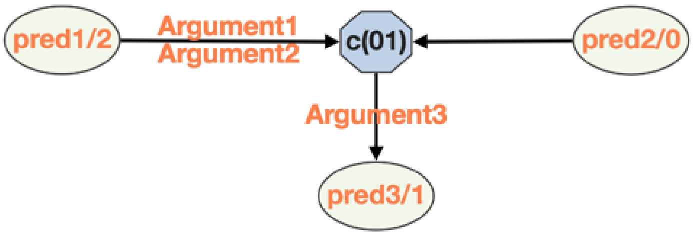


Figure 4.6 A visual representation of a background knowledge rule.

Users can choose to create the entire background knowledge using the tools of the visual editor. Using the “edit” button, users can add literals, rules, edges, and priorities between rules. More specifically, users choose the desired element and click on the white area of the graph, the “canvas”. When a rule is added, the label is automatically set to create a unique name (e.g., `c01, c02, p01, p02 ...`). When literals are added, the user is asked to provide the literal’s name, arity, and polarity (positive or negative). Users connect literals with rules using the “edge drawing tool”, and can set priorities between rules by drawing a “dashed edge” from one rule node to another. Adding and updating arguments to literals is performed by clicking on the connecting edges between the literal and the rule. A dialog box appears for typing each argument.

For every input (textual or visual) and every action on the canvas that does not conform with the STAR syntax (e.g., having two literals in the head of a rule) a guidance message (not simply an error message) is shown, which explains to the user in a visual way (e.g., by highlighting nodes or edges) what needs to be changed to lift the error. This is helpful in teaching scenarios, where students get to know the environment and the basic semantics of the STAR system (or logic-based programming, more generally).

In cases of stories with a large background knowledge, a user can group rules together and minimize or maximize the view of individual groups, isolating the part of the background knowledge that the user wants to inspect. To support this, a type of code folding functionality is implemented, which allows users to focus on a specific subset of the rules on the screen. In Figure 4.7 the code folding/unfolding capability of the IDE is presented.

Moreover, users can zoom in and out of the graph and can change its layout dynamically. There are a number of available layouts for users to choose from, such as “the circle layout” where nodes are put in a circle, “the breadthfirst layout” where nodes are put in a hierarchy based on a breadthfirst traversal of the graph, etc. Furthermore, users can search for a rule, literal or argument using the search tool. When the element is found, it is maximized and focused along with its neighboring elements. There is also an option for fitting the graph to

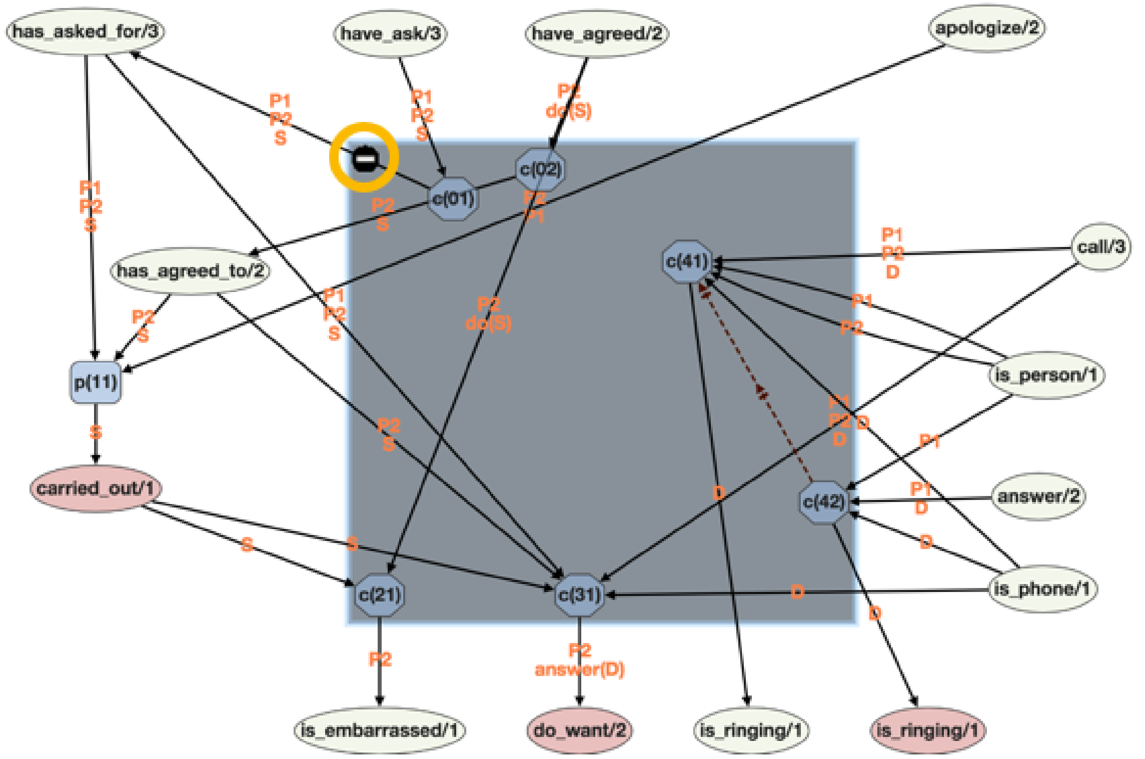


Figure 4.7 A screenshot of the folding/unfolding code capability of the Web-STAR IDE. Grouped rules can be maximized and minimized by clicking at the top left corner of the grouping.

the screen, as well as a “graph navigator” option that allows users to have a bird’s eye view of the whole graph and navigate to the desired part of it.

The Web-STAR IDE allows filtering out elements of the graph that are not needed for a specific job. For example, users can choose to “toggle” the visibility of causal or property rules, priorities, rules with low density or rules that are not connected with other rules. This functionality is part of the Web-STAR’s IDE ability to handle large background knowledge bases with rules.

The background knowledge graph can be exported in various formats, including image formats (png, jpg), JSON, and GraphML (Brandes et al., 2013), which can subsequently be used with third-party applications to present or process the graph and its data.

Expert users can use the source code editor in parallel with the visual one. A similar in look and feel editor with the one for preparing stories is available, and users can take advantage of the included templates for adding rules.

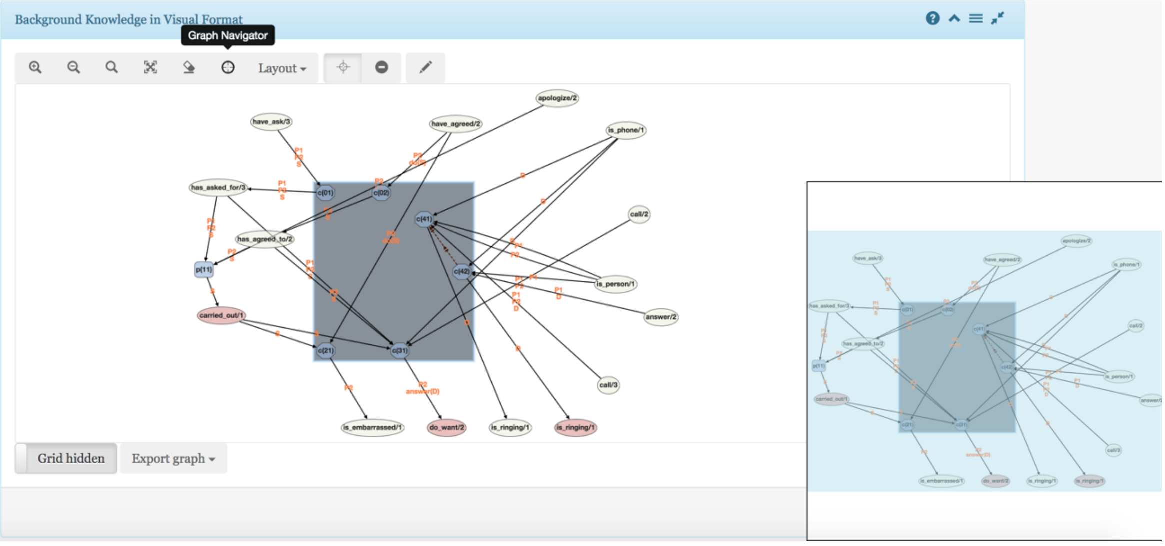


Figure 4.8 A screenshot of the “graph navigator” window (bottom right), where users can have a bird’s eye view of the background knowledge graph and navigate to the desired part of it.

Converting Background Knowledge From Visual to Textual Format

Converting background knowledge from visual to textual format and vice versa, is performed with the click of a button, allowing the user to encode parts of the knowledge in the one format or in the other. Web-STAR’s internal mechanisms read each graph element and perform the conversion of visual background knowledge rules to STAR format. Details on this process are provided in Algorithm 1.

4.3.5 Story Comprehension Process and Output

After completing the story preparation and the background knowledge encoding, users can proceed with the story comprehension process. Users can click the “Start reading” button and immediately see results coming from the STAR system in the “Story Comprehension Output” area in real-time. A number of reporting options can activate and expose the internal processes of the STAR system, including the argumentation mechanism applied for story comprehension, for debugging or educational purposes. In particular, users can choose to view all arguments (`Universal`), the subset of acceptable arguments (`Acceptable`), arguments removed during a specific session (`Retracted`), arguments added during a specific session (`Elaborated`), and information about which arguments qualify other arguments (`Qualified`).

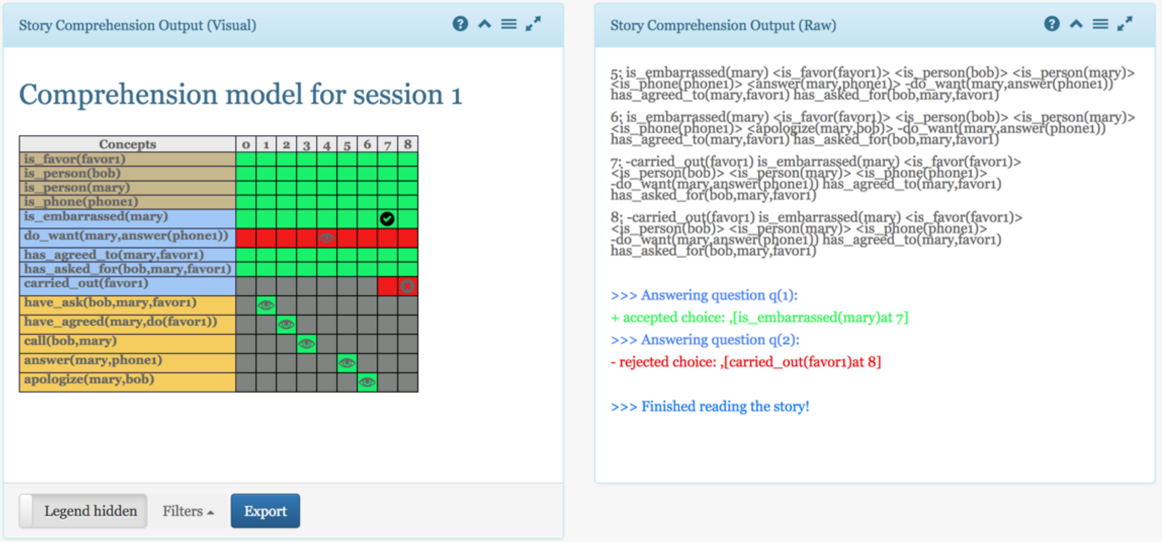
Algorithm 1 Convert background knowledge graph to the STAR syntax

```
procedure CONVERT_GRAPH_TO_STAR(GRAPH_OBJECT)
    % Get all the rule nodes of the graph object
    for Each node $i$ do
        %Get the edge directed outwards of the node (Head)
        for Each edge $j_1$ do
            literal=edge[$j_1$].connected_node
            head=convert_predicate_star(literal,edge[$j_1$].label)
            push_to_list_of_head_literals(head)
        end for
        %Get all the edges directed towards the node (Body)
        for Each edge $j_2$ do
            literal=edge[$j_2$].connected_node
            body=convert_predicate_star(literal,edge[$j_2$].label)
            push_to_list_of_body_literals(body)
        end for
        %Proceed and create the textual representation of the rule
        if node[$i$].type===“property” then
            rule_type=“implies”
        else
            rule_type=“causes”
        end if
        rule=node[$i$].name :: node[$i$].body_literals rule_ type node[$i$].head_literal
    end for
end procedure
```

When the reading process is completed, users can view both the comprehension model and the answers to questions posed. This can be done both in a visual and textual format. The visual output might be preferred for tracking each concept across the story time-line, and the textual output might be preferred for debugging. Each panel is dynamically updated when new information is sent from the STAR system.

In Figure 4.9, the visual output of the comprehension model is depicted, presenting the state of each concept at each time-point. Green, red, and dark grey represent concepts whose value is, respectively, positive, negative, or unknown at that time-point

The magnifying glass, marks concepts whose value is observed at that time-point, i.e., they are extracted from the narrative directly. Concepts with orange background, represent an instantaneous action. Concepts with light blue background represent a persisting fluent and concepts with purple background represent a constant type (e.g., `person(bob) at always.`).


Story Comprehension Output (Visual)
Story Comprehension Output (Raw)
Legend hidden
Filters ▴
Export

Comprehension model for session 1

Concepts	0	1	2	3	4	5	6	7	8
is_favor(favor1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_person(bob)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_person(mary)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_phone(phone1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
is_embarrassed(mary)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	✓	🟩
do_want(mary,answer(phone1))	🟥	🟥	🟥	🟥	👁	🟥	🟥	🟥	🟥
has_agreed_to(mary,favor1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
has_asked_for(bob,mary,favor1)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
carried_out(favor1)	■	■	■	■	■	■	■	🟥	☒
have_ask(bob,mary,favor1)	■	👁	■	■	■	■	■	■	■
have_agreed(mary,do(favor1))	■	■	👁	■	■	■	■	■	■
call(bob,mary)	■	■	■	👁	■	■	■	■	■
answer(mary,phone1)	■	■	■	■	■	👁	■	■	■
apologize(mary,bob)	■	■	■	■	■	■	👁	■	■

```
5; is_embarrassed(mary) <is_favor(favor1)> <is_person(bob)> <is_person(mary)>
<is_phone(phone1)> <answer(mary,phone1)> -do_want(mary,answer(phone1))
has_agreed_to(mary,favor1) has_asked_for(bob,mary,favor1)

6; is_embarrassed(mary) <is_favor(favor1)> <is_person(bob)> <is_person(mary)>
<is_phone(phone1)> <apologize(mary,bob)> -do_want(mary,answer(phone1))
has_agreed_to(mary,favor1) has_asked_for(bob,mary,favor1)

7; -carried_out(favor1) is_embarrassed(mary) <is_favor(favor1)>
<is_person(bob)> <is_person(mary)> <is_phone(phone1)>
-do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
has_asked_for(bob,mary,favor1)

8; -carried_out(favor1) is_embarrassed(mary) <is_favor(favor1)>
<is_person(bob)> <is_person(mary)> <is_phone(phone1)>
-do_want(mary,answer(phone1)) has_agreed_to(mary,favor1)
has_asked_for(bob,mary,favor1)


>>> Answering question q(1):
+ accepted choice: ,[is_embarrassed(mary)at 7]
>>> Answering question q(2):
- rejected choice: ,[carried_out(favor1)at 8]

>>> Finished reading the story!
```

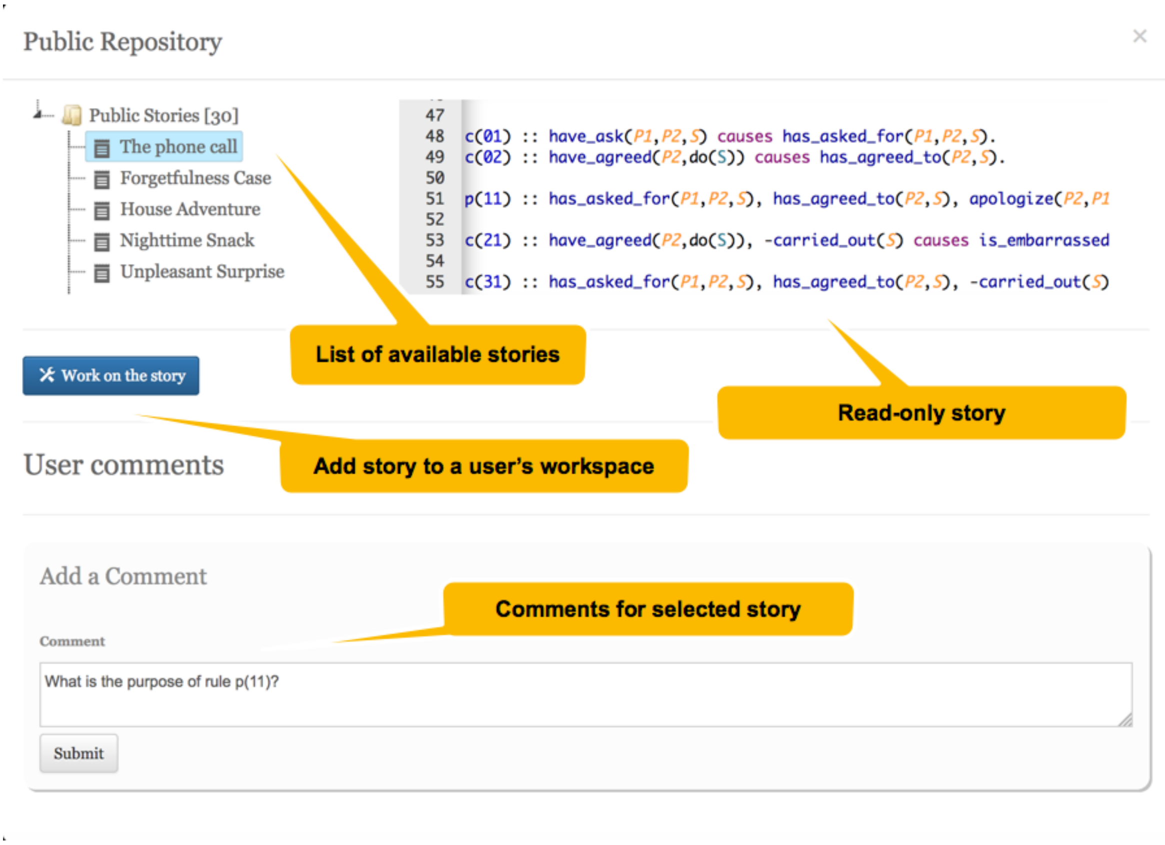
Figure 4.9 A screenshot of the “Story comprehension output” workarea of the IDE, depicting the comprehension model. The legend above the comprehension model provides details on the meaning of symbols and colors in the visual representation of the model and its visibility can be toggled by using the relevant switch. On the right side, the raw output for the same story is presented.

Users can apply filters on the output of the comprehension model and focus their attention of particular concepts. They can choose, for instance, to filter out fluents, actions, and constants, or to view only concepts whose value changes through time, concepts that have a high frequency in the background knowledge, or even concepts that are part of causal rules. The latter are a good indication of the focus of the story and its parts that are most interesting to a reader (Goldman et al., 1999).

The model can be exported in various formats and can be used for educational purposes. The Web-STAR IDE has also a textual format of the story comprehension output that presents the raw output of the STAR system as it would appear when executed as a standalone application. This output is enhanced with color highlighting to identify questions, positive or negative answers to questions, and debugging messages.

Collaboration and “social” Options

Apart from the typical IDE functionality, Web-STAR IDE also provides functionality for sharing publicly a story with other users. By clicking the “Share it” button, a story is added to the public stories repository (cf. Figure 4.10) and appears in the “public stories” tab in



```
c(01) :: have_ask(P1,P2,S) causes has_asked_for(P1,P2,S).
c(02) :: have_agreed(P2,do(S)) causes has_agreed_to(P2,S).

p(11) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), apologize(P2,P1

c(21) :: have_agreed(P2,do(S)), -carried_out(S) causes is_embarrassed

c(31) :: has_asked_for(P1,P2,S), has_agreed_to(P2,S), -carried_out(S)
```

What is the purpose of rule p(11)?

Figure 4.10 A screenshot of the Public Stories Repository. Users can add comments on stories and ask questions. If interested, they can start working on a story by copying it to their personal workspace.

the story browser dialog. Users can read shared stories and add comments, supporting the education of new users from more expert ones.

Beyond sharing, users can collaboratively write a story using the collaboration functionality provided. The system produces a link that can be sent to anyone interested in collaborating for a specific session. The recipient of the link can see the screen and the mouse pointer of each participating user, and changes of content in real-time, while also being able to chat through text and audio. This setting enables teams to collaborate on preparing a story, and allows students to learn by working together on class projects.

User Support and Feedback

The Web-STAR IDE offers a number of features to help its users achieve their goals. Firstly, users can follow a guided tour through the Web-STAR IDE features. Users can then start testing the functionality of the platform with the examples available in the story browser. These examples were carefully crafted for teaching the STAR semantics.

Moreover, in each panel there is an online help option, for guiding users to the specific functionality available for that panel. The icons and graphics chosen for buttons and toolbars are inline with the ones users are familiar in other IDEs. In cases where some users are not aware of the meaning of an icon, a tooltip is available.

To allow users to provide feedback on new desired functionalities or encountered problems, Web-STAR offers a built-in feedback functionality that stores a user's message in the platform's database and alerts the developers through email.

4.3.6 Technical Details and Challenges

For designing and implementing the Web-STAR IDE, we chose to use technologies that are mature, do not require license fees, have a large community of contributors, and can be deployed easily. Furthermore, all technologies used are available as free and open source software, and their communities release frequent updates and new capabilities in each new release. These considerations are important for a project that seeks to be expandable, scalable, and easy to maintain.

The system is based on PHP for backend operations, on the MariaDB database for the data storage, and on the JQuery JavaScript library for the front-end design. Behind this infrastructure lies the STAR system (Diakidoy et al., 2015) and the SWI-Prolog (Wielemaker et al., 2012) interpreter. A wrapper is employed for sending the story file from the front-end to the back-end and returning the results in real-time from the Prolog interpreter using the HTML5 "Server-Sent Events" functionality to dynamically update the interface.

All data storage is handled with the MariaDB database. In particular, a number of tables are used for storing user data, user profiles, and the STAR web service queue.

For the interface design, the Bootstrap framework is used. Bootstrap is an HTML, CSS, and JS framework for developing responsive projects on the web. This framework has a number of ready-to-use components like buttons, panels, toolbars, etc., and is also supported by a large community that develops extra components. The JQuery library (https://jquery.com/) is used to add intuitive UI components and AJAX functionality.

Collaboration functionality is provided using both AJAX components for sharing and commenting on stories, and the TogetherJS library (https://togetherjs.com/). TogetherJS is a JavaScript library from Mozilla that uses the Web RTC (Johnston and Burnett, 2012) technology to enhance communication. It provides audio and chat capabilities between users, and allows users to see each other's mouse cursors and clicks, and the screen content.

The source code editor is based on the ACE editor (https://ace.c9.io), an open source web editor which is used by many other popular cloud IDEs. This editor was chosen because of its maturity, its open source license, and for its popularity. ACE is a code editor written in JavaScript and includes features like syntax highlighting, theming, automatic indent and outdent of code, search and replace with regular expressions, tab editing, drag-drop functionality, line wrapping, and code folding. Moreover, this editor can handle huge documents with more than one million lines of code.

For the visualisation of the background knowledge, we sought a component that is able to represent rules in a graph format and can additionally allow interaction with the user and the graph elements. For that reason, Cytoscape.js (Franz et al., 2016) was selected, which is an open source JavaScript-based graph library that allows users to interact with the graph, supports both desktop browsers and mobile browsers. It can also handle user events on graph elements like clicking, tapping, dragging, etc. This library also provides a large number of extensions that are employed to enhance the functionality of the Web-STAR IDE. The code folding/unfolding capability uses the "expand-collapse" extension[3], which provides an API for expanding and collapsing compound parent nodes on a cytoscape graph. The "edge drawing" tool of the visual editor uses the "edgehandles" extension[4], which provides a user interface for dynamically connecting nodes with edges. The graph navigator capability is based on the "navigator" extension[5], which provides a bird's eye view with pan and zoom control from the graph.

For converting a story from natural language to the STAR syntax, we use a custom-built component developed at our lab, which uses the Stanford CoreNLP for natural language processing, a python script for processing the NLP output, and PHP for orchestrating and delivering the results through a RESTful API. The Web-STAR IDE integrates this component into its workflow, while the same methodology can be used by other systems to acquire this functionality.

The Web-STAR platform publishes two web services that can be used by third party applications, for adding a domain file to the STAR system queue in order to process, and for retrieving the results after the completion of the reasoning process (cf. Figure 4.1): the "`add_story_queue`" web service takes as a parameter the story in the STAR syntax and returns a unique identifier; the "`retrieve_story_results`" web service takes as a parameter the unique identifier previously sent by the "`add_story_queue`" web service, and returns the results of the comprehension process. This approach was chosen to minimize the waiting time in cases of large story files that require extensive processing.

[3] https://github.com/iVis-at-Bilkent/cytoscape.js-expand-collapse

[4] https://github.com/cytoscape/cytoscape.js-edgehandles

[5] https://github.com/cytoscape/cytoscape.js-navigator

4.4 Web-STAR IDE Evaluation

An important step in the design and deployment of a web-based IDE is the evaluation of its usability, i.e., “the degree to which users are able to use the system with the skills, knowledge, stereotypes, and experience they can bring to bear” (Eason, 2005). Usability evaluation can be conducted using interviews, task analysis, direct observation, questionnaires, and heuristic evaluation, among others (Barnum, 2001). In terms of evaluating an IDE, Kline and Seffah (2005) presented three techniques which can also be applied for the Web-STAR IDE’s evaluation: 1) the unstructured interviews, 2) the heuristic evaluation and psychometric assessment, and 3) the laboratory observation combined with the cognitive walkthrough. Moreover, in the work by Pansanato et al. (2015), the capturing of user interaction is stretched for usability evaluation of rich web interfaces. The authors present a number of tools and methods that go beyond simple capturing of log files from the web server, like the recording of user interaction from the client side, i.e., the browser.

4.4.1 Evaluation Setting

In this work, we followed a hybrid approach for the Web-STAR IDE’s evaluation that combines the cognitive walkthrough method (Blackmon et al., 2002; John and Packer, 1995) with questionnaires and user interaction capturing techniques. The process was divided into the design phase, the pilot phase, and the actual evaluation phase, and sought to:

- Evaluate the web-interface in terms of ease of use, understanding, learnability, and efficiency.
- Detect possible usability problems of the Web-STAR IDE.
- Perform the above for both experts and non-experts that use the IDE.

Design Phase

The design phase involved the selection of the participants for the evaluation, the design of the tasks that each participant would undertake, the preparation of the questionnaires, and the technical methods for tracking each participant’s interaction with the system.

Participants were chosen from both groups that would have an interest in using the Web-STAR IDE: 1) experts, and 2) non-experts. The expert group included computer scientists and psychologists with prior experience in using the STAR system as a standalone Prolog application, and computer scientists or computer science students with programming skills in Prolog or other declarative programming languages. The non-expert group included psychologists, school teachers, law students, and students of psychology, who had very little or no experience in using IDEs or programming languages. A total of 15 participants were selected, which, according to the relevant bibliography (Macefield, 2009), is an appropriate sample for detecting the majority of the usability problems of a system.

We compiled a list of **Cognitive Walkthrough Tasks** that were specifically designed to evaluate the major functions and aspects of the Web-STAR IDE. Each task instructed the user to perform a sequence of actions, as follows:

- **Task 1 (Create an account)**: Navigate to the Web-STAR IDE link and create an account. Activate the account and log into the system.
- **Task 2 (Follow the guided tour)**: Follow the guided tour to learn the basic functionality of the IDE.
- **Task 3a (Write a new story in natural language)**: Write a given story along with its questions in natural language and convert it to the STAR syntax. Then add the background knowledge given in a visual format, using the visual editor and convert it to the STAR syntax. Save the story.
- **Task 3b (Write a new story in the STAR syntax)**: Write a given story along with its questions in symbolic format, using the source code editor. Then add the background knowledge given in the STAR syntax, and convert it to the visual format. Save the story.
- **Task 4 (Load a story and initiate the comprehension process)**: Choose a public story, load it, and initiate the story comprehension process.
- **Task 5a (Modify the background knowledge using the visual format editor)**: Load a story, add a new background knowledge rule given in visual format, update and remove an existing rule, all using the visual editor. Finally, initiate the comprehension process.
- **Task 5b (Modify the background knowledge using the source code editor)**: Load a story, add a new background knowledge rule given in the STAR syntax, update and remove an existing rule, all using the source code editor. Finally, initiate the story comprehension process.
- **Task 6 (Filter the output of the comprehension process)**: Load a story, initiate the comprehension process, and filter the output to present only the concepts that change while the story unfolds.

- **Task 7 (Share a story)**: Load a story and share it in the public story repository.
- **Task 8 (Comment on a user's story)**: Find a story in the public story repository and add a comment on that story.
- **Task 9 (Initiate the collaboration tool)**: Initiate the collaboration functionality, and send the generated collaboration link to another person using the feedback option.

Since not all participants are experts in encoding stories in a symbolic language, one of the major considerations when preparing these tasks was to obtain comparable results from the users. Towards that aim, both the story and the background knowledge in Tasks 3a and 3b were provided in the instructions given to the users.

After observing each participant performing the above tasks, the experimenters tried to answer the following **Cognitive Walkthrough Questions**, as explained in the work of Wharton et al. (1994):

- Does the user try to achieve the right effect?
- Does the user notice that the correct action is available?
- Does the user associate the correct action with the effect that the user is trying to achieve?
- If the correct action is performed, does the user see that progress is being made toward the solution of the task?

When the answer to any of these questions was "No", an error was counted towards the total number of errors for that task.

The time needed to complete each task was calculated from the time the participant logged into the IDE until the time the participant logged out of it, with an exception in the first task where the time was measured from the time the participant clicked the register button until the time the participant logged out of the IDE.

After the completion of the tasks, a **Demographics Questionnaire** was completed by participants to record their gender, age, degree, occupation, previous experience in using IDEs, knowledge of programming languages, and prior experience in using story understanding systems and more specifically the STAR system. A **Post-task Questionnaire** was also completed to capture the participants' opinion for using the IDE for each specific task. The questionnaire included questions that covered the various parts of the system invoked for each task: the interface (e.g., menu bar, panels, dialogs, buttons, and labels), the online help material, the visual editor, the public story repository, and the outcome of the story comprehension process. It included true/false questions, multiple choice questions, and questions in the five-point Likert scale. The questionnaire is available in Appendix B and it also includes a section with questions from the **System Usability Scale (SUS) standardized questionnaire** (Brooke, 1996), a ten-item questionnaire using a five-point scale for the assessment of perceived usability (Lewis et al., 2015). Both surveys were designed and deployed online and access to them was restricted to participants of our evaluation.

Finally, we implemented a logging functionality to capture detailed information from the participants' interaction with the Web-STAR IDE (e.g., login, menu selection, button click, visual editor usage) and measure the time between these interactions. This functionality was seamlessly integrated with the Web-STAR IDE using AJAX technology. Each event was stored in a database table and included the user-id of the participant that performed the action, the time the action was performed, the component used (e.g., login screen, menu bar, visual editor), the action (e.g., button click, visual editor graph node added), the data sent, and the response of the IDE.

The metrics chosen for the evaluation were both qualitative (e.g., user satisfaction, ease of use) and quantitative (e.g., number of successfully performed tasks, task completion time, number of errors occurred, number of times participants used the online help functionality, number of times participants clicked on a control).

Pilot Phase

Before the actual evaluation phase, we performed a pilot evaluation identical to the actual one, but with only two users, to verify that all tasks are feasible and understandable. This also allowed us to test that data were recorded properly and to get familiar with the testing process.

The pilot evaluation was performed in a laboratory environment with a computer connected to the Internet with access to the Web-STAR IDE URL. We also set up the screen recorder software to capture all interactions of the user with the interface (e.g., keystrokes, mouse movements, information dialogs, and visual editors) in a video file.

Both participants ended up needing more than an hour to complete the tasks and respond to the questionnaires.

Evaluation Phase

All participants in the evaluation phase performed the experiment in a controlled environment which included a laptop with an Intel CORE i7 processor, 4GB of RAM, and a 15.4 inches screen. An external mouse was attached to the laptop, and participants had instructions to use it (instead of the laptop's integrated mousepad). The laptop was constantly plugged into the power source. In terms of software, the Firefox web-browser was used to load the WebSTAR IDE interface, and the Camtasia screen recording software was activated before each session to record the participant's actions. Each session was performed in a quiet room with only the participant and the experimenter present, aiming to minimize outside interference and noise from the environment.

As a first step, each participant completed a statement of informed consent regarding the reason for the evaluation, and the data collection and data handling policy. This consent was mandatory for participating in the evaluation. The participants were then asked to complete the Demographics Questionnaire online.

Following that, participants were presented with a document listing the Cognitive Walkthrough Tasks (cf. Appendix A). During the cognitive walkthrough, participants had continuous access, through the Web-STAR IDE, to online help files provided by the IDE, the STAR syntax guide, and the guided tour; i.e., the same type of help that any typical user of the IDE would have available while using it. Participants had the option to choose any type of viewing mode they saw fit when completing the tasks.

During each task, the experimenter recorded all observations made in a notebook, answered the cognitive walkthrough questions, and recorded problems and errors occurred while the participant used the IDE. After each task, the participant was presented with the post-task questionnaire for that specific task. The experimenter avoided providing any kind of verbal or non-verbal additional help to the participant while conducting the cognitive walkthrough.

After the completion of all the tasks, participants were presented with the System Usability Scale (SUS) standardized questionnaire. Finally, the experimenter stopped the screen recording, stored the capture, and saved all questionnaire answers online for later processing.

4.4.2 Evaluation Results

Fifteen people (8 male, 7 female) participated voluntarily to the evaluation. All had Greek as their mother tongue and reported to have a normal or corrected-to-normal vision. All participants completed the whole evaluation process. Figure 4.11 represents analytics regarding their gender, age group, education, employment status, and knowledge of programming languages and IDEs. More than half of the participants (8 out of 15) reported that they had heard the notion of story understanding, but only 2 reported that they had used a story understanding system before. In both cases, this system was the STAR system. The group of non-experts included 10 participants and the group of experts 5.

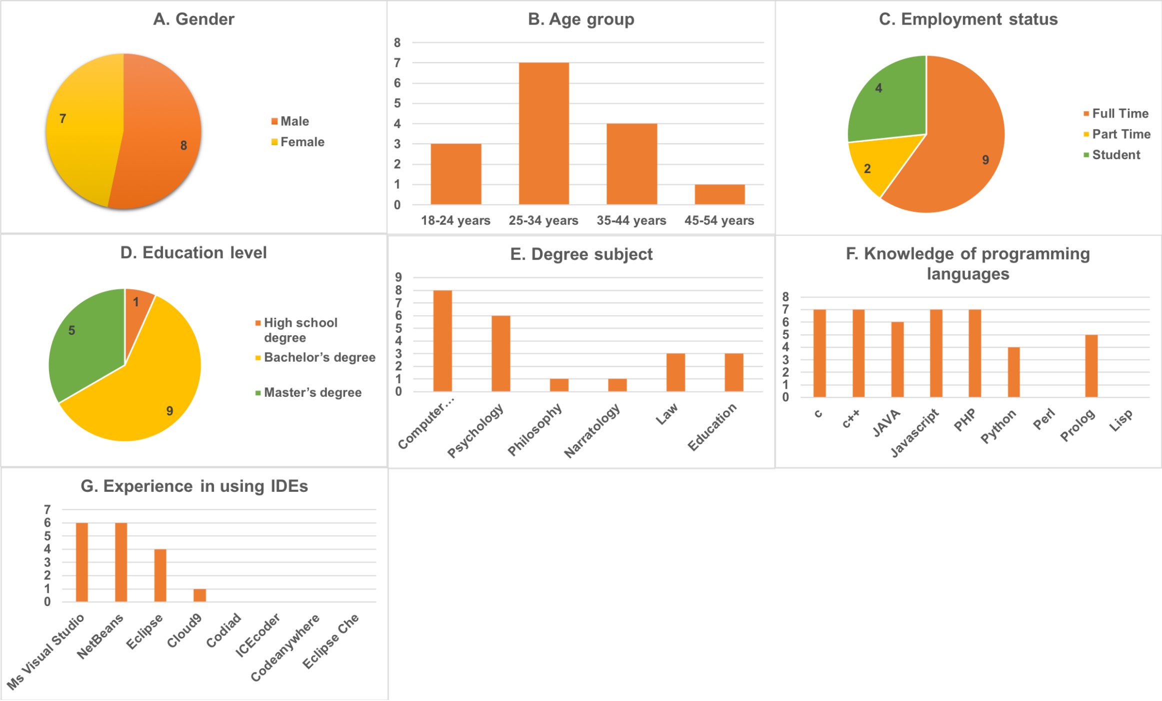


Figure 4.11 Participants' demographics. Graph A depicts their gender, Graph B their age group distribution, Graph C their employment status, Graph D their education level, Graph E their degree subject, Graph F their knowledge of programming languages, and Graph G their experience in using IDEs.

After the completion of the evaluation, the notes taken by the examiner for each participant with answers to the Congitive walktrough questions were carefully examined along with the answers in the post-evaluation questionnaire. The log files of each participant were also analyzed, and the aggregated results are presented in Table 4.1 and Figure 4.12.

The average total time for completing the evaluation tasks from both groups satisfy the normality assumption based on the Kolmogorov-Smirnov test. On average, experts performed the tasks of the cognitive walkthrough in less time (M=2622.40 seconds, SE=107.44) than non-experts (M=3150.00 seconds, SE=98.01), and the difference was significant $t(13)=3.32$, $p<.05$.

Further analysis of the results per task gives more insights on how participants interacted with the system (cf. Table 4.1).

Results From the Cognitive Walkthrough Process

In the following paragraphs we present the findings of the cognitive walkthrough process:

Table 4.1 Performance at the Cognitive Walkthrough evaluation

Task	Completed[a]	Avg[b]	Std[c]	Max[d]	Min[e]	Errors[f]
Task 1	100%	111	45	202	37	0
Task 2	100%	692	269	1303	224	0
Task 3a	100%	640	168	1016	387	0
Task 3b	100%	119	23	156	74	0
Task 4	100%	186	95	381	100	0
Task 5a	100%	615	105	861	438	0
Task 5b	100%	244	58	338	164	0
Task 6	100%	85	19	114	54	0
Task 7	100%	98	40	186	41	0
Task 8	100%	107	42	230	56	0
Task 9	100%	75	23	132	38	0

[a] Percentage of participants that successfully completed the task.
[b] Average time (in seconds) needed to complete the task.
[c] Standard time deviation (in seconds) needed to complete the task.
[d] Maximum time (in seconds) needed to complete the task.
[e] Minimum time (in seconds) needed to complete the task.
[f] Number of errors recorded by the experimenter during the task.

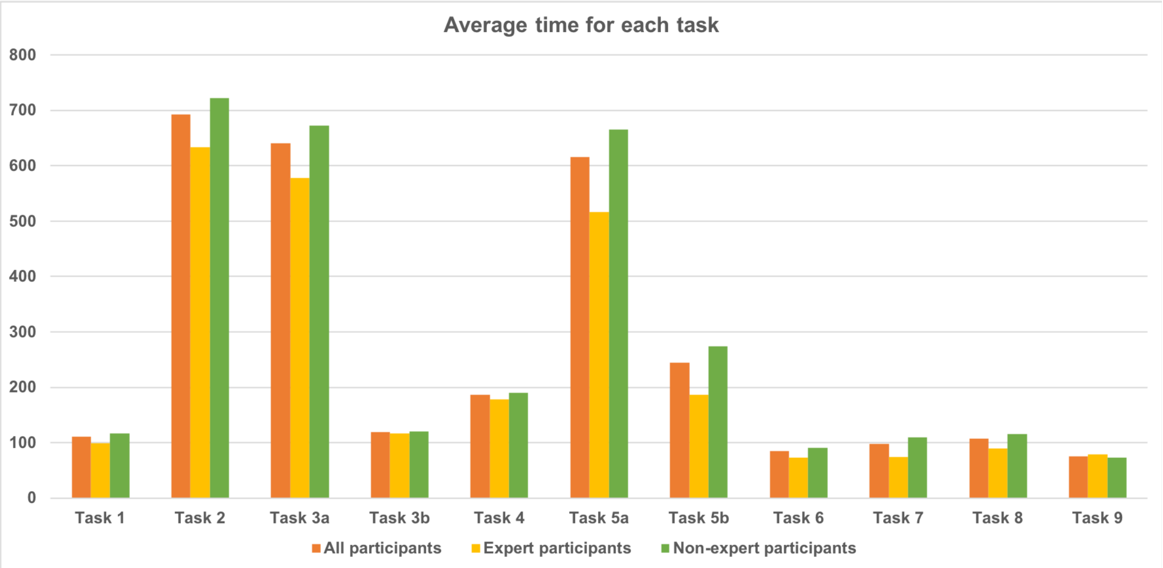


Figure 4.12 Average time in seconds needed per task, for experts, non-experts, and all participants.

Task 1: All participants completed this task successfully. Some participants did not receive the activation email immediately due to email provider delays. The majority of participants clicked the links included in the introductory text while some others chose the "Register" option from the menu bar. It was also common for participants to try and press the enter button after filling up their credentials, but it was not working and they needed to click the "Login" button instead to proceed.

Task 2: All participants completed this task successfully. The majority of participants initiated the guided tour using the assistant dialog. A small number of participants did not understand that the assistant's message is clickable, and used the "Help menu" to find and start the guided tour. Some participants interacted with the IDE while following the guided tour and performed actions like "reading a story", "drawing background knowledge" using the visual editor and converting both the story and the background knowledge to test it. There were cases of participants who went back to a specific step, to test a functionality mentioned later in the guided tour. Moreover, one participant also expressed the opinion that it would be very useful if there was an option to watch a video instead of the guided tour. When the guided tour was showcasing the output panel, participants expected to have the story comprehension output area filled up with story information, but it was empty, since the story comprehension process was not activated by the guided tour.

Task 3a: All participants completed this task successfully. Some participants watched the help video first to properly perform the task. Participants used the visualizations, e.g., highlight of literals and rules which can be connected with an edge while drawing it, and red highlighting for incomplete rules along with debugging messages. Some participants were double clicking on the nodes and edges to move them and add arguments, when only a single click was needed.

Task 3b: All participants completed this task successfully. Participants found easily where to add the story and the background knowledge. A number of participants who tested the toolbar of the source code editor used functions and controls like the "text wrap" and the "font size".

Task 4: All participants completed this task successfully. Some participants had difficulties finding the "Read Story" button and they tried to locate it on the menu bar or on the top area of the IDE. Furthermore, non-experts read the confirmation message to understand where they could find the story output and when the reading process was completed.

Task 5a: All participants completed this task successfully. Some non-experts deleted the rule, but they forgot to delete the connected literal, whereas experts deleted the connected literal along with the rule. When the former users tried to convert the graph to the STAR syntax, the system's debugging messages guided them to delete the connected literal as well before proceeding with he conversion. Some participants did not notice that some literals were existing and when they tried to add them, the debugging messages informed them that the literal they were trying to add already existed, so then they proceeded with connecting the existing literal with the rule.

Task 5b: All participants completed this task successfully. The majority of experts, when instructed to delete a rule, they commented it out, whereas non-experts proceeded with erasing it. Some experts also used the search functionality to find the rule and then delete it.

Task 6: All participants completed this task successfully. They found the filtering options very easily. Some participants tried to use the filtering option before the story comprehension process was completed and they could not, since the option was available only after the completion of the process. Hence, they waited for the reading process to finish and then tried to apply the filter.

Task 7: All participants completed this task successfully. The majority of participants had difficulty in locating the share button. First, they searched for it on the menu bar and then at the story area. Only after careful examination of the screen they were able to locate it. Some participants browsed to the save story window and chose the “private/public” toggle to share the story. In most cases, participants scrolled up and down the IDE page to find the relevant control to share a story.

Task 8: All participants completed this task successfully. Some participants had difficulties locating how to comment on a story. They searched for the button on the menu bar and then they navigated to the public story repository to find the commenting functionality.

Task 9: All participants completed this task successfully. Some participants did not locate the “Start Collaboration” button immediately and searched for it in the public story repository.

Results From the Post-task Questionnaire

Results from the post-task questionnaire show that for **Task 1**, on average, participants strongly agree that the process of creating a new account (M_E=5.0, M_{NE}=4.9)[6] and activating it (M_E=4.8, M_{NE}=4.9) is easy and is the same (for creating, M_E=4.6, M_{NE}=4.9), (for activation, M_E=4.8, M_{NE}=5.0) with that of the other systems they are using.

For **Task 2**, on average, experts agree and non-experts strongly agree that it is easy to find and start the guided tour (M_E=4.4, M_{NE}=4.8). On average, both experts and non-experts strongly agree that the duration of the guided tour is appropriate for learning the basics of the

[6] M_E and M_{NE} represent the means of the Likert scale scores given by expert and non-expert participants, respectively.

IDE (M_E=4.6, M_{NE}=4.5). In terms of feeling confident in using the IDE after the guided tour, on average, experts strongly agree that this is the case and non-experts agree as well (M_E=4.6, M_{NE}=3.9).

For **Task 3a**, on average, participants strongly agree that it is easy to write the story in natural language (M_E=5.0, M_{NE}=4.9) and automatically convert it to the STAR syntax (M_E=5.0, M_{NE}=5.0). On average, experts agree and non-experts strongly agree that it is easy to add the background knowledge of the story using the visual editor (M_E=4.4, M_{NE}=4.5). Both groups strongly agree that the automatic conversion of the background knowledge in visual format to the STAR syntax is easy (M_E=5.0, M_{NE}=5.0). As for saving the story, participants strongly agree that it is an easy task (M_E=5.0, M_{NE}=5.0). Four non-experts have used the online help facility to perform this task and on average, they strongly agree that the help available from the system to perform this task is adequate (M_{NE}=4.8).

For **Task 3b**, on average, participants strongly agree that it is easy to write the story in the STAR syntax (M_E=5.0, M_{NE}=4.6). On average, experts agree and non-experts strongly agree that it is more efficient to write the story in natural language and then convert it to the STAR syntax than writing the story directly using the STAR syntax (M_E=4.4, M_{NE}=5.0). On average, participants strongly agree that it is easy to add the background knowledge in the source code editor (M_E=5.0, M_{NE}=4.9). One non-expert stated that this does not apply. Both groups on average, strongly agree that it is easy to convert the background knowledge from the STAR syntax to visual format (M_E=5.0, M_{NE}=5.0). Both experts and non-experts, on average, agree that it is easier to understand the background knowledge rules in visual format than in the STAR syntax (M_E=3.6, M_{NE}=4.3). Although the means of the two groups on this question appear to have a difference larger than that of other questions, further analysis revealed that this difference was not found to be statistically significant based on the Mann-Whitney test, U=14.00, $z = -1.42$, $p > .05$, $r = -0.37$. Participants strongly agree that it is easy to save the story. None of the participants has used the online help facility to perform this task.

For **Task 4**, on average, participants strongly agree that it is easy to find a story and load it (M_E=5.0, M_{NE}=5.0) and that the story load window is easy to use (M_E=5.0, M_{NE}=4.9). On average, experts agree and non-experts strongly agree that it is easy to find how to initiate the story comprehension process (M_E=4.2, M_{NE}=4.8). On average, both groups strongly agree that the system provides continuous feedback on the comprehension process status (M_E=4.8, M_{NE}=4.9). In terms of finding the answer that the system gave to a question, on average, both experts and non-experts strongly agree that it is easy to find the answer to the question using the visual output panel (M_E=5.0, M_{NE}=4.7). One expert participant stated that this does not apply since he/she used only the raw output. On average, experts strongly agree and non-experts agree that it is easy to find the answer to the question using the raw output panel (M_E=5.0, M_{NE}=4.0). Five participants (1 expert and 4 non-experts) stated that this does not apply since they used only the visual output. On average, both experts and non-experts strongly agree that the visual output panel presents the story concepts and questions in an understandable way (M_E=4.8, M_{NE}=4.7). For the raw output panel, on average, experts strongly agree and non-experts agree that it presents the various story concepts and questions in an understandable way (M_E=5.0, M_{NE}=4.0). Five participants (1 expert and 4 non-experts) stated that this does not apply since they used only the visual output. Two participants (one from each group) used the online help facility and all participants were able to find the correct answer to the question.

For **Task 5a**, on average, experts agree and non-experts strongly agree that it is easy to add a rule using the background knowledge visual editor (M_E=4.4, M_{NE}=4.9). On average, both experts and non-experts strongly agree that it is easy to delete (M_E=4.6, M_{NE}=4.8) and edit (M_E=5.0, M_{NE}=4.8) a rule using the background knowledge visual editor. Regarding the controls available in the background knowledge visual editor, on average, experts strongly agree and non-experts agree that they are easy to use (M_E=4.6, M_{NE}=4.4). On average, both experts and non-experts strongly agree that it is easy to understand the functionality of the controls in the visual editor's toolbar (M_E=4.8, M_{NE}=4.6). Only one non-expert participant has used the online help facility. All participants were able to find the correct answer to the question.

For **Task 5b**, on average, both experts and non-experts strongly agree that it is easy to add, delete and edit a rule using the background knowledge source code editor (M_E=5.0, M_{NE}=5.0). Moreover, on average, both groups strongly agree that the controls available in the background knowledge source code editor's toolbar are easy to use (M_E=4.8, M_{NE}=5.0). On average, experts strongly agree and non-experts agree that it is easy to understand what is the functionality of the controls in the background knowledge source code editor (M_E=4.8, M_{NE}=4.9). One non-expert participant stated that this does not apply. In terms of what is the most efficient method to modify the background knowledge, on average, experts neither agree nor disagree that it is the visual editor, whereas non-experts agree that the visual editor is more efficient than the source code editor (M_E=2.8, M_{NE}=4.0). One non-expert participant stated that this does not apply. This difference between the means of the two groups was not found to be statistically significant based on the Mann-Whitney test, U=10.50, z= -1.65, p>.05, r= -0.44. None of the participants had used the online help facility. All participants but one, were able to find the correct answer to the question.

For **Task 6**, on average, both experts and non-experts strongly agree that it is easy to find and apply the filtering functionality (M_E=5.0, M_{NE}=5.0). Moreover, on average, participants strongly agree that the filters available can help extract information from the comprehension model (M_E=4.8, M_{NE}=4.9). None of the participants had used the online help facility.

For **Task 7**, on average, both experts and non-experts strongly agree that it is easy to find a demo story and load it (M_E=5.0, M_{NE}=5.0) and that the story browser window is easy to use (M_E=5.0, M_{NE}=5.0). On average, both groups agree that it is easy to find how to share a story (M_E=4.0, M_{NE}=3.5). None of the participants had used the online help facility.

For **Task 8**, on average, experts agree and non-experts strongly agree that it is easy to find a story in the public story repository (M_E=4.4, M_{NE}=4.9). On average, both groups agree that it is easy to comment on a story (M_E=4.4, M_{NE}=4.4) and strongly agree that comments added by others are clearly presented on the screen (M_E=4.8, M_{NE}=4.6).

For **Task 9**, on average, both experts and non-experts strongly agree that it is easy to find how to initiate the collaboration functionality (M_E=4.6, M_{NE}=4.9). On average, experts agree and non-experts strongly agree that the collaboration functionality could be useful for teaching logic programming (M_E=4.4, M_{NE}=4.7), collaboratively creating stories (M_E=4.4, M_{NE}=4.7) and collaboratively designing knowledge (M_E=4.2, M_{NE}=4.7).

For all tasks, on average, participants strongly agree that the feedback messages from the system are helpful.

Results From the Logging Functionality

During the experiment, all participants' interactions with the IDE were captured and stored in the database. The clicks per user for both experts and non-experts are presented in the following graphs, with a focus on the clicks on the help facilities (cf. Figure 4.13), and on the 10 most clicked functions per user (cf. Figure 4.14).

As the results show, the background knowledge visual editor is the most clickable area. This was expected since participants had to draw and edit knowledge rules using the visual editor. In general, experts and non-experts had little difference in the number of clicks per area and function.

Results From the System Usability Scale (SUS) Questionnaire

Results from the System Usability Scale (SUS) standardized questionnaire show an average score of **88.33** out of **100**. The maximum score of the participants was 100, the minimum

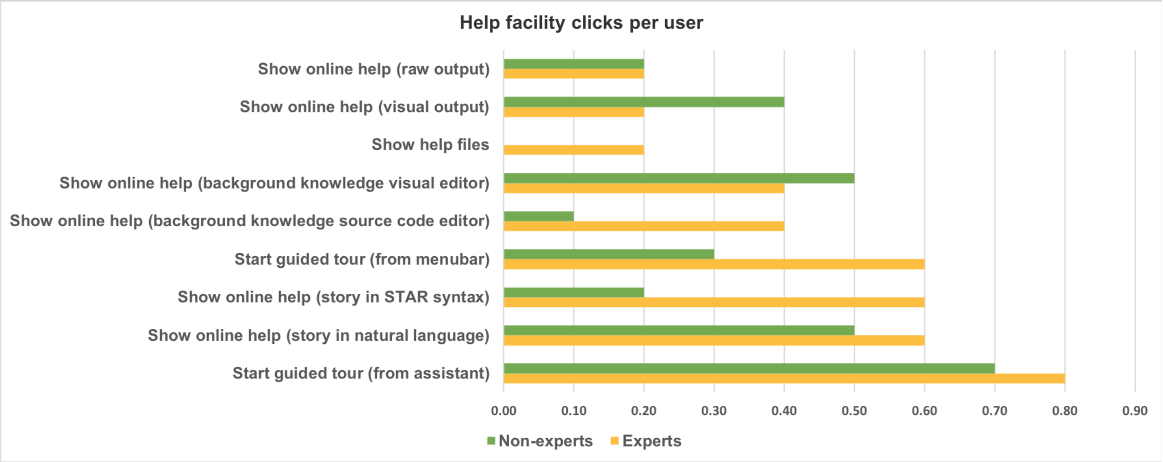


Figure 4.13 Mean number of clicks per user for expert and non-expert participants on the help options of the IDE.

Table 4.2 Results of the System Usability Scale (SUS) standardized questionnaire

Group	Average score	Std[a]	Max[b]	Min[c]
Experts	90.00	7.07	95.00	77.50
Non-Experts	88.25	9.43	100	70
TOTAL	**88.83**	8.50	100	70

[a] Standard deviation.
[b] The maximum score.
[c] The minimum score.

was 70 and the standard deviation was 8.5. Results are depicted in Table 4.2 for both groups as well as for the entire set of participants.

Compared to the SUS scores obtained from the evaluations of other systems, the Web-STAR is ranked in the top category, i.e., between “excellent” and “best imaginable” in the adjective ratings scale (Bangor et al., 2009).

4.4.3 Analysis of Results

The evaluation process followed allowed a thorough investigation of the participants’ actions, impressions, and feedback while using the IDE. The combination of a cognitive walkthrough, with questionnaires, and with close monitoring offered information that could not have been obtained only by using a single method for evaluation. The diverse group of participants in this evaluation gives insights into how people from different backgrounds and prior

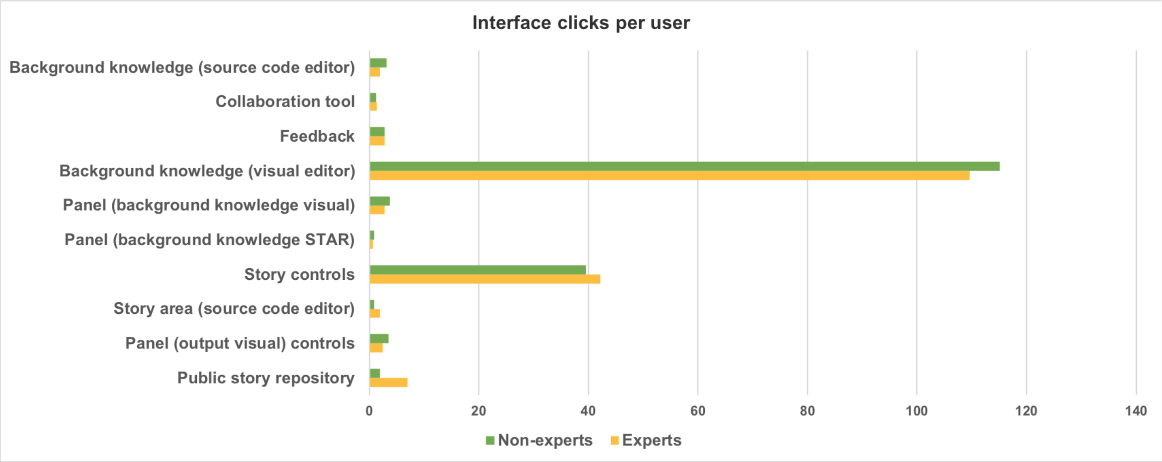


Figure 4.14 The 10 most clickable parts of the interface per participant. The X-axis represents the mean number of clicks.

experience in using story understanding systems and IDEs in general can benefit from the various features of the Web-STAR IDE.

After examination of the results, we report that all participants, experts and non-experts, managed to complete all the tasks. In general, participants did not have much difficulty while performing the tasks. For some tasks, like sharing a story and commenting on it, participants had some trouble finding the relevant controls since they were not in the "expected" area of the IDE (e.g., the menu bar).

Both experts and non-experts managed to setup an account, activate it and access the IDE in less than 2 minutes time. Participants were able to do that because the registration process is similar to that of other online systems they already have accounts on and use. They were able to start using the IDE in a very short time, by following the guided tour.

For the main task that the IDE facilitates which is writing stories, both experts and non-experts were able to encode stories either by converting them from natural language to the STAR syntax or by writing them directly in the STAR syntax. Regarding the background knowledge, participants were able to encode it easily using the visual editor and the source code editor (even thought they just had to copy the prepared story in symbolic format). All participants agree that it is easier to write the story in natural language and then convert it to the STAR syntax than writing it directly in symbolic format. Moreover, all participants were able to understand the background knowledge rules when using the visual editor and the graph representation of the background knowledge. This was clear by the answers given to the post-task questionnaire and from the time the participants took to complete the relevant tasks. This is important, since participants can use the component that best fits their working style and needs, to perform this action.

For editing the background knowledge, expert participants found the usage of the source code editor more efficient than that of the visual editor. This is to be expected, since it is presumably more time-consuming to draw a rule using the visual editor than to write it in the source code editor. For non-expert users, this was clearly not the case, since they agree that the visual editor is more efficient for changing the background knowledge. We assume that this could be because they can understand better the graph representation of the knowledge instead of the STAR syntax that they are not familiar with.

In terms of finding the answer to the questions posed, all participants were able to perform this task quite easily using either the visual or the raw output panel. Experts preferred the raw output which was enhanced with color highlighting for questions and answers and non-experts preferred the visual output with the time-line format. A number of non-expert participants chose to use only the visual output to find the answer. This is justified by the fact that the answer could be easily extracted from the time-line without the need to explore the raw output.

At this point, we observed that when a story had several scenes and a participant tried to find the answer to a question from the first scene using the raw output, he/she needed to scroll up to find it. Hence, we decided to add in the next version of the system, the option to split the raw output to scenes, so that this user burden can be avoided, by making it easier to browse each scene from the raw output. Both groups benefited from the time-line format since it was easier to understand the various concepts of each story, apply filters on them, and find answers to questions.

The social and collaboration features of the IDE were also simple to use. Both groups were able to share a story or add comments to a story in the public story repository in a very short time. We observed that a number of participants had a problem spotting the relevant controls, since they were not located in the expected area. Hence, we decided to add a menu option that groups all these controls and buttons together for easy access in the next version of the system. Participants also found the collaboration feature very useful, since they confirmed that it could be useful for teaching logic programming, collaboratively creating stories, and collaboratively designing knowledge.

Results from the SUS standardized questionnaire dictate that the Web-STAR IDE is a friendly, easy to use, and easy to learn IDE. This evaluation led to some minor changes in the IDE to enhance user experience and productivity.

4.5 Discussion

This chapter focuses on Web-STAR, a platform built on top of the STAR system for story comprehension to facilitate the interaction of users with a story comprehension system and for acquiring commonsense knowledge. We presented the various features of the platform through examples, and have argued that the platform is designed to appeal to both expert and non-expert users. The argument is supported, in particular, by the visualization that Web-STAR offers for the background knowledge that is used during story comprehension, and for the output of the story comprehension process. A comprehensive evaluation of the usability of the platform has supported that the platform is, indeed, friendly, easy to use, and easy to learn.

Moreover the fact that non-experts are able to encode knowledge in a logic-based language is a big plus for the field and an argument in favor of the knowledge representation chosen by the STAR system developers, which seems to find users outside of the exert users sphere.

The evaluation of the Web-STAR IDE provides evidence that using visual interfaces is an appropriate method for acquiring knowledge by both experts and non-expert users. Furthermore, the graph representation used in the Web-STAR IDE is an intuitive method to add background knowledge since it was positively evaluated by both experts and non-expert users. User contributed knowledge is also suitable for question answering on stories using an argumentation-based reasoning system.

The platform is currently used for educational purposes, helping students and researchers engage with the problem of automated story understanding. More than 70 users have registered so far, and have contributed more than 80 stories. Furthermore, the platform has received more than 5200 web service calls for processing STAR domain files.

The webservices of the platform are also used as part of an implicit knowledge workflow and is used in the Robot Trainer Game (cf. Chapter 5), a crowdsourcing game with a purpose to gather commonsense knowledge; in this context, knowledge contributed by users was processed in real-time to determine its sufficiency to answer story questions. The demonstrable ease of use of the Web-STAR platform, and its online and visual environment, makes it a prime candidate for use by domain experts in law, history, or literature, who may wish to comprehend text in the form of narratives.

Future versions of the platform will aim to refine its interface and extend its functionality. In terms of the latter, we are considering the addition of the option to import and process relevant background knowledge from existing knowledge bases like the ones already presented in Chapter 2, ConceptNet (Speer et al., 2017), YAGO (Mahdisoltani et al., 2015), NELL (Mitchell et al., 2015) and the OpenCyc project (Lenat, 1995). Other sources of knowledge (implicit or explicit) could include Games With A purpose (von Ahn and Dabbish, 2008), payed workers from crowdsourcing platforms like Amazon Mechanical Turk (Buhrmester et al., 2011b); or even from machine learning algorithms that produce rule-based knowledge bases (Michael, 2009, 2016, 2017). The component that converts natural language stories into the STAR syntax could be further extended, by incorporating systems that extract knowledge from natural language (Corcoglioniti et al., 2016), and identify the temporal ordering of events (UzZaman et al., 2013).

The Web-STAR IDE is a prime example of a web-based automated story understanding system that in addition to its core functionality, is also able to provide the user with explanations on why the system came to a certain outcome. Users can view the comprehension model in a visual format and furthermore they can toggle the debugging options of the system and view the activated rules, the accepted arguments and the conflicts between the arguments that led the STAR system to produce a particular comprehension model of a story.

Work on Web-STAR could serve as a basis for establishing a story-sharing and story-processing community, towards the advancement of work in automated story understanding through symbolic knowledge and reasoning.

Related Publications:

(1) Christos T. Rodosthenous and Loizos Michael. Web-STAR: A Visual Web-based IDE for a Story Comprehension System. Theory and Practice of Logic Programming, 19(2):317–359, 2019.

(2) Christos T. Rodosthenous and Loizos Michael. Web-STAR: Towards a Visual Web-Based IDE for a Story Comprehension System. In Proceedings of the 2nd International Workshop on User-Oriented Logic Paradigms (IULP2017), Espoo, Finland, 2017. arXiv.

5

Story Understanding Using Games for Commonsense Knowledge Acquisition

> *"When we program a computer to make choices intelligently after determining its options, examining their consequences, and deciding which is most favorable or most moral or whatever, we must program it to take an attitude towards its freedom of choice essentially isomorphic to that which a human must take to his own."*
>
> – John McCarthy, *Ascribing Mental Qualities to Machines (1979)*

5.1 Introduction

So far we have worked on how we can acquire knowledge using explicit acquisition methods, i.e., methods where the contributor is aware and knowledgeable on how to contribute knowledge. Even though this method can yield good results, it has a number of drawbacks, such as the slow acquisition rate, the need for users that can encode knowledge, and the need for tools to support this method. Encoding of knowledge is not trivial task, as we have already explained in Chapter 4, since it requires a number of preconditions, such as proper representation of knowledge, different types of knowledge (e.g., rules, facts), methods to resolve conflicts in knowledge, and methods to select relevant knowledge.

In this chapter, we present our work on knowledge acquisition using implicit knowledge acquisition methods, i.e., methods where users are contributing knowledge as a side task of a bigger task, such as a game or a learning application. We focus on describing a method for acquiring background knowledge through crowdsourcing (cf. Chapter 2, Section 2.3), and we initiate an investigation of whether *a fully* crowdsourced method for knowledge acquisition is feasible, and competitive against other automated or semi-automated (hybrid) approaches.

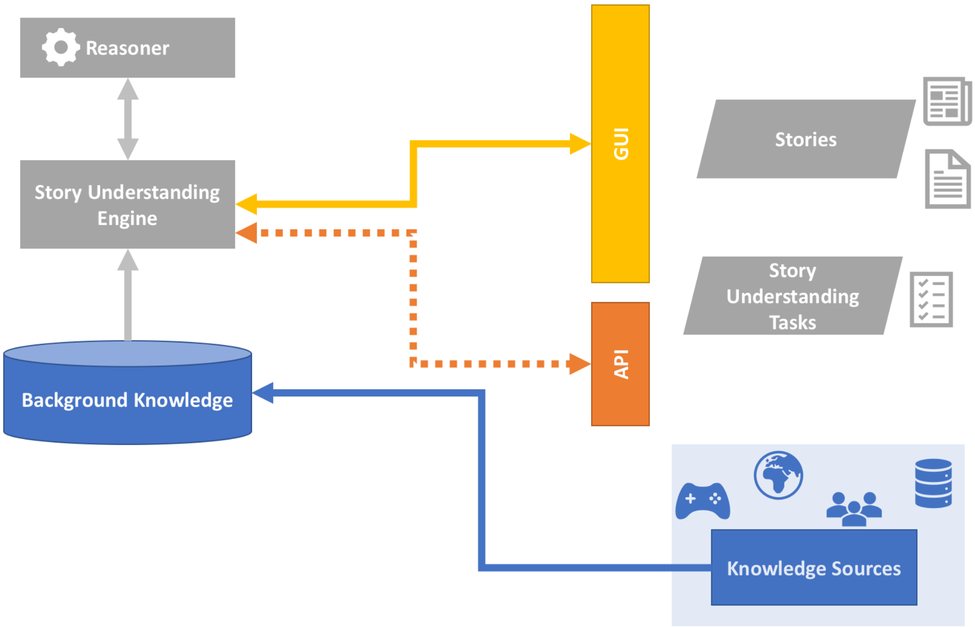


Figure 5.1 An architecture for a fully-fledged story understanding platform. Users can write stories or import them and send them to the story understanding engine. Background knowledge from external sources is continually updating and feeds the engine. The system is connected to a reasoner for inference generation. The user selects a story understanding task (e.g., question answering, paraphrasing, summarization)

Towards this direction, we created a platform suitable for developing Games With A Purpose (GWAPs) and monitoring experiments. A high-level architecture for the platform is depicted in Figure 5.1.

5.2 A Platform for Knowledge Acquisition, Applications and Games

Following our vision for acquiring commonsense knowledge using crowdsourcing, we designed a platform which offers features and services that can be used to facilitate commonsense knowledge gathering from a number of paradigms, such as games, crowdsourcing tasks and mini applications. Most of the platform's specifications are applied in the majority of crowdsourcing platforms and applications and some of them are specific for the task of acquiring commonsense knowledge.

5.2.1 Platform Specifications

For developing the platform, we considered the following key design options: 1. the selection of a suitable technology for delivering task-based applications and GWAPs, 2. the handling of contributors' profiles, and 3. the representation of knowledge in a structured form that can be reused and verified. The platform should also allow monitoring of the acquisition process both in terms of contributors and acquired knowledge.

Furthermore, the platform should be able to offer a number of design elements needed in games and educational applications. These include but are not limited to: 1. leader boards, 2. contributors' ranking, 3. medals and awards, 4. progress-bars, 5. live feedback with notifications (both synchronous and asynchronous) for the events, and other gamification elements needed to provide the user with a pleasant experience while contributing (Mekler et al., 2013).

On the back-end, the platform should be able to provide tools for designing a crowdsourcing application and managing contributors. These tools should provide developers the ability to easily change parameters of the application, e.g., number of raters for acquired knowledge to be valid, dynamic loading and changing of datasets (testing and validation) and export statistics on the system usage.

We chose to develop a web-based system using the Joomla[1] content management system (CMS) framework. The specific CMS inherently covers a lot of the aforementioned features in its core and it has a plethora of extensions for users to install, such as a community building component for creating multi-user sites with blogs, forums and social network connectivity. Additionally, the CMS provides a very powerful component development engine, that enables developers to deploy additional elements that can be reused in multi-domain applications.

There are many cases where crowdsourcing applications require functionality from other systems or knowledge bases, e.g., automated reasoning engines, datasets and natural language processing systems. For the crowdsourcing platform we constructed an Application Programming Interface (API) to the Web-STAR system (cf. Chapter 4) for story understanding related processing and we offer a direct integration to the Stanford CoreNLP (Manning et al., 2014) system. It is also able to retrieve and process factual knowledge, from ConceptNet (Speer et al., 2017), YAGO (Suchanek et al., 2007) and WordNet (Fellbaum, 2010). Developers can integrate other SPARQL-based (Quilitz and Leser, 2008) knowledge bases since the methodology used is generic.

The crowdsourcing platform offers a number of features for promoting the application to groups of users, either in social media or user forums. Contributors can share their contribution status/points/awards to social media groups. This tactic can increase user

[1] https://www.joomla.org/

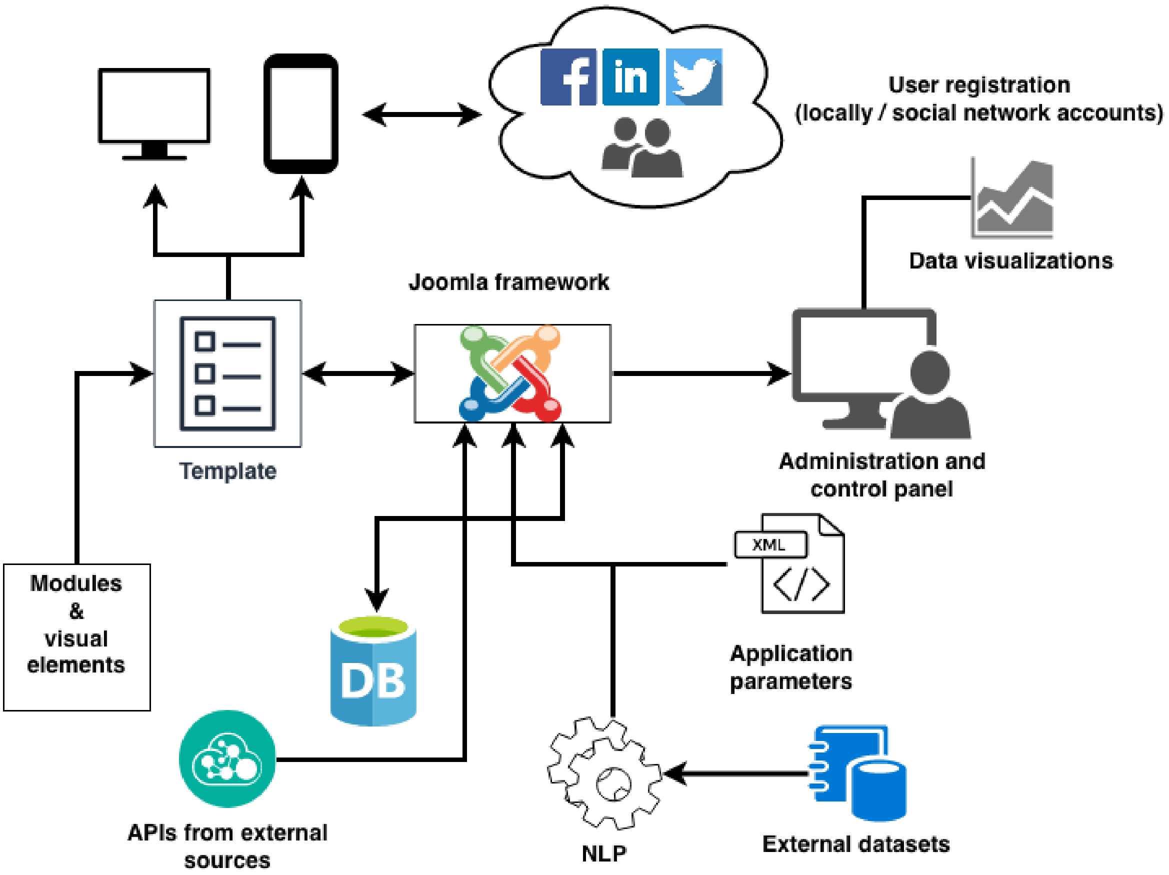

User registration
(locally / social network accounts)
Data visualizations
Joomla framework
Template
Administration and control panel
XML
Application parameters
Modules & visual elements
DB
APIs from external sources
NLP
External datasets

Figure 5.2 The architectural diagram of the Crowdsourcing platform, presenting the main components of the platform and the data flow between the components.

retention to the application. Moreover, developers can enable the "invitations" functionality, where contributors gain extra points when they invite other people to contribute.

5.2.2 Steps for Designing a Crowdsourcing Application Using the Platform

In this section, we showcase the steps needed for a developer to design and deploy a crowdsourcing application. These steps are also depicted in Figure 5.2. First, a template must be selected to match the application domain. There are a number of templates available to match a number of crowdsourcing paradigms (e.g., GWAPs, language learning applications) which can be customized according to the specific needs of the task.

Developers need to prepare the main functionality of their system by coding it in PHP, or any other language and encapsulate its executable in the platform and deliver the result using HTML, CSS and JavaScript. During this process, they need to prepare a list of parameters that can be used in the experiments and code it in XML format. These parameters can be incorporated in the code and control how various elements are displayed (e.g., display/hide web tour and guidance, choose what knowledge is presented for verification, etc.).

The next steps involve the selection of knowledge acquisition tasks. Developers can select among acquisition, verification and knowledge preference identification tasks and map the methodology steps to application screens or game missions (depending on the chosen paradigm). The knowledge preference selection task involves the ability of a human contributor to choose pieces of knowledge that are used in a given situation and discard the ones that are not. For example, when reading a story about birds, readers can infer that birds can fly. From a similar story, where it is explicitly mentioned that birds are penguins, readers can infer that penguins cannot fly.

For each task, a data stream is required. The data stream can be anything from text inserted directly from contributors, i.e, a dedicated task in the application, a pre-selected dataset such as Triangle-COPA (Maslan et al., 2015) or ROCStories (Mostafazadeh et al., 2016), or the outcome of another task.

Developers are free to design and code the logic behind each task as they see fit to achieve their goals. The platform has a number of pre-defined functions for storing commonsense knowledge in the form of rules or facts, both in natural language and in a logic-based format, e.g., `hug(X,Y)` implies `like(X,Y)` where `X` and `Y` are arguments and intuitively means if a person `X` hugs a person `Y` then person `X` likes person `Y`.

Moreover, the platform incorporates a number of visualization libraries (e.g., d3.js[2], Cytoscape.js[3], chart.js[4]) to provide live feedback to the contributor.

For each application, developers need to choose how contributed knowledge is selected and what are the criteria for storing this knowledge in the accepted knowledge pool. Developers can choose among a number of strategies or a combination of them, such as selecting knowledge that was contributed by at least n number of persons, knowledge that is simple (e.g., rules with at most n predicates in their body), knowledge that is evaluated/rated by at least n raters and knowledge that is evaluated by an automatic reasoning engine. Depending on the type of application, developers also need to choose a marking scheme that fits the logic behind the application and reward contributors, e.g., points and medals for games.

When the design of the various tasks is completed, the developer needs to choose how contributors will have access to the platform (e.g., anonymously, through registration or social networks) and what details need to be filled in their profiles.

[2] https://d3js.org/

[3] http://js.cytoscape.org/

[4] https://www.chartjs.org/

5.2.3 Technological Infrastructure

In terms of technological infrastructure, the platform relies on a web-server with Linux-Apache-MariaDB-PHP (LAMP) stack and on the Joomla framework. The platform also utilizes the JQuery[5] and the bootstrap frameworks both for designing elements and for application functionality.

Moreover, the platform employs the Joomla Model-View-Controller (MVC)[6] framework that allows the development of components by separating the data manipulation functions from the view controls. The controller is responsible for examining the request and determining which processes will be needed to satisfy the request and which view (presentation layer) should be used to return the results back to the user. This architecture allows the usage of both internal (e.g., database) and external data sources (e.g., APIs, files) and of course deliver these services in an abstraction layer that can be used by other applications.

For user authentication, both the Joomla internal mechanisms and the OAuth[7] authentication methods are used, that permit the seamless integration of social network authentication with the platform.

5.2.4 Data Visualization

An important and useful feature of a knowledge acquisition platform is the ability to visualize data. Application developers should be able to visualize acquired knowledge for better understanding what and how users behaved during the crowdsourcing experiment. In Figure 5.3 an example of a Sankey type graph is presented for the Robot Trainer game, presented in Section 5.4, where results for both the contributors and the acquired knowledge are depicted on the same diagram. This type of functionality is possible by using the d3.js library with data feed from the database and a graph theory (network) library for visualization and analysis called Cytoscape.js. Cytoscape.js was also used for representing and contributing commonsense knowledge rules in a graphical manner in Web-STAR (cf. Chapter 4, Section 4.3.4) and was evaluated positively by novice users in conjunction with using a text-based editor for the same task.

5.2.5 Acquisition Process Monitoring

For managing, controlling and monitoring the knowledge acquisition process, we have implemented an administration console for presenting information in real time and in visual form.

[5] https://jquery.com/

[6] https://docs.joomla.org/Model-View-Controller

[7] https://oauth.net/2/

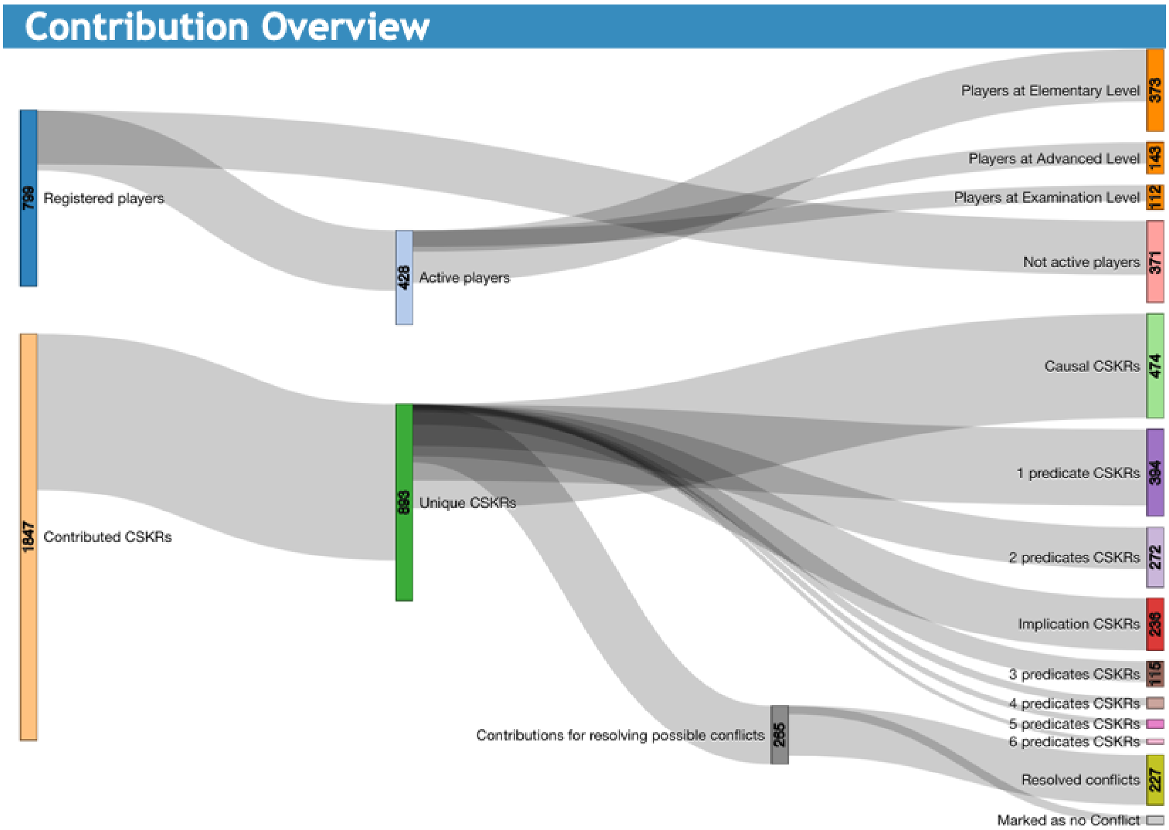


Figure 5.3 Screenshot of a data visualization diagram where readers can follow the data flow in the system for both players (top stream) and commonsense knowledge (bottom stream).

The administration console incorporates features for managing acquired data and preparing them for further processing. These features include CSK (commonsense knowledge) rules filtering, junk rules detection and experiment preparation. Researchers are able to control each experiment workflow, parameterize and monitor it, view the results and analyze them. This design, allows the use of existing infrastructure for security, presentation and integration with experiment data. Researchers can also set options for the experiment, like configuring the corpus used and choosing the reasoning engine by selecting an available webservice (e.g., the STAR system webservice).

Furthermore, there are options for filtering acquired CSK based on evaluation results, type, contributions, etc. These can be grouped in a custom setup option so that they can be reused by others. The platform keeps track of all actions and keeps data in a database where both backup, security and indexing features are enabled. The knowledge acquisition platform allows researchers to build a number of crowdsourcing applications for engaging human participants in contributing knowledge.

5.3 “Knowledge Coder” Game: A Fully Crowdsourced Approach for Knowledge Acquisition

In this section we concentrate on a fully crowdsourced method for acquiring CSK using a GWAP. We first analyze the formal framework used to represent and reason with the background knowledge, and our approach is compared to other existing works. The methodology used to gather background knowledge is then presented, as a sequence of steps needed to get from raw text to structured knowledge. We cast our methodology as a crowdsourcing task, and demonstrate how Games With A Purpose (GWAPs) can be used to implement it. Finally, an empirical setting and results from a deployment of our developed GWAP are presented. One can think of a story understanding engine as comprising three main modules:

- a module for converting stories from a given modality (e.g., text) to a formal representation
- a module for gathering background knowledge and representing it formally
- a module for reasoning by integrating story information with background knowledge

In an analogous context, Gordon and Schubert (2011) proposed a method for acquiring conditional knowledge by exploiting presuppositional discourse patterns to create general rules. Clark and Harrison (Clark and Harrison, 2009) developed a system able to extract simple statements of world knowledge from text, which aims to improve parsing and the plausibility assessment of paraphrase rules used in textual entailment.

In Chapter 2, a review of knowledge representation methods was presented and according to the literature, an appropriate method to represent knowledge is by using loose connections between concepts, following an argumentation type of representation as the one presented in Chapter 3, Section 3.2.2. This type of representation is in line with relevant psychological evidence (Kintsch, 1988; McNamara and Magliano, 2009).

5.3.1 Knowledge Acquisition Process

Following our main goal of investigating whether a fully crowdsourced approach suffices for knowledge acquisition, we propose a general scheme for going from raw text to background knowledge represented in terms of structured rules. We illustrate the steps of our methodology below, using the following simple story snippet as a running example:

> *Story snippet*: A cat chased the mice. The mice managed to hide in a nearby hole.

Step 1 A story is selected and is split into sentences, using punctuation marks to determine the end of each sentence. A sentence is then selected for processing. Human participants are asked to remove articles (e.g., "a", "the"), change the tense of verbs (e.g., "chased" to "chase") and lemmatize words (e.g., "mice" to "mouse"). This step converts sentences and words to a simpler form by reducing inflectional forms, and removing stop words.

Selected sentence: A cat chased the mice.
After processing: cat chase mouse

Step 2 Human participants are asked to identify nouns and verbs given the previously processed phrases. The outcome will be later used to produce formal expressions, which allow verbs being used as predicate name and nouns being used as predicate arguments.

Selected phrase: cat chase mouse
After separation: {cat, mouse} are nouns, and {chase} is a verb

Step 3 Predicates are constructed using verbs and nouns from the previous step. More specifically, human participants choose which verbs to use as predicate names and which nouns to use as predicate arguments. In addition to nouns, each constructed predicate can be used as an argument for new predicates that are created, leading to higher-order predicates. Human participants are required to choose whether a predicate is an action or a fluent.

Selected words: {cat, mouse} are nouns, and {chase} is a verb
Formal expression: chase(cat,mouse) is an action

Step 4 The next step seeks to identify logical rules that are built on the identified predicates. What is expected here is for the human participants to introduce new predicates that are not explicitly present in a sentence, but are implied by it, and relate those new predicates to the existing ones in the form of rules. For each rule, human participants are asked to specify whether this rule causes or implies the deduced predicate.

Selected predicate: chase(cat,mouse)
Possible rule 1: chase(cat,mouse) `causes` fear(mouse,cat)
Possible rule 2: chase(cat,mouse) `implies` can(cat,run)

Step 5 In the penultimate step, human participants generalize previously identified rules. For each rule certain predicates and arguments can be chosen and replaced with variables.

When an argument α is replaced with a variable V, a new predicate of the form $\alpha(V)$ is appended to the body of the rule. Human participants can choose whether this predicate should be retained.

Effectively, this step transforms each rule to a form that is applicable more generally and not only in the context of the story or sentence from which it originated.

Selected rule: chase(cat,mouse) `implies` can(cat,run)
Possible generalized rule 1: cat(X) `and` chase(X,mouse) `implies` can(X,run)
Possible generalized rule 2: chase(X,Y) `implies` can(X,run)

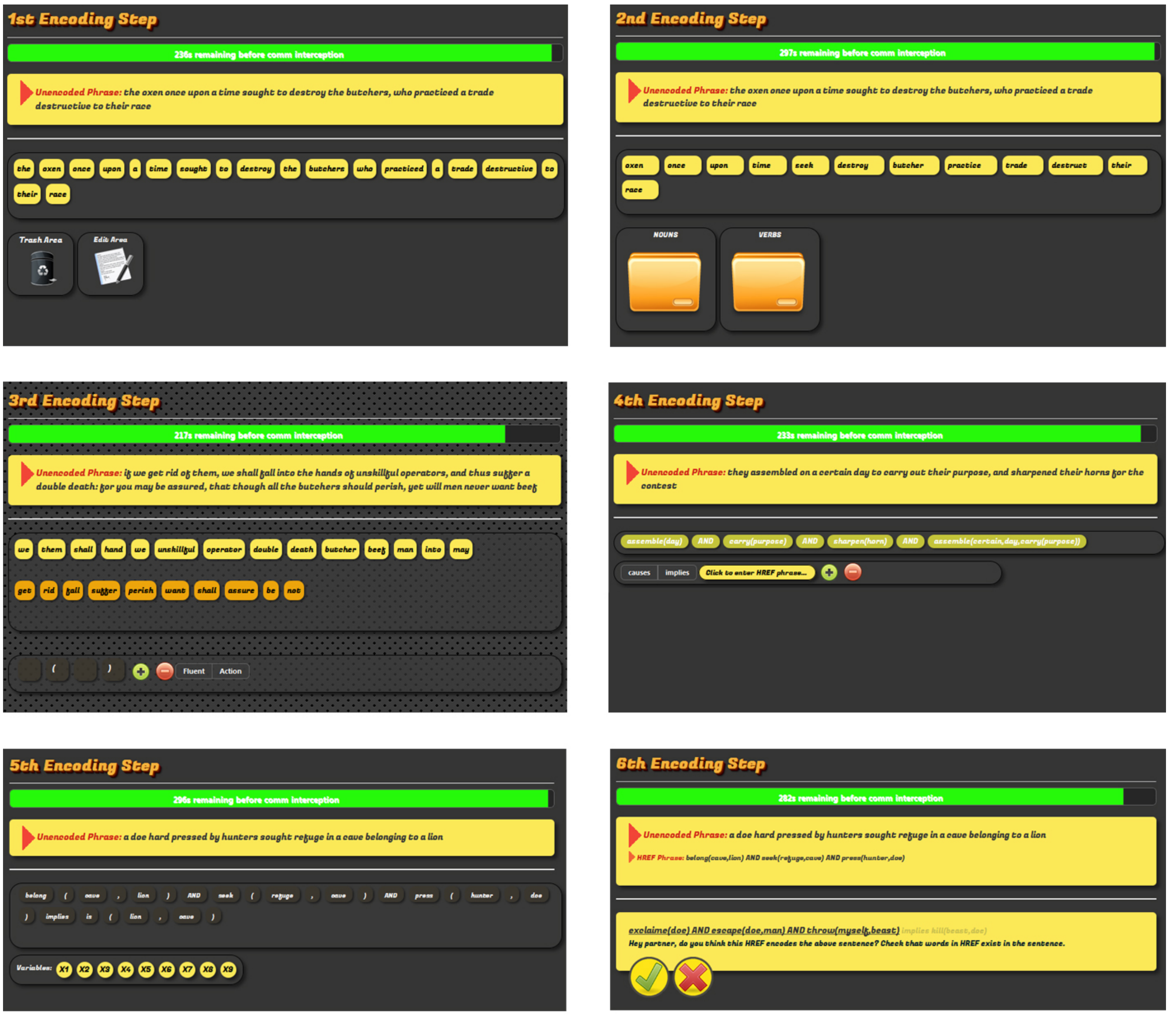
Step 6 During the final step, the acquired knowledge is validated. Firstly, a sentence other than the one from which a given rule originated, is selected. Human participants are asked to verify whether the conditions in the body of the rule are met in the context of the selected sentence. If they are, human participants are asked to decide whether the head of the rule follows from the sentence. If the player answers affirmatively to the first question then the rule receives a positive applicability vote; otherwise, the rule receives a negative applicability vote. If the player answers affirmatively to the second question then the rule receives a positive validation vote; otherwise, the rule receives a negative validation vote.

Selected context: A policeman was chasing a burglar near the town center.
Selected rule: chase(X,Y) `implies` can(X,run)
Results: The conditions in the body of the rule are met in the context of the selected sentence, and the head of the rule follows from the selected sentence. Thus, the rule receives a positive applicability vote and a positive validation vote.

After all six steps are completed, the resulting background knowledge comprises those rules that have been found to be sufficiently applicable and sufficiently validated.

In an ideal setup, human participants are knowledgeable, honest and willing to participate. This is not always the case and in games we may have players that try to cheat or provide misleading contributions on purpose. Our methodology provides measures to mitigate these negative effects of the actions of less knowledgeable or dishonest participants. One such measure is already present in the methodology. The multiple steps it comprises reduce the possibility of user error since players are focused on a single task at a time. Moreover, knowledge rules are validated by other players before entering the knowledge pool. Since the methodology provides multiple steps, one can choose to partially automatize some of these without the need to interfere with the other steps.

The proposed methodology is materialized through a GWAP we developed called "Knowledge Coder" and a prototype version is accessible online at: http://cognition.ouc.ac.cy/narrative. In Figure 5.4 the six game missions are depicted.



1st Encoding Step

236s remaining before comm interception

Unencoded Phrase: the oxen once upon a time sought to destroy the butchers, who practiced a trade destructive to their race

2nd Encoding Step

297s remaining before comm interception

Unencoded Phrase: the oxen once upon a time sought to destroy the butchers, who practiced a trade destructive to their race

3rd Encoding Step

217s remaining before comm interception

Unencoded Phrase: if we get rid of them, we shall fall into the hands of unskillful operators, and thus suffer a double death: for you may be assured, that though all the butchers should perish, yet will men never want beef

4th Encoding Step

233s remaining before comm interception

Unencoded Phrase: they assembled on a certain day to carry out their purpose, and sharpened their horns for the contest

assemble(day) AND carry(purpose) AND sharpen(horn) AND assemble(certain,day,carry(purpose))

5th Encoding Step

296s remaining before comm interception

Unencoded Phrase: a doe hard pressed by hunters sought refuge in a cave belonging to a lion

belong (cave , lion) AND seek (refuge , cave) AND press (hunter , doe) implies is (lion , cave)

6th Encoding Step

282s remaining before comm interception

Unencoded Phrase: a doe hard pressed by hunters sought refuge in a cave belonging to a lion

HREF Phrase: belong(cave,lion) AND seek(refuge,cave) AND press(hunter,doe)

exclaime(doe) AND escape(doe,man) AND throw(myself,beast) implies kill(beast,doe)

Hey partner, do you think this HREF encodes the above sentence? Check that words in HREF exist in the sentence.

Figure 5.4 Screenshots of the six game missions in the "Knowledge Coder" game.

Our approach falls into the output-agreement games template (von Ahn and Dabbish, 2008), requiring players to agree on the same output they produce. The game follows closely the methodology described in the previous section, with each step corresponding to a "mission" in the game.

The game story takes place in the near future, where planet Earth is captured by alien forces capable of intercepting human communications in natural language. Players are asked to join the resistance forces and help their co-defenders encode human knowledge in a structured form that is not readable by aliens, and thus guard it from being intercepted.

Players are introduced to a game environment containing a mission instructions area, a time countdown bar, a high scores area, and an active mission area. Players also have access to mission specific instructions and online help during game play.

As with other games, players are encouraged to play using competitive motives (Garris et al., 2002). For each successful mission attempt, players are rewarded with points that are added to their total score. Players are also rewarded with extra points when other players contribute and verify the former players' mission results and vice versa. These extra points are used to separate the knowledgeable and honest players from the rest. After a player reaches a certain score, an award is issued and added to the player's profile. These methods are commonly applied techniques to encourage and promote competition among players in games (Hamari and Eranti, 2011).

A common problem in online games is cheating through, for instance, communication between players outside the game (Mönch et al., 2006). To reduce such effects, missions are time-bounded to prevent players from using external help to complete them. The anonymity of players is pursued and no contact details are made available throughout the game play. Also, each player's Internet address is recorded and associated with each attempt on a mission, so that individual players masquerading as two or more different players are detected and are filtered out. Finally, every mission is initiated with a random sentence, so that the probability of two players attempting to work on the same instance of a task is minimized.

Players can provide feedback through the game interface. Feedback submitted is valuable both for debugging purposes and for further game development. Players can request new features, changes to the user interface, or extra missions, or suggest improvements.

5.3.2 Empirical Setting and Results

For our initial empirical evaluation of the game we prepared an evaluation process using a small group of people and two stories loaded into the game. Both chosen stories were short and used simple English words. For the purposes of this evaluation we selected two Aesop Fables: "The Oxen and the Butchers" and "The Doe and the Lion" (Aesop, 2009), depicted in Figure 5.5.

Five participants were trained on how to play the game on a test deployment of the game. This group included both men and women aged eighteen and above, all with a high school education, and with some of them enrolled in a university. All missions were presented and each player had the opportunity to familiarize themselves with the look and feel of the game. For the purposes of the experiment, each player created a game account. The game was available for one week, at the end of which each player was asked to complete a questionnaire. All knowledge gathered was analyzed, and our conclusions are presented below.

```
The oxen once upon a time sought to destroy the Butchers, who
practiced a trade destructive to their race. They assembled on a
certain day to carry out their purpose, and sharpened their horns
for the contest. But one of them who was exceedingly old (for
many a field had he plowed) thus spoke: "These Butchers, it is true,
slaughter us, but they do so with skillful hands, and with no
unnecessary pain. If we get rid of them, we shall fall into the
hands of unskillful operators, and thus suffer a double death:
for you may be assured, that though all the Butchers should perish,
yet will men never want beef."

A DOE hard pressed by hunters sought refuge in a cave belonging
to a Lion. The Lion concealed himself on seeing her approach, but
when she was safe within the cave, sprang upon her and tore her
to pieces. "Woe is me," exclaimed the Doe, "who have escaped from
man, only to throw myself into the mouth of a wild beast?"
```

Figure 5.5 The two Aesop fables used in the experimental setting of the “Knowledge Coder” game. “The Oxen and the Butchers” depicted at the top and the “The Doe and the Lion” depicted at the bottom.

Analysis of Results

We collected approximately one hundred user-generated rules; Table 5.1 presents some relevant information. Below we present and discuss a sample of the collected rules.

R1: horn(X) and assemble(X) and carry(purpose) and sharpen(X) and assemble(certain,X,carry(purpose)) implies have(ox,horns)

R2: assemble(day) and carry(purpose) and sharpen(horn) and assemble(certain,day,carry(purpose)) implies prepare(ox,war)

R3: beast(X) and throw(Y,mouth,X) implies kill(X,Y)

R4: beast(X) and man(Y) and doe(Z) and exclaime(Z) and escape(Z,Y) and throw(Z,X) implies kill(X,Z)

Number of stories	2
Number of sentences	7
Number of players	5

Number of rules generated	93
Number of causality rules	15
Number of implication rules	78

Table 5.1 Relevant information from the experimental deployment of the “Knowledge Coder” game.

As one can observe, rules R1 and R2 are too specific and tightly coupled to the story used to generate them ("The Oxen and the Butchers"). This level of specificity is inappropriate for gathering broad background knowledge. The metric of applicability can be used to filter such rules out. By requiring rules with high applicability, we are more likely to end up with rules like rule R3 which can be usefully applied in almost any story with wild animals. The fact that the majority of the rules produced by the first five steps of our methodology did not receive a high applicability score during the sixth step, suggests the need for an additional incentive in the game so that players produce simpler and more general rules. Such an incentive, for example, would allow players to suggest the deletion of predicates man(Y), doe(Z), exclaime(Z) and escape(Z,Y), from rule R4 to produce a rule similar to rule R3.

Note that rule R4 includes a misspelled predicate name (i.e., "exclaime" instead of "exclaim"), demonstrating that output-agreement does not guarantee that the gathered knowledge is error-free, and that additional incentives might be needed to reduce such errors.

Player Feedback

After completing the game, each player was asked to complete a questionnaire for assessing the game design, concept, usability, enjoyment and other factors such as playing time, game scoring, etc. Feedback was also requested on how well players understood the instructions given for each mission and the time needed for them to comprehend them before starting to play. Finally, players were asked whether missions are relevant to the game concept and what they would like to see changed for the game to become more engaging.

By analyzing this feedback we conclude that players found the game story interesting and that they would be willing to advertise the game to their friends. Most players found the first two missions (i.e., "sentence processing" and "verb and noun identification") easy to play and the instructions given informative. For the next two missions (i.e., "predicate construction" and "rule construction"), players seemed to require some time before understanding fully what they were expected to do. These two missions were also characterized as the most interesting ones and kept players engaged throughout the game play.

Four out of five responders characterized the fifth mission (i.e., "rule generalization") as not very challenging, since they understood that they only had to replace arguments with variables. On the one hand, this feedback suggests a misunderstanding on the part of the players on what they were expected to do, which can be avoided by improving the mission instructions. On the other hand, this feedback is in line with the acquisition of not highly applicable rules, which suggests the need for stronger incentives to simplify the rules.

Several of the comments received, concerned the creation of a tablet and mobile version of the game and integration with social media for posting score to the players' friends and contacts. One responder suggested that more languages should be available for the game.

5.3.3 Discussion on the Game

Designing an engine that can handle broad background knowledge for story understanding is far from being a trivial task, due to the fact that this knowledge is not given explicitly in the actual story text. The background knowledge gathered from our developed game, offers some encouraging results in terms of the feasibility of our methodology. Nevertheless, the problem of acquiring not highly applicable knowledge and knowledge that is specific is still an obstacle that we need to overcome. Moreover, an important enhancement to our methodology would be the addition of an extra step to denote preferences among conflicting knowledge as the one presented in Chapter 3, Section 3.2.2. This could also be reflected in the game in the form of an extra mission, after the currently last mission of "rule evaluation".

Some missions could also be partially automatized and use players to evaluate and correct the results of the process. For example, missions 1,2 and 3 could be partially automatized by using the natural language to symbolic format module of the Web-STAR IDE presented in Chapter 4, Section 4.3.3 or use a NLP system for mission 2 to identify nouns and verbs and then ask players to verify this. The results of this process, could then be presented to players to correct or verify them, instead of asking players to perform trivial tasks.

Although our work has centered on the task of knowledge acquisition for story understanding, we believe that our methodology is applicable more generally, and can find use in other lines of research that assume as given commonsense knowledge in a structured form.

5.4 "Robot Trainer" Game: A Hybrid Approach to Acquire Knowledge

In this section we take under consideration the results obtained from the "Knowledge Coder" GWAP and we proceeded with designing a new experiment which follows the approach that background knowledge can be acquired by harnessing the power of the crowd and combining it with efforts and work from knowledge engineers and machines. More specifically, we investigate techniques to acquire background knowledge in the form of commonsense knowledge (CSK) rules from short narratives using a GWAP and propose a specific methodology that allows the acquisition of CSK rules, the resolution of possible conflicts using CSK rule preferences and evaluation of the appropriateness of the acquired knowledge. We present an implementation of a GWAP developed called “Robot Trainer”, the experimental setup used for gathering CSK along with the results of this effort and an example of using the acquired knowledge to answer questions on unknown stories.

5.4.1 Knowledge Acquisition Process

We used the knowledge acquisition platform described previously in Section 5.2 to develop a second GWAP called “Robot Trainer”[8]. This game aims in harnessing human player activities for contributing CSK. A player takes the role of a teacher that aims in training a robot that will travel in deep space for a long journey, so as to avoid the destructive consequences of the death of our solar system. The trained robot will be able to transfer the human knowledge needed for the continuity of our species and culture in other planets, along with embryos that will evolve into humans after arriving in their new habitat.

The goal of the player is to teach the robot how to answer simple questions on short narratives by trying to explain the way we think for answering such questions. Players have to construct CSK rules using natural language phrases, help the robot resolve possible conflicts with these CSK rules and evaluate the appropriateness of their fellow players contributions. Players can join the game by creating an account using their email address or their social media accounts. When authenticated, players are redirected to the “Introduction screen” of the game. There, they get to view a short two minute introduction video, take the online tutorial, select a level to play or share their game status with others in social media.

Data and Game Mechanisms Selecting an appropriate dataset for a knowledge acquisition game is not a trivial task. We seeked for a dataset that has a predefined dictionary of terms and stories with situations that change through the course of time when certain events occur. Such a dataset is the Triangle-COPA which includes a set of one hundred short stories with animations and questions. These stories focus on the interactions between two triangles, a circle, and a box with a door. This dataset can be extended with more stories and animations using the Heider-Simmel Interactive Theater[9]. Each story is also accompanied by its representation in ISO-standard Common Logic Interchange Format[10], prepared by the authors of the dataset.

For using this dataset, we needed to convert each story, phrase and question in symbolic form. This is a very time consuming and prone to errors job, since it requires stories to be entered by hand by a knowledge engineer and that currently restricts the mass addition of

[8] The game is available online at http://cognition.ouc.ac.cy/robot.

[9] Heider-Simmel Interactive Theater is available online at http://hsit.ict.usc.edu

[10] https://standards.iso.org/ittf/PubliclyAvailableStandards/c066249_ISO_IEC_24707_2018.zip

new stories to the system and hence the scaling up of CSK acquisition. We also made some adjustments to the initial dataset, like changing predicates that were actually the negation of others (e.g., *unhappy* and *dislike* changed to *not happy* and *not like*) to reduce the number of predicates and help the automated reasoning engine. We selected a subset of that dataset that included twenty one narratives with a common theme. We randomly selected sixteen narratives for feeding the game’s database and five narratives that will later be used for evaluating the effectiveness of the acquired CSK rules. When a player constructs a CSK rule in natural language, it is automatically converted to symbolic language using the conversions entered initially by the knowledge engineer.

For generalizing CSK rules, we use the platform’s internal mechanism to substitute all instances of “shapes” in contributed rules with variables. These variables are of type person (e.g., person(big_triangle)) since in the Triangle-COPA dataset each shape actually behaves as a person.

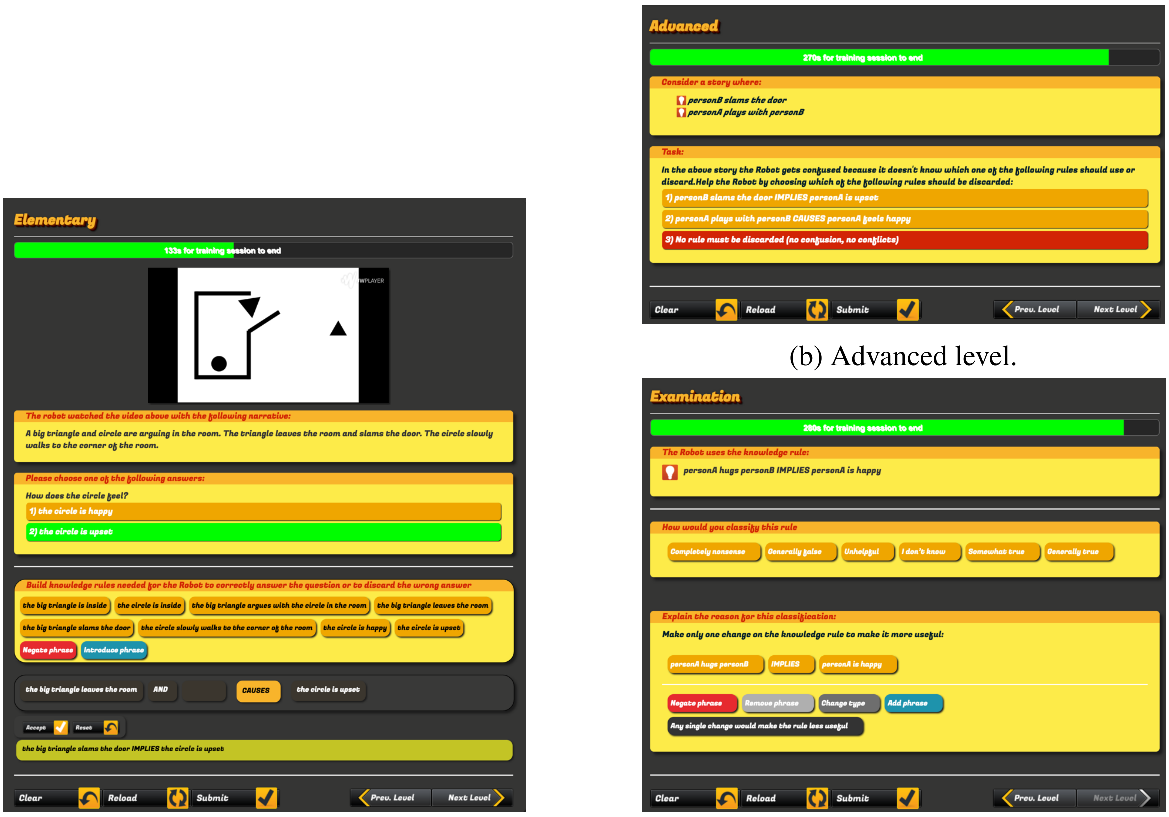
For the game to start, a player chooses one of the three available levels: Elementary (cf. Figure 5.6a), Advanced (cf. Figure 5.6b) and Examination (cf. Figure 5.6c). Any level can be chosen at any time and players are not required to complete a level before proceeding to the next one. We present each level in the next paragraphs, using real examples from the game in both natural language (NL) and symbolic language (SL) and screenshots of level specific information.

First Level (Elementary)

At the first level, a short story is randomly selected from the pool of available stories. Players read the short story which is accompanied by a short animation and then answer a question about that story. The next step is to build and submit CSK rules using phrases prepared by the knowledge engineers. Players can build CSK rules by dragging phrases on the body or the head of the rule (cf. Figure 5.6a). Before submitting the CSK rule, a player must choose whether it is a causal or an implication rule.

When players believe that the available phrases are not sufficient for building appropriate CSK rules that answer the question, they can search for new phrases (based on a predefined dictionary of 122 phrases) by typing the first three letters of the phrase, select the desired phrase template and then select the subjects involved (e.g., big triangle (BT), little triangle (LT), circle (C) etc.).

Narrative: The little triangle was limping.
Question: Why was the little triangle limping?



Elementary

133s for training session to end

The robot watched the video above with the following narrative:

A big triangle and circle are arguing in the room. The triangle leaves the room and slams the door. The circle slowly walks to the corner of the room.

Please choose one of the following answers:

How does the circle feel?

1) the circle is happy

2) the circle is upset

Build knowledge rules needed for the Robot to correctly answer the question or to discard the wrong answer

the big triangle is inside | the circle is inside | the big triangle argues with the circle in the room | the big triangle leaves the room | the big triangle slams the door | the circle slowly walks to the corner of the room | the circle is happy | the circle is upset

the big triangle leaves the room AND CAUSES the circle is upset

the big triangle slams the door IMPLIES the circle is upset

(a) Elementary level.

Advanced

270s for training session to end

Consider a story where:

- personB slams the door
- personA plays with personB

Task:

In the above story the Robot gets confused because it doesn't know which one of the following rules should use or discard.Help the Robot by choosing which of the following rules should be discarded:

1) personB slams the door IMPLIES personA is upset

2) personA plays with personB CAUSES personA feels happy

3) No rule must be discarded (no confusion, no conflicts)

(b) Advanced level.

Examination

280s for training session to end

The Robot uses the knowledge rule:

personA hugs personB IMPLIES personA is happy

How would you classify this rule

Completely nonsense | Generally false | Unhelpful | I don't know | Somewhat true | Generally true

Explain the reason for this classification:

Make only one change on the knowledge rule to make it more useful:

personA hugs personB | IMPLIES | personA is happy

Any single change would make the rule less useful

(c) Examination level.

Figure 5.6 Robot Trainer level screenshots.

Answers: [A] The little triangle is angry. [B] (*correct*) the little triangle is injured.
Rule (SL): limp(LT) `IMPLIES` injured(LT)

Second Level (Advanced)

Moving to the second level, players are instructed to help the robot in resolving possible conflicts when using specific pairs of overlapping CSK rules. These pairs are selected randomly by searching the pool of already acquired CSK rules. Next, we describe the selection algorithm in detail. Consider the following CSK rules: (a) BODY A `IMPLIES` HEAD A and (b) BODY B `IMPLIES` HEAD B. Then the following overlapping CSK rule pairs could lead to possible conflicts: (a) HEAD B = -HEAD A, (b) HEAD A exist in BODY B and (c) BODY B = BODY A.

A new story is created dynamically by using phrases from the overlapping CSK rules, and the player must decide if these rules are conflicting or not. If they are, the player must choose which of the two CSK rules should be discarded (less preferred), otherwise the player must state that this pair is not conflicting.

> **Narrative**: Person A is angry and Person A plays with Person B.
> **Possible conflicting rule 1 (NL)**: Person A is angry `IMPLIES` NOT_TRUE_THAT[Person A is happy]
> **Possible conflicting rule 2 (NL)**: Person A plays with Person B `CAUSES` Person A is happy
>
> **Player’s response**: Rule 1 is preferable to rule 2

Third Level (Examination)

The third level of the game is the Examination. Players are instructed to evaluate the appropriateness of CSK rules added by their fellow players for helping the Robot know which rules can be generally used and which are too specific. In Figure 5.6c, the game level is presented showing the evaluation options. When a player selects any of these: “Completely nonsense”, “Generally false”, “Unhelpful”, “I don’t know”, “Somewhat true” and “Generally true”, they are also asked to make one change to the CSK rule for making it more useful. Changes that are allowed are: add a phrase, remove a phrase, negate a phrase and change the rule type. There is also the option to proceed without doing any changes, for cases where any single change will make the rule less useful.

> **Rule (NL)**: Person A hits Person B `IMPLIES` Person A is angry at Person B
> **Player’s evaluation**: “Somewhat true”
> **Change proposed**: “add more phrases”

Whenever a player contributes a change on a specific CSK rule, this change is presented to a fellow player in the first level as a “tip” while building the same rule.

Help Facility We incorporated a number of help tools to the game for players to feel more comfortable in playing it. More specifically, players can read the intro of each level and then play a demo with guidance from the game itself. After doing so, they can choose to skip this step in next levels and enable it again if needed from their profile settings. Moreover, players are presented with the goals of each level throughout the game. They also have the option to view the online tutorial for a quick description of the game area, modules and controls. At any point, players have the option to contact us and provide feedback or report a bug of the game.

Player Incentives and Motives Robot Trainer is a GWAP and as with any other game of this type, players are motivated to play it for fun and of course to compete with other players. The game offers a flexible scoring framework for assigning points for various actions, like: contributing rules, contributing new rules, contributing rules that answer a specific true/false question, matching contributions of other players, contributing rules within a timeframe etc. Whenever a player contributes a CSK rule, the game automatically produces a STAR system program using the story information, the player contributed CSK rules and the story question in symbolic form. This program is sent via a webservice for execution to the Web-STAR IDE. When processing is completed, the results are returned to the game, the player receives a notification and the relevant points are added to the total score if the story question gets answered. Players can view a detailed score sheet for better understanding their score and prepare their game tactics. Besides the above score scheme, players are also informed and gain points when other players contribute CSK rules that match theirs. Players get real time information on where they stand compared to their fellow players and their progress in each mission, using the high score module. The most points are given for players that confirm other players' contribution.

Player Recruitment Recruiting players for the game is not an easy task. Fortunately enough, there are many players out there that are willing to try non-mainstream games and eventually contribute to the broader cause. To promote the game we used three channels, the university email to inform students at the university of the game, social media and game forums. Firstly, we announced the game to the university community and provided a short description and the link to play the game. Secondly, we created a FaceBook page at https://www.facebook.com/robotTrainerGWAP/ and launched a campaign to promote the game through FaceBook. Thirdly, we published articles about the game on forums that are specialized on GWAPs and non-mainstream games.

5.4.2 Empirical Setting and Results

The CSK acquisition experiment was active for a period of five months (153 days). During that period, players registered and played Robot Trainer GWAP. The following section presents the game, players and CSK rules analytics. Before the deployment of the game, we decided to have a short calibration period for 25 days, so that possible problems, bugs and minor improvements could be applied before deploying the final version of the game and running the experiment.

Calibration Period

During the calibration period, 24 people played the game and contributed 410 CSK rules, 182 of which were unique. The majority of contributed rules were implication (56%), whereas the causal rules were 44%. At that point, we suspected that the reason players contributed implication rules was because of it being the default option for building a new CSK rule. To verify our suspicion, we decided to change this setting to the final version of the game. At the end of the calibration period we interviewed the players to get a better understanding of how they understood the game and the different levels. From the interviews we concluded the following: (i) the first level (Elementary) was the most interesting for players, (ii) the second level (advanced) had a lot of information that was not necessary for completing the task and (iii) the third level (Examination) lack the option to select that nothing can be done to make the rule more useful. We changed the second level to make it easier for the players and we redesigned the third level for allowing evaluation of the appropriateness of the CSK rules. Also, we selected a scale similar to that of the developers of ConceptNet5 while evaluating the acquired CSK rules, so that we can compare our findings with theirs (Zang et al., 2013).

Experiment Period

In this section, we present the acquired data from the experiment period. These data include player analytics, CSK rules analytics, and examples of CSK rules acquired. We also present an example of using these CSK rules to answer multiple choice questions on unknown stories.

For the period of 153 days, 799 persons played the game from various regions of the world. More specifically, we had players from Asia (72.25%), Europe (11.10%), America (10.99%), Africa (4.29%) and Oceania (0.73%). This fact, along with the fact that the experiment was not conducted in a closed, supervised environment (e.g., a lab or classroom) allowed players to contribute CSK rules without researchers intervening or influencing players in this process. The majority of players preferred the first level (Elementary). Currently, more than 50% of the registered players contributed to the game on any level. On average, a player needed 2.08 minutes for contributing a CSK rule, 0.50 minutes for contributing a CSK rule preference and 0.42 minutes for evaluating the appropriateness of a CSK rule. In terms of average contributions, a player contributed 10 rules, resolved 7 conflicts and evaluated 13 rules.

During the experiment period, players contributed 1847 CSK rules, 893 of which were unique. A CSK rule is unique if there are not any other CSK rules with the same head and body in the acquired CSK database. CSK rules with the same head and body but with different order of predicates are not considered unique. Another important finding, is that players chose to contribute simple CSK rules (i.e., rules with only one predicate at the body).

Over 74.60% of the acquired unique CSK rules had a maximum of two predicates at the body of the rule. The type of acquired CSK rules is another metric we took into consideration. The majority of CSK rules contributed (67.30%) were causal. Comparing this result to that of the calibration period, we observe that most players followed the default option set while contributing CSK rules.

Robot Trainer also collects CSK rule preferences between possible conflicting pairs. From the 893 unique CSK rules we have gathered, we detected 31199 overlappings that could lead to possible conflicting pairs (cf. Section 5.4.1). The majority of overlapping CSK rules (60.37%) were of type B (a predicate at the head of one rule is present in the body of the other). Players were presented with some of these pairs during the second level of the game. Players contributed on resolving 1053 (371 unique) of them. From those 371 pairs, 52 (14.02%) were reported as not possible to lead to conflicts.

Players also evaluated a number of CSK rules while playing the third level. For 1847 contributed CSK rules, players provided 1501 evaluations. For better filtering and presentation of the results, we grouped "Somewhat true" and "Generally true" as "Positive" evaluations, "Completely nonsense", "Generally false", "Unhelpful" as "Negative" evaluations and "I don't know" as "Neutral" evaluations. When a CSK rule has equal number of "Positive" and "Negative" evaluations, it is considered as "Neutral". In terms of unique CSK rules, 415 (46.47%) of 893 CSK rules or 350 (39.19%) if "Neutral" answers were ignored, were evaluated by at least one evaluator. Players evaluated 221 (63.14%) CSK rules as "Positive" out of definite responses (i.e., the responses discarding "Neutral" evaluations).

In a similar evaluation process, Witbrock et al. (2005) reported that reviewers marked 7.5% of the acquired CSK rules as "correct" and 35% as "correct with minor adjustments". Moreover, when comparing these results with results from the evaluation of ConceptNet 5 (Zang et al., 2013), our methodology lays at the middle of the range (60%-70%) of facts gathered from WordNet, Wiktionary (English-only), and Verbosity in ConceptNet database, that were reviewed by evaluators. Comparison between the two systems cannot lead to safe conclusions, since ConceptNet deals with gathering CSK of different type.

In terms of CSK rule evaluation speed, Robot Trainer allows the evaluation of 143 CSK rules per hour. In similar measurements (Witbrock et al., 2005), a reviewer evaluated 20 CSK rules per hour.

Furthermore, players added a "Positive" evaluation to simple CSK rules (i.e., rules with one or two predicates in the body) instead of more complex ones. More specifically 62.07% of the "Positive" evaluated CSK rules had one predicate and 22.99% had two predicates. Figure 5.7 presents an overview of the acquired CSK rules and shows that most contributors prefer building simple CSK rules (orange and blue lines).

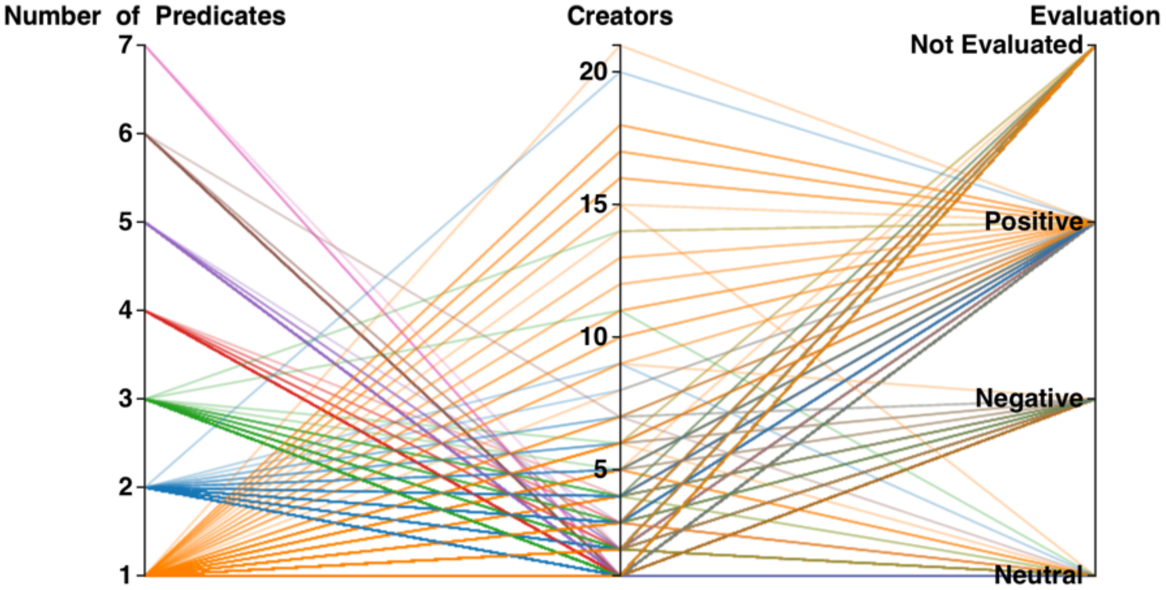


Figure 5.7 An overview of the acquired CSK rules per number of predicates, creators and evaluations.

Examples

In this section, we present some examples of CSK rules acquired during our experiments.

R1: injured(A) `CAUSES` limp(A) **R4**: pull(A,B) `CAUSES` -happy(B)
R2: hug(A,B) `IMPLIES` like(A,B) **R5**: hug(A,B) `CAUSES` happy(A)
R3: hit(A,B) `IMPLIES` angry(A) **R6**: argueWith(B,A) `CAUSES` -happy(A)

R1 was contributed by 21 players and evaluated by 12. 58.33% evaluated this CSK rule as “Somewhat true” and “Generally true”. R2 was contributed by 17 players and evaluated by 35. 85.71% evaluated this CSK rule as “Somewhat true” and “Generally true”. R3 was contributed by 16 players and evaluated by 24. 79.16% evaluated this CSK rule as “Somewhat true” and “Generally true”.

R4 is an example of a not so useful CSK rule gathered during the acquisition process. It was contributed by 6 players and evaluated by 19. 26.31% evaluated this CSK rule as “Somewhat true” and “Generally true”. This CSK rule most probably would not be included in any knowledge database due to its low evaluation score.

In terms of CSK rule preferences acquisition, we consider the CSK rules R5 and R6. Five players contributed on resolving possible conflicts between R5 and R6. More specifically, players were presented with a short story where B argues with A and A hugs B. 60% reported that R5 is preferable to R6.

Table 5.2 Processing results for the 5 narratives. [R], [A] and [?] indicate that the answer is rejected, accepted and possible respectively. The answer in bold text is the correct one.

Narrative title	Question/Answer	Process time
Date night	**[A] friend(LT,C)** | [?] stranger(LT,C) and stranger(C,LT)	0.45 min
Cold outside	[?] -happy(LT) | **[A] cold(LT)**	0.23 min
Run and hug	[R] -happy(BT) | **[A] excited(BT)**	1.10 min
Argue & Trudge	**[A] -happy(LT)** | [R] happy(LT)	0.27 min
Punch wall	**[?] -goal(angry(BT),BT)** | [A] excited(BT) and happy(BT)	3.83 min

Question Answering Using the Acquired CSK

The acquired CSK rules can be used to answer questions on unknown narratives. For demonstrating this, we prepared the following experimental setup: Firstly, we used the 5 randomly selected narratives from the Triangle-COPA dataset which were not seen by the game players. Each of these stories was accompanied by a multiple choice question with 2 possible answers. Then, we created a knowledge pool using CSK rules acquired previously using the Robot Trainer GWAP. More specifically, we selected CSK rules that were evaluated by at least 2 evaluators, the majority of the evaluators added a "Positive" evaluation and they had a maximum of 4 predicates in the body of the CSK rule. We also used CSK rule preferences that were chosen by the majority of the contributors.

We aim in correctly answering as many questions as possible using only the CSK acquired from players and the STAR system. The STAR system returns three possible results for each question: "accepted", "rejected" and "possible". A question is answered if any of the following conditions are met: 1) the STAR system responds with a different definite result ("accepted" or "rejected") for both answers or 2) the STAR system responds with a definite result ("accepted" or "rejected") for one of the 2 possible answers and the result for the other answer is "possible". Responses are the result of the STAR system reasoning process, that finds arguments to support or defeat a possible answer. From the 5 narratives processed with the automated reasoning engine, we retrieved answers to all questions. From the 5 questions, we retrieved correct answers for the 4 of them. Details of the processing procedure are depicted in Table 5.2.

For better understanding the question answering process, we present the "Argue and trudge" narrative example with a subset of the CSK rules used in the reasoning process.

> **Narrative (NL)**: The little triangle wants to go out and party with its friends but it's mom wants it to do its homework. The little triangle goes to sulk in the

corner.
Narrative (SL): argueWith(BT,LT) at 1, inside(LT) at 1, moveTo(LT,corner) at 2.
Question (NL): Why does the little triangle trudge to the corner of the room?

Answers (SL): (A) -happy(LT) at 4 or (B) happy(LT) at 4

The following rules are used for building the argument to support that the little triangle is happy at time point 4.

R1: fight(X,Y) implies -happy(Y)
R2: argueWith(Y,X) implies fight(Y,X)
R3: argueWith(Y,X) implies -happy(X)
R4: argueWith(Y,X), moveTo(X,corner) implies -happy(X)

5.4.3 Discussion on the Game

In this section, we presented a hybrid methodology and a knowledge acquisition platform that bridges three different approaches of gathering CSK; knowledge engineers, automated reasoning and crowdsourcing. We presented a GWAP developed using the platform components and the results of the acquisition process. Results of this methodology are comparable to other systems, with the difference that this system is not only used to gather CSK in the form of rules, but it also gathers CSK rule preferences and evaluates a CSK rule appropriateness. Acquired CSK can be used for story understanding tasks (Mueller, 2003), question answering systems and more complex applications like cognitive agents.

This game is an example of how a graphical interface can be used for acquiring commonsense knowledge from untrained contributors. Similar to the Web-STAR IDE presented in Chapter 4, where a visual interface facilitates non-expert users in adding background knowledge for story comprehension, the Robot Trainer GWAP provides the means for untrained users to contribute commonsense knowledge in symbolic format that can be used by automated story understanding systems.

One of the main challenges in deploying this methodology in large scale is the problem of automating story conversion from natural language to symbolic language. In this implementation of the game the process of transforming the Triangle-COPA dataset to symbolic language suitable for the STAR system, required a number of changes to the original dataset and hence added overhead to the overall effort needed.

5.5 Discussion

In this chapter we have investigated implicit methods for knowledge acquisition using crowdsourcing. Both experiments show that crowdsourcing is an approach which can be used for acquiring commonsense knowledge. Fully crowdsourced approaches provide more flexibility in knowledge acquisition, but limit the applicability of knowledge for automated story understanding systems.

We provided evidence that a hybrid methodology is applicable for acquiring knowledge that is simple and general so as to be used in answering questions on unseen stories.

The experiments showed that games are appropriate interfaces for knowledge acquisition. Using games, one can take a task that is not that interesting, such as contributing commonsense knowledge, and engage untrained players in doing it. This is obvious by the large number of players registered for the Game (799) and the amount of knowledge rules contributed (1847).

Furthermore, the experiments showed that contributed crowdsourced knowledge can be used for automated question answering using an argumentation-based reasoning system, such as STAR. The fact that our methodology is able to acquire commonsense knowledge encoded in symbolic format and commonsense knowledge rule preferences made it possible to use this knowledge for question answering, where the answers are inferred and are not explicitly found in the story text.

When our work is compared to existing work on crowdsourcing approaches for acquiring commonsense knowledge, one can easily identify that the presented GWAP is able to acquire knowledge both in natural language and in symbolic format, whereas the majority of available GWAPs are focused in acquiring knowledge in natural language only which requires additional steps for processing it. Furthermore, our methodology does not rely only on human evaluation, but it also uses an automated story understanding system to check the contributed knowledge in near real-time and it also tests the acquired knowledge on answering questions on unseen stories, where their answer is not explicitly mentioned in the story text.

One of the main contributions of our work is the commonsense knowledge rules dataset[11] that complements the Triangle-COPA dataset. Rules contained in that dataset are contributed and verified by the game players and are also in suitable format for using them as background knowledge of automated reasoning systems, such as the Web-STAR IDE presented in Chapter 4. Moreover the representation of knowledge is in a suitable format that is both machine and human readable.

[11] The dataset is available at https://cognition.ouc.ac.cy/robot

In future versions of the game, some changes to the mechanism that selects the CSK rules that will be evaluated or presented for possible conflict resolution could be applied. Currently, we use a random selection algorithm, but this mechanism allows specific CSK rules to be evaluated by many, whereas others are not evaluated. This happens when a CSK rule is introduced early in the game or by many, and players are presented with this rule more often. The solution to this, is to change the selection algorithm to present CSK rules that have the lowest number of evaluations first.

Future work could also include using the games in other domains, like teaching. The games can be used as a learning activity for non-English language speakers. The use of simple English phrases employed in the games' missions is ideal for practicing while studying English language courses.

Related Publications:

(1) Christos T. Rodosthenous and Loizos Michael. A Platform for Commonsense Knowledge Acquisition Using Crowdsourcing. In Katerina Zdravkova, Karën Fort, and Branislav Bédi. Supplementary Proceedings of the enetCollect WG3 & WG5 Meeting 2018, pages 24–25, Leiden, Netherlands, 2018. CEUR.

(2) Christos Rodosthenous and Loizos Michael. A Hybrid Approach to Commonsense Knowledge Acquisition. In Proceedings of the 8th European Starting AI Researcher Symposium (STAIRS 2016), pages 111–122, Hague, Netherlands, 2016. IOS Press

(3) Christos T. Rodosthenous and Loizos Michael. Gathering Background Knowledge for Story Understanding through Crowdsourcing. In Proceedings of the 5th Workshop on Computational Models of Narrative (CMN 2014), volume 41, pages 154–163, Quebec, Canada, 2014. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

6

Crowd-contributed Ontologies for Text Comprehension

> *"It is the mark of an educated mind to be able to entertain a thought without accepting it."*
>
> – Aristotle, *Metaphysics*

6.1 Introduction

In this chapter we examine knowledge acquired using crowdsourcing and resides in popular knowledge-bases and ontologies, while in the meantime we try to tackle the problem of identifying the geographic focus of a text document. For a machine to perform this task, it needs to process the text, identify location mentions from the text, and then try to identify its geographic focus. In Chapter 2 and more specifically in Section 2.2 we presented a number of systems and their respective mechanisms used to identify the geographic focus. The majority of the systems developed in this line of research rely on gazetteers, atlases, and dictionaries with geographic-related content, that identify the geographic focus of the text. We furthermore investigate whether generic ontologies can be exploited for tackling this problem with a special focus on cases where no explicit mention of the target country exists in the text. Our research in this area also includes crowdsourcing experiments for evaluating the usefulness of the acquired knowledge in identifying the geographic focus of a text document.

The problem of identifying the geographic focus of text is tackled using computational argumentation methods (cf. Chapter 3). This approach involves treating RDF triples from ontologies and knowledge-bases as arguments that support a specific country as being the geographic focus of a text document. Our proposed methodology views existing knowledge-bases as collections of arguments in support of particular inferences in terms of the geographic focus of a given story. We also associate these arguments with weights — computed through crowdsourcing — in terms of how strongly they support their inference. Moreover, we use both textual query answering strategies and acceptance semantics to eventually return a prediction of the geographic focus.

We developed a system called **GeoMantis** to identify the country-level focus of a text document or a web page using knowledge from generic ontologies. In particular, the system takes as input any type of document, processes it, and it stores the contents of the document in a database. Independently of the previous process, the system retrieves knowledge from ontologies about countries, processes and filters it using its internal mechanisms, and stores it in a database.

The system treats RDF triples from ontologies that reference a particular country as arguments that support that country as being the geographic focus of a text that triggers that argument. In this workflow, a full-text search algorithm is used for matching each search text of the document against the search text of each triple in the country's knowledge base set. A number of filtering options are also available during this process.

The outcome of the above-mentioned search process is the set of country arguments that are activated by the document text. This outcome is used in the query answering process to produce a list of countries in order of confidence. The ordering of this list is performed using one of the four supported by the system strategies that will be presented in detail later in this work. There is also the option to use computational argumentation frameworks such as Dung's Abstract Argumentation Framework (cf. Chapter 3) and its corresponding semantics to handle acceptance of these arguments.

In the following sections, we present the system we have developed to perform this task. The GeoMantis system is presented, followed by a detailed presentation of the architecture and its components. Next, an evaluation of the system is performed on how well it identifies the geographic focus of a text document using several datasets and parameters. There is a separate section for presenting the results of the parameter selection process and the comparative evaluation of the system. In the penultimate section we describe an extended evaluation of the system using crowdsourcing for evaluating arguments and comparing the results with the ones in the previous experiment. In the final section, new features and possible extensions to the GeoMantis system are discussed.

6.2 The GeoMantis System

GeoMantis (from the Greek words Geo that means earth and Mantis, which means oracle or guesser), is a web application designed for identifying the geographic focus of documents and web pages at a country-level. Users can add a document to the system using a web-interface. The document enters the processing pipeline depicted in Figure 6.1 and gets processed.

The system uses knowledge in the form of arguments that support a specific country. These arguments are generated from Resource Description Framework (RDF) (Lassila and Swick, 1999) triples retrieved from ontologies (e.g., ConceptNet and YAGO). These triples are of the form <`Subject`> <`Relation`> <`Country Name`>, where the `Subject` has a relationship `Relation` with the `Country Name`. This represents the argument that when the text <Subject> is included in a given document, then the document is presumably about country <Country Name>. Detailed information on the RDF semantics can be found in the W3C specification document (Hayes and McBride, 2004). Triples and generated arguments are stored locally in the system's geographic knowledge database. This database can be updated at any time by querying the corresponding knowledge source online.

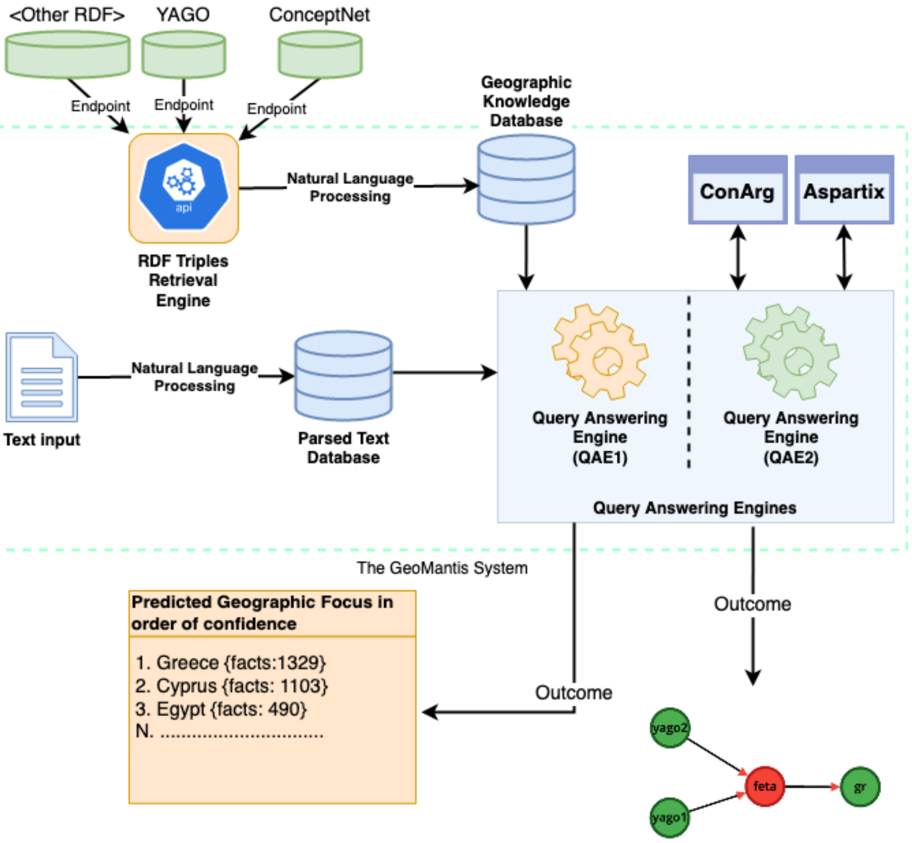
Retrieved arguments from ontologies are used for searching in each document and generate the predicted geographic focus. Instead of returning only one prediction for the target country, the system returns a list of countries in order of confidence for each prediction. Countries in the first places have a higher confidence score.

The system can be tuned using a number of parameters such as the selected ontology, the query answering strategy (cf. Section 6.2.3), and text filtering options (e.g., stopwords and named entities).

In the next paragraphs, we present how the GeoMantis system pipeline works.

6.2.1 Text Input Parsing

First, users upload a text document or type a webpage URL through a web interface. This text is firstly cleaned from HTML tags (e.g., <br>, <b>, <p>, <div>) and wiki specific format (e.g., [[Link title]]). Then, the text is parsed using a Natural Language Processing (NLP) system, the Stanford CoreNLP (Manning et al., 2014); extracted lemmas, part of speech, and named-entity labels extracted by the Named Entities Recognition (NER) process are stored and indexed in the system's database. The NER system can identify named entities of type location, person, organization, money, number, percent, date and time, duration, and miscellaneous (misc).


<Other RDF>
YAGO
ConceptNet
Endpoint
Endpoint
Endpoint
Geographic Knowledge Database
api
Natural Language Processing
RDF Triples Retrieval Engine
ConArg
Aspartix
Natural Language Processing
Text input
Parsed Text Database
Query Answering Engine (QAE1)
Query Answering Engine (QAE2)
Query Answering Engines
The GeoMantis System
Outcome
Outcome
yago2
yago1
feta
gr

Predicted Geographic Focus in order of confidence

1. Greece {facts:1329}
2. Cyprus {facts: 1103}
3. Egypt {facts: 490}
N. ....

Figure 6.1 The GeoMantis system processing workflow. The workflow includes the RDF Triples Retrieval and Processing Engine (top left), the Text Processing mechanism and the Query Answering Engines, QAE1 & QAE2. The outcome of the system based on each query answering engine appears on the bottom. QAE1 results in the predicted list of countries based on confidence and QAE2 outcome is a list of candidate countries based on the acceptance semantics of the argumentation framework used.

6.2.2 Knowledge Retrieval

The RDF triple retrieval process starts by identifying each country's official name and alternate names from the GeoNames database[1]. Geonames is a geographical database that includes more than 10 million geographical names. It also contains over 9 million unique features where 2.8 million are populated places and 5.5 million are alternate names. The database is integrating geographical data such as names of places, alternate names in various languages, elevation, population, and others from various sources. Sources include, among others, the National Geospatial-Intelligence Agency's (NGA), the U.S. Board on Geographic Names and the Ordnance Survey OpenData.

[1] http://www.geonames.org

The system retrieves triples by using an available SPARQL endpoint for every ontology integrated with the system. SPARQL (Quilitz and Leser, 2008) is a query language for RDF that can be used to express queries across diverse data sources. SPARQL contains capabilities for querying RDF graph patterns and supports extensible value testing and constraining queries by source RDF graph. The outcome of a SPARQL query can be result sets or RDF graphs. In Figure 6.1 (left part), the integration of the system with a number of ontologies is presented. GeoMantis is capable of retrieving RDF triples from any ontology that exposes a SPARQL endpoint and represents factual knowledge in RDF triples.

The final step in the knowledge retrieval workflow, is the processing of the retrieved RDF triples using the CoreNLP system. The object part of the triple is tokenized and lemmatized, and common stopwords are removed. For each RDF triple in the system's geographic knowledge base, a search string is created with lemmatized words.

Algorithm 2 presents the knowledge retrieval process. The SPARQL query created in line 6 of Algorithm 2 is used to retrieve the RDF triples and it is of the form: `SELECT * WHERE {<Countryname> ?p ?o}` when the country name is in the subject of the triple, and `SELECT * WHERE {?p ?o <Countryname>}` when the country name is in the object of the triple.

From each retrieved RDF triple, a search text is created using tokenization, lemmatization, and stopword removing techniques. The search text is stored in the GeoMantis local database.

6.2.3 Query Answering Engines

GeoMantis currently supports two query answering engines that are able to return the geographic focus of the text best on the above inputs. The first (QAE1) is able to handle simple strategies and information retrieval algorithms and the second (QAE2) uses computational argumentation methods to return the outcome of the identification process. In the following paragraphs we provide a description for both engines.

Query Answering Engine (QAE1)

For each country, a case-insensitive full-text search is executed for each unique word in the text against the search text of each argument in the country's knowledge base. An argument is activated when a word from the document exists in the argument's processed text using a full-text search (excluding common stopwords). For example, a document containing the sentence "They had a really nice dish with halloumi while watching the Aegean blue." should activate the arguments: 1) "When the text `halloumi` is found in the document, then the document is presumably about country `Cyprus`", retrieved from the triple `<halloumi>`

Algorithm 2 Knowledge retrieval from ontologies.

```
procedure RetrieveKnowledge(KB)
// Use the ISO two-letter country code
    for each countryCode in countryCodes do
        countryNames ← RetrieveNames(countryCode)
        for each countryName in countryNames do
            while $N \in \{subject, object\}$ do
                SPARQLquery ← CreateQuery(countryName,N)
                triples ← RetrieveRDFTriples(SPARQLquery)
                for each triple in triples do
                    if N="subject" then
                        arg1 ← GetPart(subject,triple)
                        arg2 ← GetPart(object,triple)
                    else
                        arg1 ← GetPart(object,triple)
                        arg2 ← GetPart(subject,triple)
                    end if
                    relation ← GetPart(predicate,triple)
                    searchText ← arg2
                end for
// Use NLP to tokenize and lemmatize
                searchText ← NLP(searchText)
// Use a common stopwords list
                searchText ← ClearStopWords(searchText)
            end while
        end for
        SaveGeoDatabase(searchText,countryCode)
    end for
end procedure
```

<RelatedTo> <Cyprus> and 2) "When the text Aegean is found in the document, then the document is presumably about country Greece", retrieved from the triple <Greece> <linksTo> <Aegean_Sea>. To maximize the search capabilities, the GeoMantis system uses lemmatized words. Full-text searching takes advantage of the MariaDB's[2] search functionality, using full-text indexing for better search performance.

The final step in the query answering process, involves the ordering of the list of countries and the generation of the predicted geographic focus. Ordering is performed using one of the following strategies:

[2] https://mariadb.org/

Percentage of arguments applied (PERCR): List of countries is ordered according to the fraction of each country's total number of activated arguments over the total number of arguments for that country that exist in the geographic knowledge bases, in descending order.

Number of arguments applied (NUMR): List of countries is ordered according to each country's total number of activated arguments, in descending order.

Term Frequency - Inverse Document Frequency (TF-IDF): List of countries is ordered according to the TF-IDF algorithm (Manning et al., 2008), which is applied as follows:
D_c is a document created by taking the arguments of a country c
TF_t = (Number of times term t appears in D_c) / (Total number of terms in D_c)
IDF_t = $\log_e$(Total number of D_c / Number of D_c with term t in it).

Most arguments per country ordering (ORDR): List of countries is ordered according to the number of arguments that are retrieved for each country, in descending order.

Query Answering Engine (QAE2)

The task of identifying the geographic focus of a text document can also be seen as an argumentative process, where two or more agents read a text document and argue about its geographic focus by providing words and phrases in the document that are linked to a specific country and by providing counter-arguments for the same phrase in the document. Both agents use the same knowledge base (e.g., YAGO). This argumentative process is repeated for all candidate countries.

As we have already discussed in Chapter 3, argumentation is a paradigm that could be used for identifying the geographic focus since this is a type of problem that does not have a specific solution (Freeley and Steinberg, 2013) and the goal is to present evidence towards a decision (Tindale, 2007) on which of the candidate countries is the geographic focus of a text document. Argumentation methods are also used in the work of Cabrio et al. (2017), where a framework called RADAR (ReconciliAtion of Dbpedia through ARgumentation) is presented using a fuzzy bipolar argumentation framework to reconcile information from the language-specific chapters of DBpedia.

To better understand how argumentation is used in the context of geographic focus identification we present the following example of such an argumentative process (in parentheses we name each argument):
Agent 1: This document's geographic focus is country C_1 ($c1$).
Agent 2: This document's geographic focus is country C_2 ($c2$).

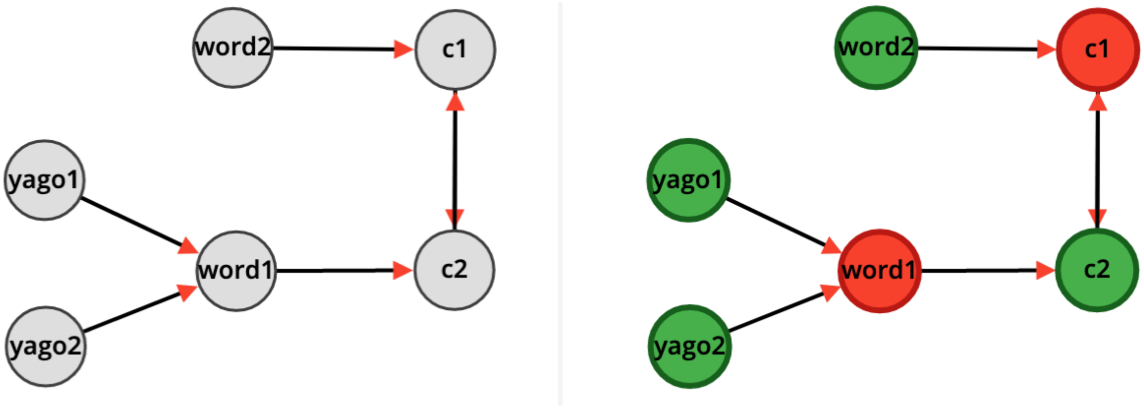


Figure 6.2 On the left site: The directed graph of the AAF produced from the conversation, depicting arguments as circles and arrows pointing to arguments as attacks. On the right site: Arguments highlighted in green are accepted under the stable, grounded and complete semantics of Dung's AAF.

Agent 1: The document mentions $word1$ which is related to country C_1 ($word1$).
Agent 2: But the document mentions $word2$ which is related to country C_2 ($word2$).
Agent 1: Word $word1$ is related to country C_1 since this is linked to $Evidence_1$ ($yago1$).
Agent 1: And word $word1$ is also related to country C_1 since this is linked to $Evidence_2$. ($yago2$).

The above dialog can be represented in an AAF using the directed graph depicted in Figure 6.2. Arguments $c1$ and $c2$ attack each other. $word2$ is an argument attacking c_1 and it is not attacked by any other argument. Arguments $yago1$ and $yago2$ attack $word1$. The grounded semantic (cf. Chapter 3) is used to determine the outcome of the argumentation process, which in this case is the set $\{c2, yago1, yago2, word2\}$.

Moreover, a weighted argumentation extension of Dung's AAF can be used to associate attacks with a weight, indicating the relative strength of the attack (Dunne et al., 2011). This extension is useful to indicate how much tolerance we are willing to accept to get solutions by tweaking the relevant parameters. The weight in each attack could represent: measures of votes in support of attacks, weights as measures of the inconsistency of argument-pairs, and weights as rankings of different types of attack. These of course are just examples and more meanings of weights can be found. In this work we use weights originating from crowdworkers' evaluations.

6.2.4 System Implementation

The GeoMantis system is built using the PHP web scripting language and the MariaDB database for storing data. The system is designed using an extendable architecture which allows the addition of new functionality. There are five distinct interfaces for the system and each one serves a different purpose. Firstly, there is a demo interface (cf. Figure 6.3) where users can test the ability of the system with a predefined strategy and a number of example stories. Secondly, there is an experiment interface (cf. Figure 6.4) where researchers are able to tweak the parameters, the strategy, the QAE and compare the results with other systems which are connected through APIs. Thirdly, there is an interface for analyzing the results of each experiment or crowd evaluation process (cf. Figure 6.5) and visualize them in easy to read graphs. Fourthly, there is an interface for evaluating arguments using crowdsourcing (cf. Figure 6.13), allowing direct integration with services such as Amazon Mechanical Turk and microWorkers. Fifthly, there is an API for direct accessing the core functions of the GeoMantis system and utilize them in various apps.

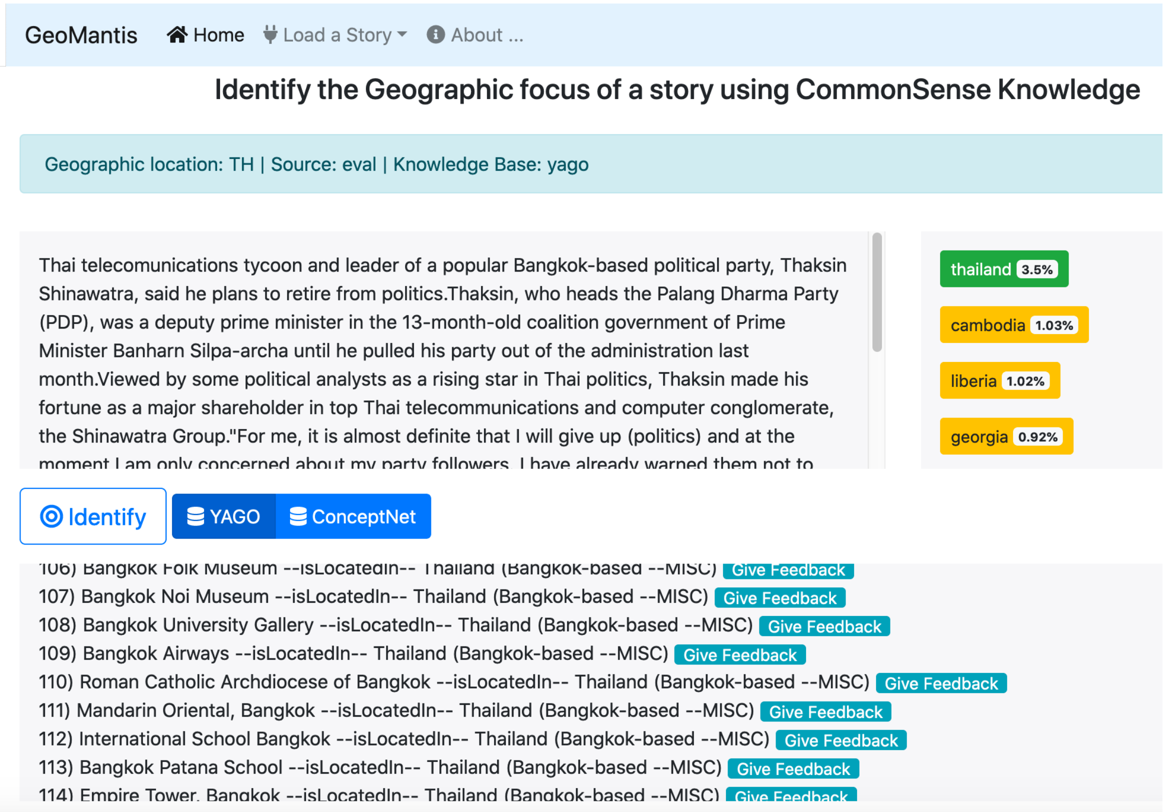
GeoMantis exposes a number of its services using a REST API, based on JavaScript Object Notation (JSON)[3] for data interchange and integration with other systems. Knowledge can be updated at any time by querying the corresponding ontology SPARQL endpoint.

Furthermore, the system has a separate module for producing statistics on documents, datasets, RDF triples, and crowd analytics. It uses a powerful graph library based on Chart.js[4] for presenting a number of visualizations (cf. Figure 6.5). For each processed document, a detailed log of activated triples is kept for debugging purposes and better understanding of the query answering process.

The system's Query Answering Engines rely on MariaDB's full-text search capabilities for the textual processing engine (QAE1) and two argumentation systems for computing arguments acceptance semantics. These systems are ASPARTIX (Egly et al., 2008) and ConArg (Bistarelli and Santini, 2011). ASPATRIX is a well known tool for computing acceptable semantics for Dung's AAF using Answer Set Programming. ConArg is another system, newer than ASPATRIX, that is based on Constraint Programming to model and solve various problems related to the Argumentation research field. This tool is able to compute both Dung's AAF semantics and Weighted argumentation semantics.

[3] http://www.json.org/

[4] http://www.chartjs.org/



Identify the Geographic focus of a story using CommonSense Knowledge

Geographic location: TH | Source: eval | Knowledge Base: yago

Thai telecomunications tycoon and leader of a popular Bangkok-based political party, Thaksin Shinawatra, said he plans to retire from politics.Thaksin, who heads the Palang Dharma Party (PDP), was a deputy prime minister in the 13-month-old coalition government of Prime Minister Banharn Silpa-archa until he pulled his party out of the administration last month.Viewed by some political analysts as a rising star in Thai politics, Thaksin made his fortune as a major shareholder in top Thai telecommunications and computer conglomerate, the Shinawatra Group."For me, it is almost definite that I will give up (politics) and at the moment I am only concerned about my party followers. I have already warned them not to

106) Bangkok Folk Museum --isLocatedIn-- Thailand (Bangkok-based --MISC)
107) Bangkok Noi Museum --isLocatedIn-- Thailand (Bangkok-based --MISC)
108) Bangkok University Gallery --isLocatedIn-- Thailand (Bangkok-based --MISC)
109) Bangkok Airways --isLocatedIn-- Thailand (Bangkok-based --MISC)
110) Roman Catholic Archdiocese of Bangkok --isLocatedIn-- Thailand (Bangkok-based --MISC)
111) Mandarin Oriental, Bangkok --isLocatedIn-- Thailand (Bangkok-based --MISC)
112) International School Bangkok --isLocatedIn-- Thailand (Bangkok-based --MISC)
113) Bangkok Patana School --isLocatedIn-- Thailand (Bangkok-based --MISC)
114) Empire Tower, Bangkok --isLocatedIn-- Thailand (Bangkok-based --MISC)

Figure 6.3 Screenshot of the demo interface where users can test the system using the TF-IDF strategy and QAE1. A list of arguments that support this decision is presented at the bottom where the user can also provide feedback whether that argument is appropriate or not for supporting the identified country.

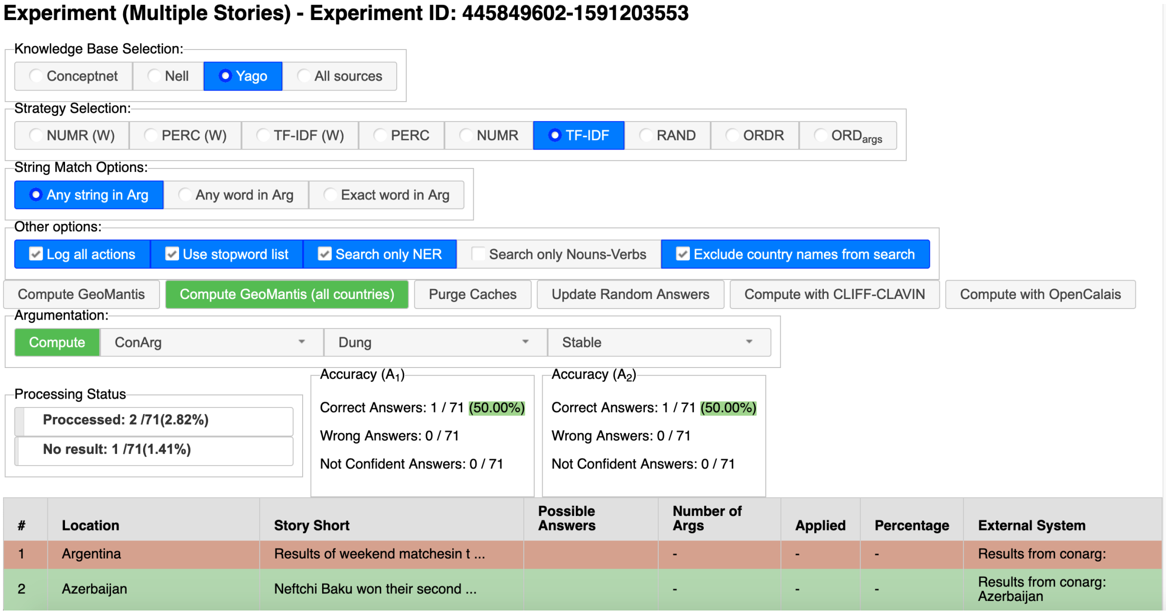
6.3 Empirical Material

The GeoMantis system evaluation, required three inputs: 1) a list of countries, 2) generic knowledge from ontologies about each of these countries, and 3) datasets where the geographic focus of the text is known.

For the first input, we chose countries which are members of the United Nations (UN). The UN is the world's largest intergovernmental organization and has 193 member states. For the other two inputs we provide information in the following sections.

6.3.1 Use of Generic Ontologies

A large amount of general-purpose knowledge is stored in databases in the form of ontologies. This knowledge is gathered from various sources using human workers, game players, volunteers, and contributors in general. We chose two popular ontologies: ConceptNet (Speer



Experiment (Multiple Stories) - Experiment ID: 445849602-1591203553

Knowledge Base Selection:
○ Conceptnet ○ Nell ● Yago ○ All sources

Strategy Selection:
○ NUMR (W) ○ PERC (W) ○ TF-IDF (W) ○ PERC ○ NUMR ● TF-IDF ○ RAND ○ ORDR ○ ORD$_{args}$

String Match Options:
● Any string in Arg ○ Any word in Arg ○ Exact word in Arg

Other options:
☑ Log all actions ☑ Use stopword list ☑ Search only NER □ Search only Nouns-Verbs ☑ Exclude country names from search

Argumentation:
Compute ConArg Dung Stable

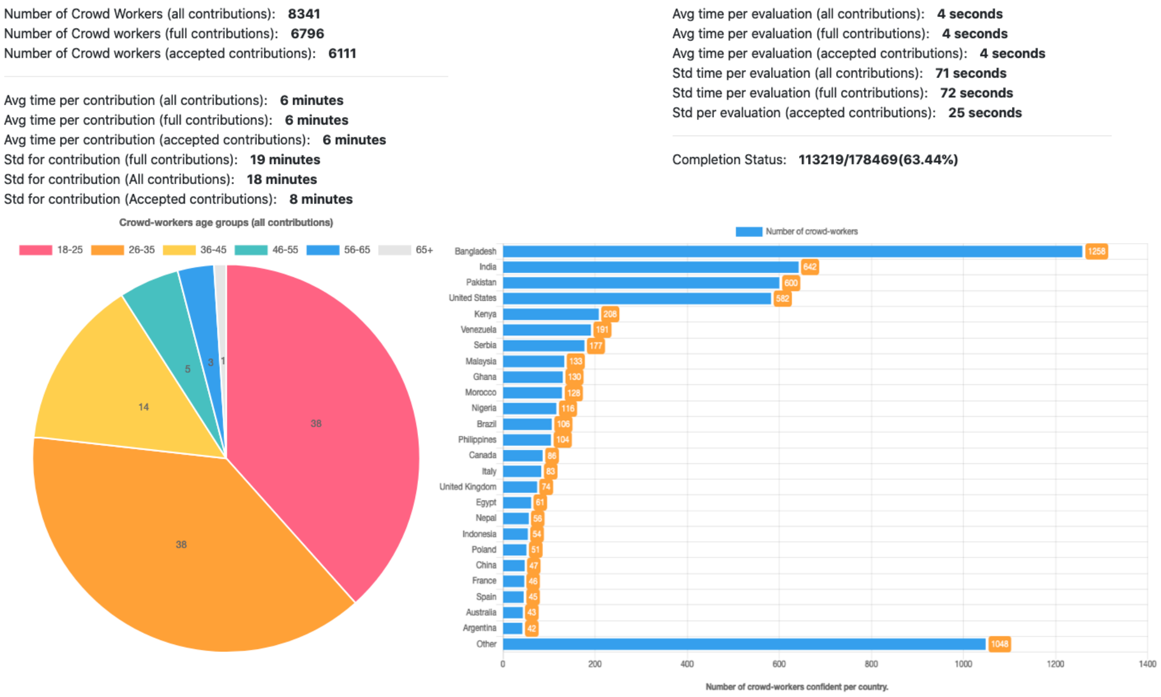
Processing Status
Proccessed: 2 /71(2.82%)
No result: 1 /71(1.41%)

Accuracy (A_1)
Correct Answers: 1 / 71 (50.00%)
Wrong Answers: 0 / 71
Not Confident Answers: 0 / 71

Accuracy (A_2)
Correct Answers: 1 / 71 (50.00%)
Wrong Answers: 0 / 71
Not Confident Answers: 0 / 71

#	Location	Story Short	Possible Answers	Number of Args	Applied	Percentage	External System
1	Argentina	Results of weekend matchesin t ...		-	-	-	Results from conarg:
2	Azerbaijan	Neftchi Baku won their second ...		-	-	-	Results from conarg: Azerbaijan

Figure 6.4 Screenshot of the experiments interface. Users can tweak all system parameters and load stories from specific datasets.



Number of Crowd Workers (all contributions): **8341**
Number of Crowd workers (full contributions): **6796**
Number of Crowd workers (accepted contributions): **6111**

Avg time per contribution (all contributions): **6 minutes**
Avg time per contribution (full contributions): **6 minutes**
Avg time per contribution (accepted contributions): **6 minutes**
Std for contribution (full contributions): **19 minutes**
Std for contribution (All contributions): **18 minutes**
Std for contribution (Accepted contributions): **8 minutes**

Avg time per evaluation (all contributions): **4 seconds**
Avg time per evaluation (full contributions): **4 seconds**
Avg time per evaluation (accepted contributions): **4 seconds**
Std time per evaluation (all contributions): **71 seconds**
Std time per evaluation (full contributions): **72 seconds**
Std per evaluation (accepted contributions): **25 seconds**

Completion Status: **113219/178469(63.44%)**

Figure 6.5 Screenshot of the crowd-analyzer interface. Various statistics are presented related to the argument evaluation process, along with corresponding visualizations of the gathered data.

Table 6.1 Information on triples retrieved from ConceptNet and YAGO ontologies for UN countries. The filtered YAGO ontology (YAGO_Fil) is also depicted in this table and is described in Section 6.4.1.

Property	**ConceptNet**	**YAGO**	**YAGO_Fil**
Total Number of triples	51,771	2,966,765	2,903,186
Number of unique relations	33	373	300
Country with highest number of triples	China	USA	USA
Number of UN countries with triples	193	192	192

and Havasi, 2013) and YAGO (Hoffart et al., 2011; Suchanek et al., 2007, 2008) which include generic knowledge for countries instead of only geographic knowledge that exists in a gazetteer. An overview of these ontologies is presented in Chapter 2 and in the following paragraphs.

ConceptNet is a freely-available semantic network that contains data from a number of sources such as crowdsourcing projects, Games With A Purpose (GWAPs) (von Ahn and Dabbish, 2008), online dictionaries, and manually coded rules. In ConceptNet, data are stored in the form of edges or assertions. An edge is the basic unit of knowledge in ConceptNet and contains a relation between two nodes (or terms). Nodes represent words or short natural language phrases. ConceptNet version 5.6 includes 37 relations, such as “AtLocation”, “isA”, “PartOf”, “Causes” etc. The following are examples of edges available in ConceptNet: <cat> <RelatedTo> <meow>, <statue> <AtLocation> <museum>. ConceptNet is not originally represented in an RDF format, but there is relevant work that suggests such a conversion (Najmi et al., 2016).

For each UN country, its name along with its alternate names are extracted and the ConceptNet 5.6 API[5] is queried for returning the proper Uniform Resource Identifier (URI) in the database. In ConceptNet, each URI includes the language (e.g., “en”) and the term. This is an example of a complete URI: “/c/en/peru”. When the term includes spaces (e.g., “United Kingdom”), these are substituted by underscores, i.e., “c/en/united_kingdom”.

For each obtained URI, all facts are retrieved in the form of triples <Arg1> <Relation> <Arg2> and are stored in the GeoMantis geographic knowledge database. In ConceptNet, the country name can appear either in <Arg1> or <Arg2> and an additional check is needed to capture the appropriate search string. For example, when a search for “Greece” is performed, facts like the ones presented in Figure 6.6 are returned, which after processing

[5] http://api.conceptnet.io/

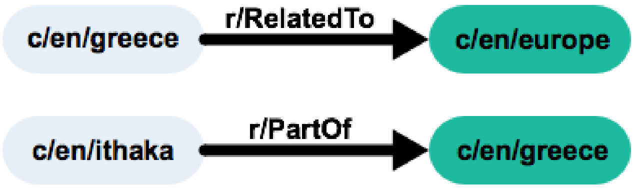


Figure 6.6 Examples of facts retrieved from ConceptNet when the search term "Greece" is used.

(cf. Algorithm 2) result to the search strings: `europe` and `ithaka`. In Figure 6.7, the 20 most frequent relations in the retrieved knowledge are depicted.

YAGO (Yet Another Great Ontology) is a semantic knowledge base built from sources like Wikipedia, WordNet (Fellbaum, 2010) and GeoNames[6]. More specifically, information from Wikipedia is extracted from categories, redirects and infoboxes available in each wikipedia page. Also, there is a number of relations between facts that are described in detail in the work of Hoffart et al. (Hoffart et al., 2011). Currently, YAGO contains 447 million facts and about 9,800,000 entities. Facts in YAGO were evaluated by humans, reporting an accuracy of 95%.

Relations in YAGO are both semantic (e.g., "`wasBornOnDate`", "`locatedIn`" and "`hasPopulation`") and more technically oriented ones (e.g., "`hasWikipediaAnchorText`", "`hasCitationTitle`"). A search for "Greece" in YAGO returns facts like the ones presented in Figure 6.9.

Moreover, YAGO has a number of spatial relations that place an object in a specific location (i.e., country, city, administrative region, etc.). For example, relations "`wasBornIn`", "`diedIn`", "`worksAt`" place an entity of type `Person` in a location, e.g., `<Isaac_Asimov>` `<wasBornIn>` `<Petrovichi>`. In Figure 6.8 the 20 most frequent relations in triples retrieved from YAGO ontology about UN countries are depicted.

For retrieving facts, the YAGO SPARQL endpoint[7] was queried for each UN country name along with its alternate names.

6.3.2 Corpora and Datasets

The last of the inputs needed for the evaluation process are the pre-tagged text corpora. These are collections of texts whose geographic focus is known and available for machine reading.

[6] http://www.geonames.org

[7] https://linkeddata1.calcul.u-psud.fr/sparql

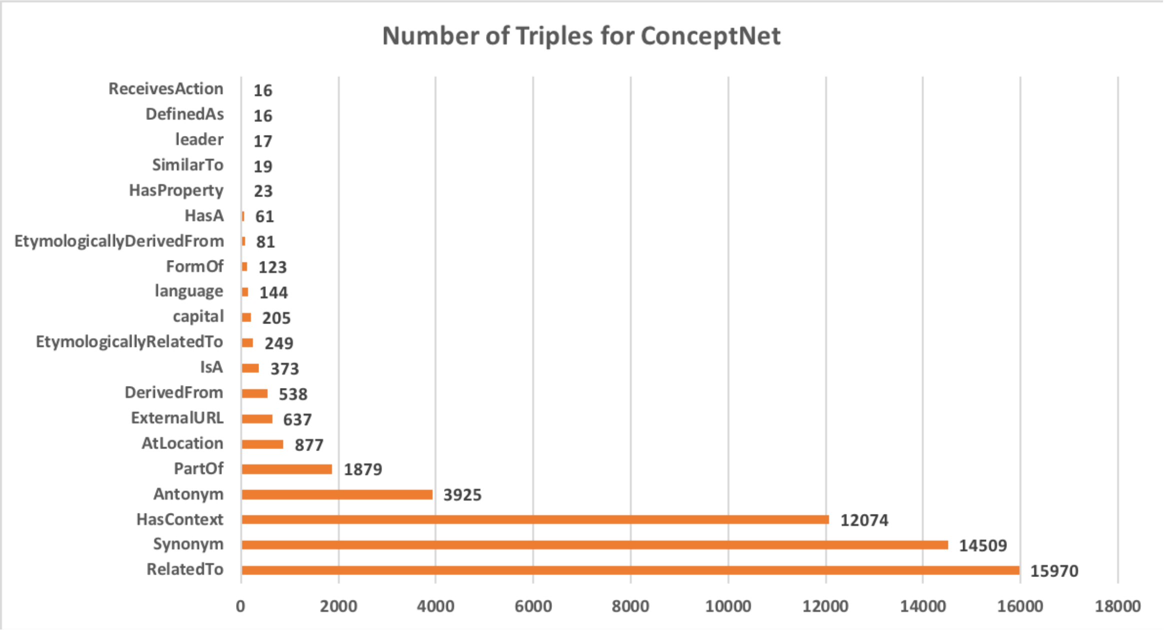


Figure 6.7 The 20 most frequent relations in triples retrieved from ConceptNet ontology about UN countries.

To evaluate the GeoMantis system in a challenging setting, we processed a number of documents from popular corpora by removing references to the country of focus for that document and its alternate names, i.e., a document with geographic focus in “Greece” will not have the word “Greece” or “Hellas” or “Hellenic Republic” in its text after the processing.

There are two commonly used corpora for conducting experiments in this line of research; the Reuters Corpus Volume 1 (RCV) and the New York Times Annotated Corpus (NYT), both of which are presented in Chapter 2. The available content is tagged with location metadata at country-level. Moreover, they contain a plethora of documents for experimentation from different news topics and about various countries.

From the above two corpora we created six datasets to use in the evaluation of the GeoMantis system. These datasets had either the target country and its alternate names obscured, i.e., substituted with the word “unknown” or not present at all. To the best of our knowledge, there is no corpus that guarantees that there is no mention of the target country inside the document. For that reason, we used corpora that are frequently used in this line of research and we constructed datasets either by obscuring or by selecting texts that do not have a mention of the target country to evaluate GeoMantis. The alternate names of the countries were retrieved from the GeoNames database and were limited to English alternate names only.

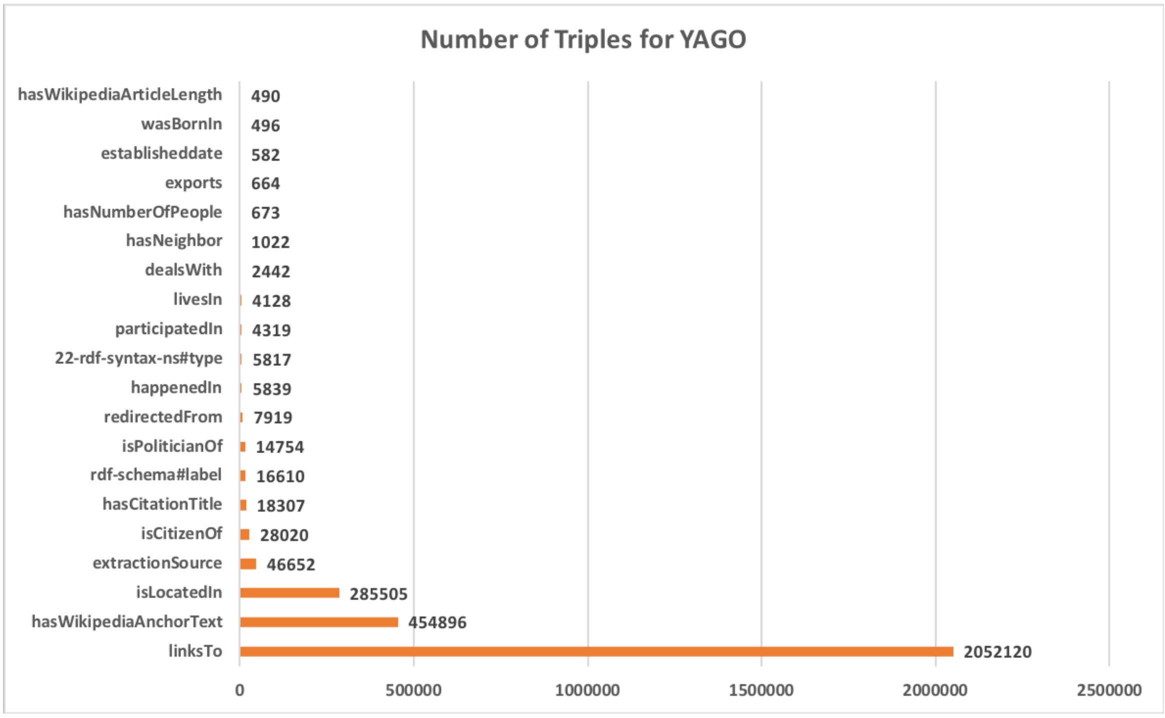


Figure 6.8 The 20 most frequent relations in triples retrieved from YAGO ontology about UN countries.

From the RCV corpus, two datasets were created using 1,000 documents, uniformly randomly selected, without replacement, from the set of news stories in the dataset: the RCV_obs, where the target country and its alternate names are obscured and the RCV_npr, where the target country and its alternate names are not present in the document's text.

From the NYT corpus, two datasets were created using 1,000 news stories, uniformly randomly selected, without replacement, from the set of news stories in the dataset that belong to the "Top/News/World/ Countries and Territories/" category with a single country tag: the NYT_obs, where the target country and its alternate names are obscured, and the NYT_npr, where the target country and its alternate names are not present in the document's text.

The majority of stories in the NYT corpus are geographically focused on the United States of America and Russia, and the majority of stories in the RCV1 corpus are geographically focused on the United States of America and the United Kingdom. For each of the four datasets, we tried to have a balanced distribution of news stories per target country of focus, hence five news stories were uniformly randomly selected, without replacement (if they were available), for each UN member country from the respective corpus. The remaining

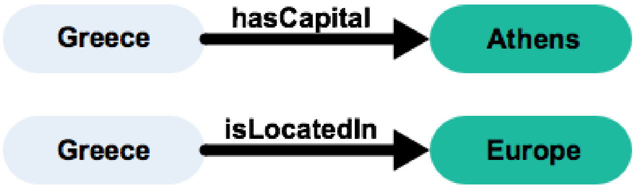


Figure 6.9 Examples of facts retrieved from YAGO when the search term "Greece" is used.

documents were uniformly randomly selected, without replacement, from the whole pool of documents of that corpus.

We also created two new datasets for the comparison of GeoMantis with other systems and two baseline metrics, the `EVA_obs` and the `EVA_npr`.

The `EVA_obs` dataset included 500 uniformly randomly selected without replacement news stories from the RCV corpus and 500 uniformly randomly selected without replacement news stories from the NYT corpus categorized under the "Top/News/World/ Countries and Territories/" category with a single country tag, in a similar way as with the rest of the datasets. Every occurrence of the target country was substituted with the word "unknown". For the `EVA_npr` dataset the same procedure was followed, but each story in the dataset did not have any occurrence of the target country or its alternate names. In Table 6.2 an analysis of the six datasets is presented including number of words in dataset, mean number of words per document, percentage of named entities, etc.

For uniformity, from each of the two corpora, two documents were uniformly randomly selected without replacement (if they were available) for each UN member country. The remaining documents were uniformly randomly selected without replacement from the whole pool of documents. As before, this process allowed a balanced distribution of stories per country in the dataset.

6.4 Evaluation and Analysis

The GeoMantis system is evaluated on whether it can identify the geographic focus of a text document, when the country name in that text is obscured or does not exist, using only knowledge from generic ontologies. The process followed, the metrics, and the results of the evaluation are presented in this section.

A two phase evaluation was conducted: the 1st phase measured the system's performance for each of the parameters (parameter selection) in identifying the geographic focus of a

Table 6.2 Characteristics of the six datasets, including number of documents, number of tagged countries, total and mean number of words and the percentage of the NER labels. Details on the identified named entities are presented as the percentage of words tagged with NER labels in each dataset along with the five labels used in our experiments which are presented as the fraction of the words tagged with each label over the total number of NER labels, converted to a percentage .

Dataset	RCV_obs	RCV_npr	NYT_obs	NYT_npr	EVA_obs	EVA_npr
Number of documents in dataset	1,000	1,000	1,000	1,000	1,000	1,000
Number of countries in dataset	180	125	171	117	186	138
Number of words in dataset	174,347	166,373	393,531	362,228	283,896	216,014
Mean number of words per document	174	166	394	362	284	216
Percentage of Named Entities	23.19%	31.76%	29.36%	24.37%	25.51%	27.86%
[location]	10.97%	9.83%	15.14%	14.68%	14.25%	12.66%
[organization]	21.78%	19.40%	15.08%	17.44%	17.16%	17.49%
[money]	2.63%	2.62%	1.49%	1.83%	1.69%	1.86%
[person]	20.25%	18.88%	23.59%	24.36%	22.31%	22.63%
[misc]	6.39%	6.36%	10.88%	9.93%	9.28%	8.69%

document at a country-level, and the 2nd phase compared the GeoMantis system using the prevailing strategy from the 1st phase, with two open source freely available systems and two common baseline metrics (comparative evaluation). For these experiments, general-purpose knowledge was retrieved for countries that are members of the United Nations (UN)[8] as described in Section 6.3.1.

[8] http://www.un.org

6.4.1 Parameter Selection

The 1st phase of the evaluation was conducted using the four datasets described in Section 6.3.2. We evaluated every combination of values for the ontology, and the PERC and TF-IDF query answering strategies.

A similar evaluation was conducted and described in detail in our previous work (Rodosthenous and Michael, 2018b). That evaluation included three datasets (two from the same sources as with this evaluation and one manually created from the WikiTravel[9] website) and knowledge from Conceptnet and YAGO. The results of that evaluation suggested that the best performing parameters were the YAGO ontology, the application of NER filtering, and the PERC query answering strategy, even though the TF-IDF strategy was also performing very well. Those datasets were processed by just obscuring the reference country name from the document, as opposed to the extensive filtering of both the name and alternate names we performed in this evaluation.

Parameters like NER filtering, were tested thoroughly in the previous evaluation of GeoMantis and found to increase the performance of the system when used, hence it was always enabled in this evaluation. NER filtering includes the use of words that were labeled as location, person, organization, and money by the NER process. Although not reported here, the application of the NER filter also significantly reduces the processing time. Furthermore, the *Number of arguments activated (NUMR)* and *Most arguments per country ordering (ORDC)* query answering strategies, were found not to perform well and were not tested in this evaluation.

For the evaluation process, the datasets were imported to the GeoMantis database and processed with the Stanford CoreNLP. Then, the system's knowledge retrieval engine was directed to ConceptNet and YAGO ontologies to retrieve RDF triples and construct arguments. These arguments were processed using the NLP system. Table 6.1 depicts the properties for the ontologies used.

The performance of each combination of parameters, was evaluated using the mean position metric and the accuracy. The mean position ($\bar{P}$) denotes the position of the target country in the ordered list of countries over the number of countries available in the dataset. For comparison purposes, this number is converted to a percentage.

The accuracy(A_i) of the system is defined as $A_i = \frac{N_i}{C}$, where $i \in \{1,2,3,...,M\}$ and M is the number of countries in the dataset, N_i denotes the number of correct assignments of the target country when the target country's position is $\leq i$ in the ordered list of countries and C denotes the number of available documents in the dataset.

[9] https://wikitravel.org

Table 6.3 Results from the parameter selection phase of the GeoMantis system evaluation. The query answering strategies and ontologies, when the NER filtering option is used, were evaluated. Rows highlighted in light blue, identify the best performing set of parameters in terms of minimum value for $\bar{P}$ and maximum value for A_1 and A_2.

#	Dataset	Ontology	Strategy	A_1	A_2	$\bar{P}$
YP1	RCV_obs	YAGO	PERCR	23.70	39.80	8
YT1	RCV_obs	YAGO	TF-IDF	41.10	61.60	6
CP1	RCV_obs	ConceptNet	PERCR	18.70	27.7	16
CT1	RCV_obs	ConceptNet	TF-IDF	19.80	29.30	16
YP2	RCV_npr	YAGO	PERC	36.30	48.80	8
YT2	RCV_npr	YAGO	TF-IDF	45.40	58.60	8
CP2	RCV_npr	ConceptNet	PERCR	29.40	42.80	12
CT2	RCV_npr	ConceptNet	TF-IDF	27.50	37.90	13
YP3	NYT_obs	YAGO	PERCR	18.60	31.20	11
YT3	NYT_obs	YAGO	TF-IDF	34.00	52.40	7
CP3	NYT_obs	ConceptNet	PERCR	11.60	22.20	14
CT3	NYT_obs	ConceptNet	TF-IDF	15.10	27.00	13
YP4	NYT_npr	YAGO	PERCR	36.40	50.70	10
YT4	NYT_npr	YAGO	TF-IDF	49.80	65.50	7
CP4	NYT_npr	ConceptNet	PERCR	26.50	44.00	11
CT4	NYT_npr	ConceptNet	TF-IDF	28.80	43.70	11

The parameter selection process was applied on the RCV_obs, RCV_npr, NYT_obs and NYT_npr datasets.

In Table 6.3, we present the results of the parameter selection process after the chosen ontology and the query answering strategy followed (cf. Section 6.2.3) are tested. These results are also depicted graphically in Figure 6.10.

Comparing the results in terms of ontology used, knowledge from YAGO yields better results than that of ConceptNet. Further analysis of the two ontologies, shows a huge gap in the amount of facts retrieved for each country. In particular, YAGO includes 2,966,765 triples against 51,771 triples in ConceptNet.

The results indicate that the common prevailing strategy for all four datasets is **TF-IDF** when the **YAGO** knowledge base is used. These results are inline with the results from our previous experiments, since the TF-IDF strategy performed almost equally well with the PERC startegy in that evaluation. Furthermore, we speculate that the increase in the amount of arguments from the YAGO ontology required a more refined method of selecting the activated argument than the simple PERC strategy.

Table 6.4 Results from fine-tuning the parameter selection phase of the GeoMantis system evaluation. We examined the performance when using the “misc” NER tag instead of “money” and the use of the filtered YAGO ontology (`YAGO_Fil`).

#	Dataset	Ontology	Strategy	A_1	A_2	$\bar{P}$
YFT1	RCV_obs	YAGO_Fil	TF-IDF	42.80	61.60	5
YFT2	RCV_npr	YAGO_Fil	TF-IDF	49.60	62.20	6
YFT3	NYT_obs	YAGO_Fil	TF-IDF	36.60	55.20	5
YFT4	NYT_npr	YAGO_Fil	TF-IDF	52.90	67.90	5

The results propose that further tuning of the selected parameters could increase the accuracy and minimize the mean position. Instead of using the “money” NER tag, we chose the “misc” tag that actually contains named entities that do not exist in any other tags. The “money” tag included words like “billion”, “4,678,909” that do not offer much in the query answering process.

Furthermore, we created a filtered version of the YAGO ontology (`YAGO_Fil`), by removing triples with relations that identify and contain technical information (e.g., “`owl#sameAs`”, “`extractionSource`”, “`hasWikipediaArticleLength`”) and relations like “`imageflag`” and “`populationestimaterank`”, that do not include useful information.

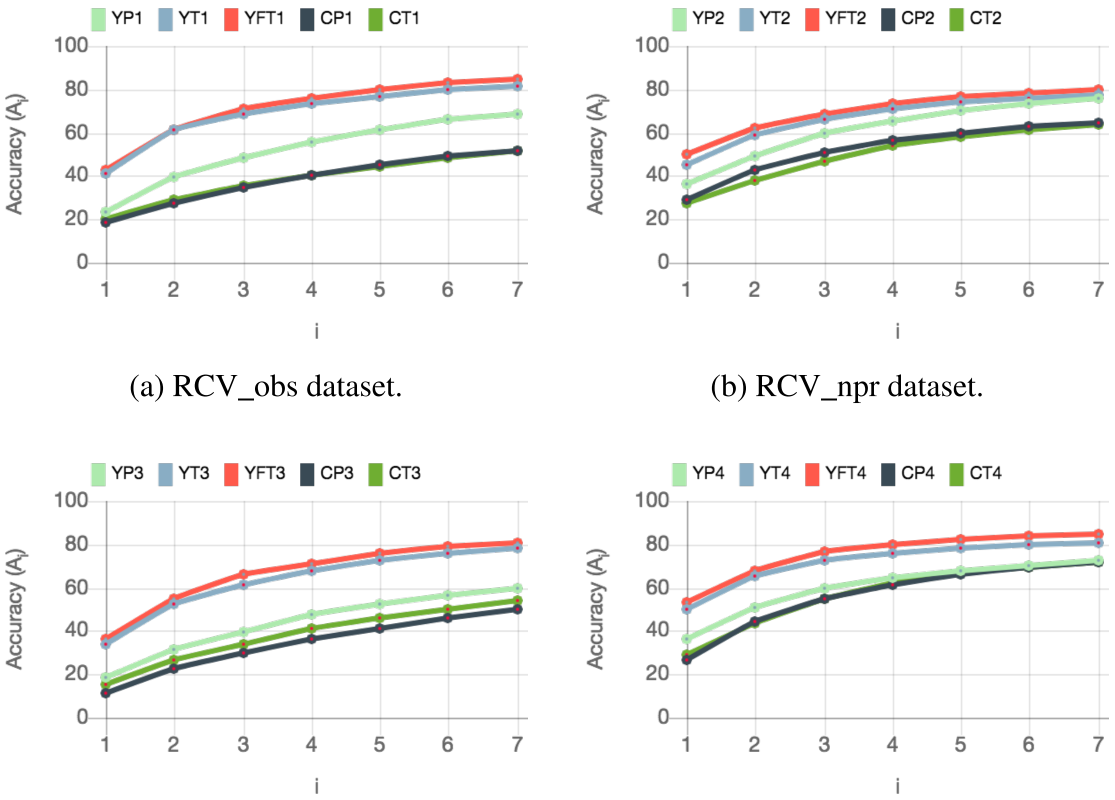
Results presented in Table 6.4, suggest that the usage of the **YAGO_Fil** ontology with the “**misc**” tag, minimize $\bar{P}$ and maximize the accuracy of both A_1 and A_2 for all four datasets. In fact, the $\bar{P}$ is decreased by two positions in three out of four datasets and A_1 and A_2 were increased for all datasets.

6.4.2 Comparative Evaluation

In the 2nd phase of the evaluation, the GeoMantis system, using the prevailing strategy identified in the 1st phase of the evaluation, was compared with two freely available open source systems, CLIFF-CLAVIN and Mordecai, and two common baseline metrics. These metrics included the random selection of countries (RAND) and the ordering of countries based on their frequency of appearance in the dataset (ORDC) for ordering the list of countries.

Two additional independent datasets were used comprising previously unseen documents from the same sources used for the 1st phase.

For the comparative evaluation, we used the accuracy metric and the unanswered metric. The unanswered metric U denotes the percentage of the number of documents processed without the system returning a result.



(a) RCV_obs dataset.

(b) RCV_npr dataset.

(c) NYT_obs dataset.

(d) NYT_npr dataset.

Figure 6.10 Graphical representation of the results when the four datasets are used. On the x-axis, i gets values from 1 to 7 and the values on the y-axis present A_i, that is the percent of the correct assignments of the target country in the first i responses of the system.

To conduct the comparative evaluation, the CLIFF-CLAVIN geolocation service was set up and a script was used to read the JSON output of the system. More specifically, the "places/focus/countries" array of the JSON results was used.

Results returned from the CLIFF-CLAVIN system are not ordered, so for comparison reasons with the GeoMantis system, the A_1 and A_7 metrics are used, where A_1 is the accuracy of the system when only one result is returned and it is the correct target country assignment and A_7 is the accuracy of the system when up to 7 results are returned and the correct target country assignment is in this set. The reason 7 was chosen is that it corresponds to the maximum number of predicted countries CLIFF-CLAVIN returns when executed on both the `EVAL_obs` and the `EVAL_npr` datasets and the target country is identified by any one of them. This weakness of the CLIFF-CLAVIN system is also stressed by other researchers (Imani et al., 2017) who used this system for comparison purposes.

For Mordecai, a webservice was not available, hence we set up the system locally, following the instructions[10] given by its developer. More specifically, this system requires Python version 3, spaCy NLP model and the GeoNames database. In order to work, Mordecai needs access to a Geonames gazetteer running in Elasticsearch[11]. We created a python script that can take a folder of documents and parse them using the Mordecai API using the `geo.infer_country` function.

The results are stored in a new file and are filtered so that only the returned tag "`country_predicted`" is stored in the output file. Mordecai returns the predicted country for each place name in ISO3 country code format (e.g., GRC, BGR). To be able to compare this system, we created a script that converts ISO3 to ISO country code format and suggests a geographic focus for the document according to a frequency-based approach, i.e., the returned countries are ordered according to their frequency of appearance. The comparative evaluation was applied on the EVA_obs and EVA_npr datasets.

In Table 6.5, rows highlighted in light green identify the best results in terms of A_1 and A_7 for each of the two datasets. In Figure 6.11 these results are presented graphically, illustrating all comparative evaluation experiments.

Results from the 2nd phase evaluation for the GeoMantis system are comparable to that of CLIFF-CLAVIN, Mordecai and that of the two baseline metrics. In cases where the target country is obscured or not present in the dataset, the GeoMantis system outperforms both CLIFF-CLAVIN and Mordecai, as well as the two baseline metrics.

The `EVA_npr` dataset presents better results in terms of accuracy, since the information present in this dataset is unaffected by the obscuring process. The way stories are written probably includes other type of information to identify the country without an explicit mention of it in the text. On the other hand, stories in the `EVA_obs` dataset have an explicit mention of the target country in the document text that was obscured. This led to fewer references left in the story text and hence, made it more difficult to identify the target country.

Furthermore, the comparison of C1 with M1 and C2 with M2 shows that CLIFF-CLAVIN performs marginally better than Mordecai, when the target country is obscured or not present in the document. This was also tested in the work of Imani et al. (2017), on sentences without the target country obscured and the results show that the CLIFF-CLAVIN system outperformed Mordecai in terms of accuracy.

In terms of the U metric, CLIFF-CLAVIN and Mordecai have a relatively high percentage of unanswered documents. More specifically, CLIFF-CLAVIN was not able to identify the

[10] https://github.com/openeventdata/mordecai

[11] https://www.elastic.co/

Table 6.5 Comparison of the GeoMantis system with CLIFF-CLAVIN, Mordecai and the Baseline. Rows highlighted in light green identify the results that are comparable.

#	Dataset	System	Parameters	$A_1(\%)$	$A_2(\%)$	$A_7(\%)$	$U(\%)$
G1	**EVA_obs**	**Geomantis**	**YAGO_Fil, TF-IDF**	**46.60**	**64.60**	**87.02**	0
C1	**EVA_obs**	**CLIFF-CLAVIN**	**default**	**42.50**	**-**	**50.00**	**10.70**
M1	**EVA_obs**	**Mordecai**	**default**	**41.10**	**51.50**	**64.00**	**7.20**
B1	**EVA_obs**	**Baseline**	**RAND**	**0.50**	**1.10**	**3.90**	0
B2	**EVA_obs**	**Baseline**	**ORDC**	**2.00**	**3.80**	**11.00**	0
G2	**EVA_npr**	**Geomantis**	**YAGO_Fil, TF-IDF**	**55.40**	**68.20**	**86.10**	0
C2	**EVA_npr**	**CLIFF-CLAVIN**	**default**	**52.70**	**-**	**59.50**	**17.90**
M2	**EVA_npr**	**Mordecai**	**default**	**52.10**	**62.20**	**66.90**	**14.80**
B3	**EVA_npr**	**Baseline**	**RAND**	**0.80**	**1.30**	**5.10**	0
B4	**EVA_npr**	**Baseline**	**ORDC**	**3.30**	**5.50**	**15.70**	0

geographic focus of 179 documents in the `EVA_npr` dataset and 107 documents in the `EVA_obs`.

6.5 A Crowdsourcing Approach

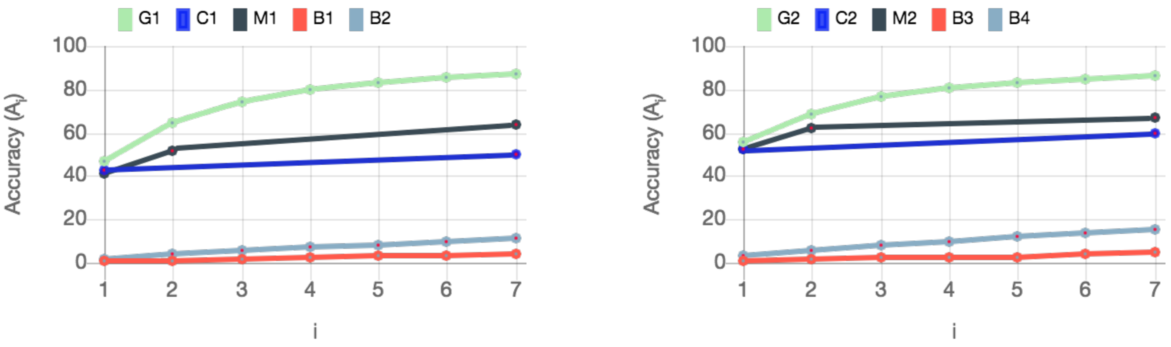
In this section we present a strategy for expanding the GeoMantis architecture by adding weights on arguments and applying it on the three query answering strategies presented in Section 6.2.3. Weights are added to arguments using a crowdsourcing evaluation methodology. This methodology uses monetary incentives and platforms such as “Amazon Mechanical Turk”[12], “Figure Eight”[13] (former crowdflower) and microWorkers[14]. **The hypothesis we test is that knowledge evaluated by the crowd can yield better results in terms of accuracy compared to the ones presented in the previous sections**.

To test the hypothesis we first need to prepare the following workflow: 1) create a dataset of stories, 2) identify activated arguments, 3) evaluate arguments using crowd-workers, 4) apply a GeoMantis strategy using the evaluated arguments and 5) test them on the dataset.

[12] https://www.mturk.com/

[13] https://www.figure-eight.com/

[14] https://www.microworkers.com



(a) EVA_obs dataset.

(b) EVA_npr dataset.

Figure 6.11 Graphical representation of the comparative evaluation results when the `EVA_obs` and `EVA_npr` datasets are used. On the x-axis, i gets values from 1 to 7 and the values on the y-axis present A_i, that is the percent of the correct assignments of the target country in the first i responses of the system.

6.5.1 Weighted Query Answering Strategies

Firstly, we extend the original query answering strategies presented in Section 6.2.3 to include the evaluations received from the crowd-workers. The ordering of the list of countries and the generation of the predicted geographic focus is performed using one of the following strategies:

Weighted Percentage of arguments applied ($PERCR_w$): List of countries is ordered according to the fraction of each country's total weight of activated arguments over the total weight of arguments for that country that exist in the geographic knowledge bases, in descending order.

Weighted Number of arguments applied ($NUMR_w$): List of countries is ordered according to each country's total weight of activated arguments, in descending order.

Weighted Term Frequency - Inverse Document Frequency (TF-IDF_w): List of countries is ordered according to the TF-IDF algorithm, which is applied as follows:
D_c is a document created by taking the arguments of a country c
TF_t = (Sum of weights of arguments in D_c where term t appears) / (Sum of weights of arguments included in D_c)
IDF_t = $\log_e$(Sum of weights of D_c / Sum of weights of D_c with term t in it).

6.5.2 Argument Evaluation System

To evaluate arguements, we designed a system that is able to handle crowd-workers, present arguments for evaluation, check the workers' confidence in evaluating an argument and handle workers' payments. The system is built on top of the GeoMantis system using the same technology stack (PHP, mariaDB, javascript). The system is available at: https://geomantis.ouc.ac.cy/eval.php and is fully integrated to work with the microWorkers platform.

Workers are presented with detailed instructions on how to evaluate each argument and are requested to complete their microWorkers' ID, country of origin and country they feel confident in evaluating arguments (cf. Figure 6.12). Next, crowd-workers are presented with arguments to evaluate. These arguments are chosen using Algorithm 3. When a worker successfully validates all arguments, a unique code is presented for the worker to copy it to the microWorkers website and receive the payment.

The microWorkers platform (Schmidt and Jettinghoff, 2016) was chosen since it includes a large community of crowd-workers and it is accessible to us, unlike the other two platforms[15], i.e., mTurk and Figure-Eight. Each crowd-worker evaluates how useful each argument is on supporting the geographic focus of a specific country. Crowd-workers can choose between three options: "not useful", "I don't know", "Useful" which correspond to -1, 0, and 1 integer values. Crowd-workers get only to see the arguments activated for each of the countries and not the story that these arguments are activated by (cf. Figure 6.13).

Each crowd-worker needs to have a basic understanding of the English language since the arguments are presented in English. When possible the system presents arguments from the country the crowd-worker originates from or from the country the crowd-worker is confident on evaluating arguments. This way, we make sure that crowd-workers can understand the argument that is presented in English language and that they can also provide useful validation of the argument contents, i.e., it would make better sense to ask a person from Brazil about an argument used to support the geographic focus of Brazil instead of any other country.

6.5.3 Experimental Material

First, the dataset used is selected. Stories from the `EVA_npr` dataset were selected since it includes stories in their original form where the country of focus is not explicitly present in the story text. The chosen stories have the country of focus among the first seven in the order list of identified countries, when a GeoMantis strategy (PERCR, NUMR, TF-IDF) is applied.

[15] The mTurk Platform is currently not available in residents of Cyprus and the new Figure-Eight platform (previously known as CrowdFlower) changed its business model completely and it is oriented towards enterprises nowadays.

Algorithm 3 Algorithm to select arguments for evaluation and present them to crowd-workers.

```
procedure SELECTARGSFOREVAL($KB$)
// Each crowd-worker evaluates N arguments
    $N \leftarrow$ NumArgToEval
// 30% of the arguments are selected from the country that the user is confident in contributing.
    $CountryConf \leftarrow$ (30%)*N
    $args \leftarrow$ GetArgs($KB$,$CountryConf$,$CountryISO$)
// 40% of the arguments are selected from the user's country of origin.
    $CountryOrig \leftarrow$ (40%)*N
    $args \leftarrow$ GetArgs($KB$,$CountryOrig$,$CountryISO$)
// 20% of the arguments are selected from arguments that have at least one evaluation.
    $OneEval \leftarrow$ (20%)*N
    $args \leftarrow$ GetArgs($KB$,$OneEval$)
// 10% of the arguments are selected in a random order.
    $Rand \leftarrow$ (10%)*N
    $args \leftarrow$ GetArgs($KB$,$Rand$)
// In case the number of arguments is not reached, we select the rest in random.
    if ($count(args) < N$) then
        $args \leftarrow$ GetArgs($KB$,$N$-$count(args)$)
    end if
    Return ($args$)
end procedure
```

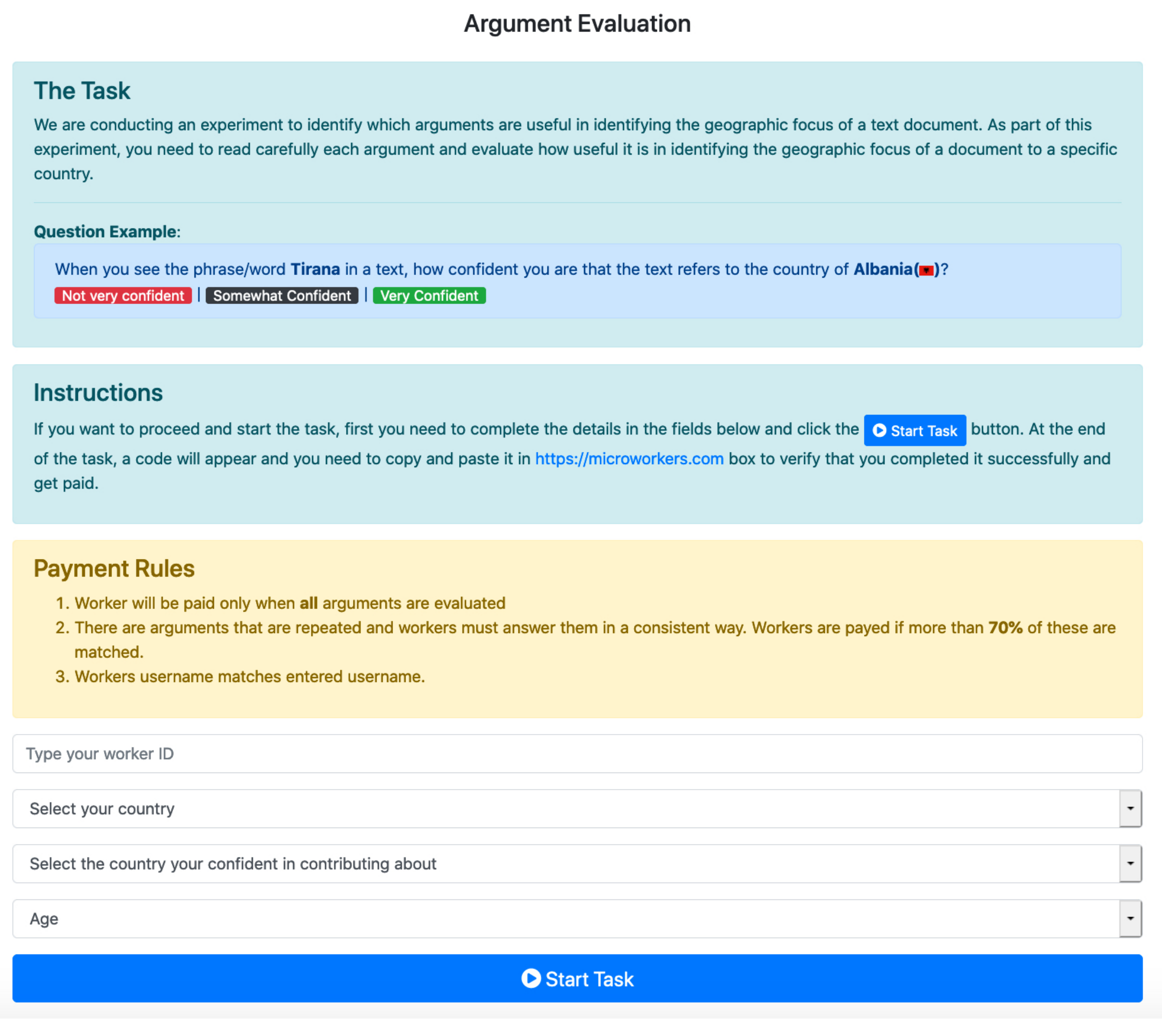
We then identify all commonsense knowledge arguments that are activated. An argument is activated when a word (or its lemma) exists in the story text, when the GeoMantis system is applied using one of the three GeoMantis strategies.

Moreover, four subsets of the datasets are created as follows:

- Dataset `Crowd_npr_1` includes N stories from the `EVA_npr` dataset, where the country of focus is correctly identified in the top position (A_1) and $|A_1 - A_2| > \lambda$, where λ is a threshold.

- Dataset `Crowd_npr_2` includes N stories from the `EVA_npr` dataset, where the country of focus is correctly identified in the top position (A_1) and $|A_1 - A_2| < \lambda$, where λ is a threshold.

- Dataset Crowd_npr_3 includes N stories from the `EVA_npr` dataset, where the country of focus is not correctly identified in the top position (A_1) and $|A_1 - A_2| > \lambda$, where λ is a threshold.



Argument Evaluation

The Task

We are conducting an experiment to identify which arguments are useful in identifying the geographic focus of a text document. As part of this experiment, you need to read carefully each argument and evaluate how useful it is in identifying the geographic focus of a document to a specific country.

Question Example:

When you see the phrase/word **Tirana** in a text, how confident you are that the text refers to the country of **Albania**()?

Not very confident | Somewhat Confident | Very Confident

Instructions

If you want to proceed and start the task, first you need to complete the details in the fields below and click the Start Task button. At the end of the task, a code will appear and you need to copy and paste it in https://microworkers.com box to verify that you completed it successfully and get paid.

Payment Rules

1. Worker will be paid only when **all** arguments are evaluated
2. There are arguments that are repeated and workers must answer them in a consistent way. Workers are payed if more than **70%** of these are matched.
3. Workers username matches entered username.

Type your worker ID

Select your country

Select the country your confident in contributing about

Age

Start Task

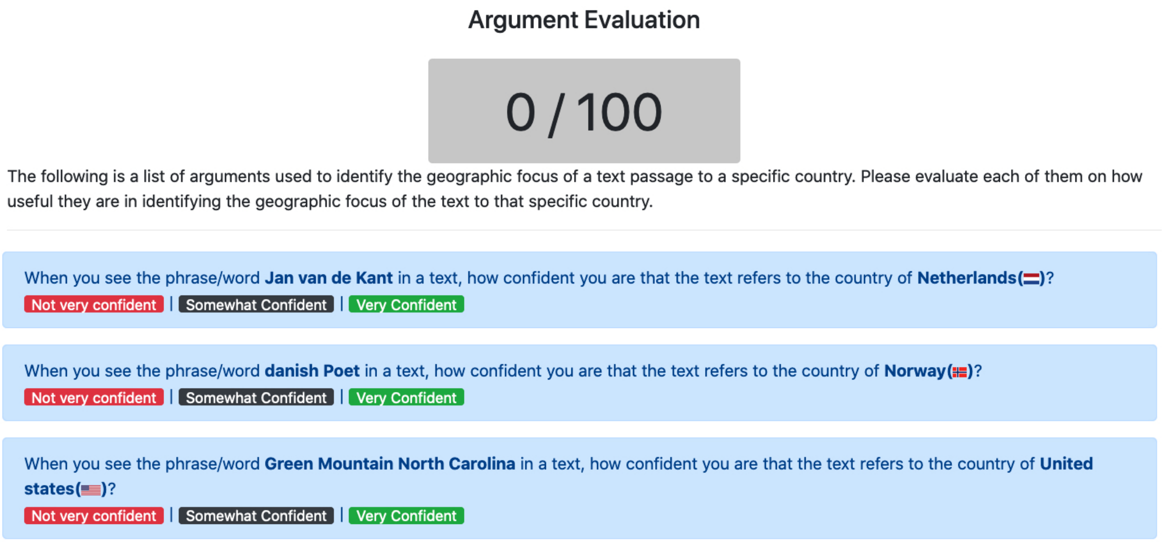
Figure 6.12 A screenshot of the GeoMantis interface for registering crowd-workers.

- Dataset Crowd_npr_4 includes N stories from the `EVA_npr` dataset, where the country of focus is not correctly identified in the top position (A_1) and $|A_1 - A_2| < \lambda$, where λ is a threshold.

The above subsets are used for testing if the argument weighting strategy can change the accuracy in both clear and borderline cases of identifying correctly the geographic focus of a story. For example the `Crowd_npr_1` subset is characterized by the number of confusing stories it includes, since the threshold for the top 7 identified countries is small.

Preliminary Experiment 1

Before proceeding with the experiment we decided to run a short first experiment to verify our workflow and validate the argument evaluation system. As a first test, we process the



Argument Evaluation

0 / 100

The following is a list of arguments used to identify the geographic focus of a text passage to a specific country. Please evaluate each of them on how useful they are in identifying the geographic focus of the text to that specific country.

When you see the phrase/word **Jan van de Kant** in a text, how confident you are that the text refers to the country of **Netherlands(🇳🇱)**?

When you see the phrase/word **danish Poet** in a text, how confident you are that the text refers to the country of **Norway(🇳🇴)**?

When you see the phrase/word **Green Mountain North Carolina** in a text, how confident you are that the text refers to the country of **United states(🇺🇸)**?

Figure 6.13 A screenshot of the GeoMantis interface for evaluating commonsense knowledge arguments for a specific country. On the top of the screen, users are presented with the argument counter showing both the number of arguments remaining and the total number of arguments for this specific crowd-worker. Next, each argument is presented along with the three evaluation buttons next to it. When an evaluation button is clicked, the argument disappears and the crowd-worker moves to the next argument.

`EVA_npr` dataset using the PERCR strategy and proceeded to generate the four `Crowd_X` subsets. For identifying an appropriate λ which will allow a representation of stories from all four subsets we executed a simulation where $\lambda \in (0.1 - 7.0)$ (heuristically identified) and we selected the λ that allows a maximum inclusion of stories from all 4 datasets. Using the results in Table 6.6 we deduce that $\lambda = 1.3$ is a good value and hence we have **210** stories for `Crowd_npr_1`, **211** stories for `Crowd_npr_2`, **74** stories for `Crowd_npr_3` and **83** stories for `Crowd_npr_4`.

Further analysis reveals that 1,203,518 arguments were activated for 138 countries in the `EVAL_npr` dataset when processed using the PERCR strategy. For all four `Crowd_npr` subsets, 1,021,290 arguments were activated from 138 countries. Next, we present the following information for each of the four subsets:

- `Crowd_npr_1`: 210 stories of which 757,045 arguments were activated for 138 countries
- `Crowd_npr_2`: 211 stories of which 506,705 arguments were activated for 137 countries
- `Crowd_npr_3`: 74 stories of which 423,175 arguments were activated for 137 countries

- `Crowd_npr_4`: 83 stories of which 417,303 arguments were activated for 137 countries

Since we need only the A_1 and A_2 metrics, we limited the number of activate arguments to a subset of arguments that identify only the countries for A_1 and A_2. This amounts to 1,319,478, a number which is very difficult to annotate using paid crowd-workers due to the high amount of resources required to perform this action. We limited the number of stories to 30, chosen randomly from each of the 4 datasets. This amounts to 1,100,441 (464,516 unique) activated arguments which is again a large amount of arguments to be verified using paid crowdsourcing. This amount is reduced to 72,724 arguments (49,248 unique) when we selected arguments that are activated for identifying A_1 and A_2 only.

Table 6.6 Results from executing the simulation to select an appropriate value for λ for the `EVAL_npr` dataset when processed using the PERCR strategy. The table depicts results in the range of $\lambda \in (0.1 - 2)$. The first column depicts the value of λ, the next 4 columns depict the number of stories for the respective subsets (`C_npr_X` is short for `Crowd_npr_X`, columns 7 and 8 depict the sum of stories for the respective datasets and the last column depict the difference from the values of column 7 and 8. The highlighted row depicts the chosen λ based on the smallest value of `diff`.

λ	C_npr_1	C_npr_2	C_npr_3	C_npr_4	C_npr_1 + C_npr_3	C_npr_2 + C_npr_4	diff
0.1	394	27	141	16	535	43	492
0.2	378	43	129	28	507	71	436
0.3	356	65	124	33	480	98	382
0.4	339	82	117	40	456	122	334
0.5	319	102	113	44	432	146	286
0.6	301	120	109	48	410	168	242
0.7	283	138	99	58	382	196	186
0.8	265	156	91	66	356	222	134
0.9	252	169	89	68	341	237	104
1	238	183	86	71	324	254	70
1.1	228	193	79	78	307	271	36
1.2	220	201	78	79	298	280	18
1.3	210	211	74	83	284	294	10
1.4	193	228	70	87	263	315	52
1.5	181	240	69	88	250	328	78
1.6	175	246	66	91	241	337	96
1.7	169	252	60	97	229	349	120
1.8	162	259	55	102	217	361	144
1.9	153	268	54	103	207	371	164
2	150	271	49	108	199	379	180

Table 6.7 Information and statistics for the short test experiment, including both information on the experiment and crowd-worker statistics.

Number of Stories	1
Number of Arguments to evaluate	357

Number of Crowd Workers (all contributions)	10
Number of Crowd workers (completed contributions)	2
Avg time per contribution (all contributions)	13 minutes
Avg time per contribution (completed contributions)	55 minutes
Avg time per evaluation (all contributions)	7 seconds
Avg time per evaluation (completed contributions)	10 seconds
Amount payed per worker	$0.50 USD

We executed a short test experiment to check if the proposed workflow is valid and can be applied in evaluating arguments on all stories. A story from the `Crowd_npr_2` subset was selected with all arguments applied to it (total of 357 arguments). An amount of $0.50 USD was paid to each worker who successfully completed the task.

We launched the evaluation system, where we presented 357 arguments to each worker for evaluation. 30% of the arguments were selected from the country that the worker was confident in contributing in, 40% of the arguments were selected from the worker's country of origin, 20% of the arguments were selected from arguments that have at least one evaluation, 10% of the arguments were selected in a random order. In case any of the former three categories had no arguments, we then retrieved arguments using random selection. From the total number of presented arguments, 10% is repeated as test (gold) questions used to evaluate the worker's evaluations. This percentage could vary from 10% to 30% (Bragg et al., 2016). More specifically, each worker is required to provide same answers for 10% of the test questions. Workers who achieve a percentage of less than the defined threshold are not accepted as valid. The threshold could vary from 50% to 70%. Results obtained are depicted in Table 6.7.

In terms of validity of the worker results, we examined the contributions of the two workers that successfully completed the task. The first worker achieved a score of 16 out of 36 (44.44%) and the second worker a score of 33 out of 36 (91.67%) for the validation questions. We also examined the order of validating the presented arguments. Both workers followed the instructions provided. On average, workers completed the test after 55 minutes and needed 10 seconds per evaluation. The fact that only 2 out of 10 workers completed the task and the amount of time needed to complete the task showed us that we needed to reduce the amount of arguments presented to workers.

At this point there was no need to test the performance of our methodology on the dataset as the purpose of this short test experiment was just to verify that the workflow is valid and identify possible problems with the argument evaluation system.

Preliminary Experiment 2

After testing the argument evaluation system, we expanded the preliminary experiment with 10 stories, taking 3 from subset `Crowd_npr_1`, 3 from subset `Crowd_npr_2`, 2 from subset `Crowd_npr_3`, and 2 from subset `Crowd_npr_4`. A total of 5,980 unique arguments were activated for identifying A_1 and A_2. The country of focus for these 10 stories includes Switzerland, Germany, Jordan, Portugal, Somalia, India, Liechtenstein, Saudi Arabia and Georgia.

We imported the arguments to the argument evaluation system of GeoMantis, setting the following requirements for acceptance of a worker's contribution:

- A total of 100 arguments should be evaluated
- At least a score of 50% at the validation test should be achieved

We identified a number of workers that tried to cheat the system, by using two or more browser windows at the same time, by trying to access the system's API directly without contributing, and by submitting random answers. All these where anticipated and measures had been taken before launching the argument evaluation and those evaluations were discarded.

In Table 6.8 we present information on the experiment and the crowd-workers' contributions. Moreover, in Figure 6.14 the statistics on the age groups of crowd-workers are depicted. The majority of crowd workers are in the age group of 26-35, followed by workers in age group 18-25.

The contributed evaluations were used to add weights to all evaluated arguments. More specifically, for each argument we counted the number of positive, negative and neutral feedback. When the sum of negative and neutral feedback was smaller than the sum of positive feedback then we added an integer weight of 600. When equal then we added a weight of 0 and when larger we added a weight of 0. We used PERCR_w and NUMR_w strategies and the results showed an increase (cf. Table 6.9) on the accuracy when compared to the original strategies. The TF-IDF strategy was not tested at that time, since it required all arguments to be evaluated, even the ones that were not activated.

A further analysis of the results reveals that arguments based on the "`linksTo`" and "`isLocatedIn`" relation get the highest score (sum of weights) (cf. Figure 6.15).

Table 6.8 Information and statistics for the 2nd preliminary experiment, including both information on the experiment and crowd-worker statistics.

Number of Stories	10
Number of Arguments to evaluate	5,980
Minimum number of evaluators per argument	3
Test acceptance percentage	≥50%
Time needed for evaluation	4 (3.8) days
Number of Crowd Workers (all contributions)	502
Number of Crowd workers (completed contributions)	280
Number of Crowd workers (accepted contributions)	217
Avg time per contribution (all contributions)	21 minutes
Avg time per contribution (completed contributions)	36 minutes
Avg time per contribution (accepted contributions)	7 minutes
Avg time per evaluation (all contributions)	27 seconds
Avg time per evaluation (completed contributions)	21 seconds
Avg time per evaluation (accepted contributions)	5 seconds
Amount payed per worker	$0.10 USD

Weighting Strategy

The argument weighting strategy used in the 2nd run of the preliminary experiment is just one possible strategy that could be used. In this section we present other possible weighting strategies that could be used, relying on the results of the preliminary experiment. Weights (W) are assigned to each of the arguments in the following manner:

- We assign an apriori weight (W) of 1 to each argument
- We count all positive feedback, i.e., "Very Confident (1)" (F_{pos})

Table 6.9 Comparative results when the GeoMantis system is used on the 10 stories from dataset Crowd_npr with the original strategies, i.e., NUMR and PERCR and their weighted expansion, i.e., NUMR$_w$ and PERCR$_w$.

#	Dataset	Strategy	$A_1(\%)$	$A_2(\%)$
CT1	**Crowd_npr**	**NUMR$_w$**	**70.00**	**100.00**
CT2	**Crowd_npr**	**PERCR$_w$**	**80.00**	**100.00**
CT4	**Crowd_npr**	**NUMR**	**50.00**	**80.00**
CT3	**Crowd_npr**	**PERCR**	**60.00**	**100.00**

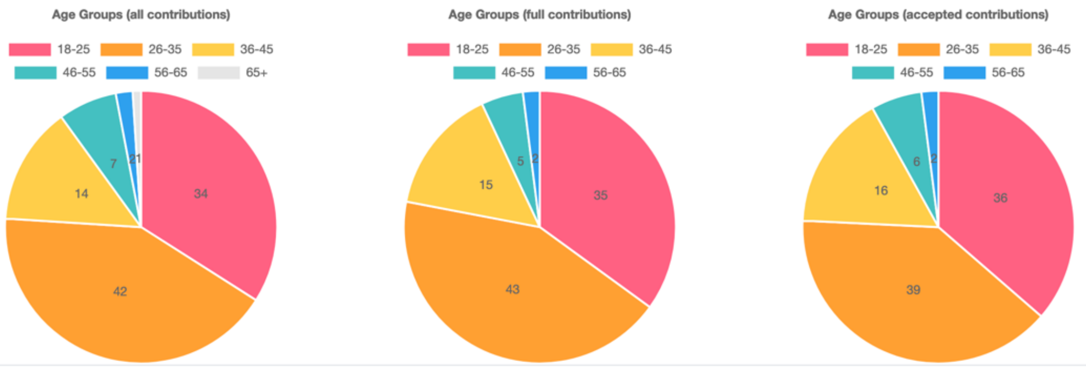


Figure 6.14 Analysis of the age groups of crowd-workers. for each of the three contribution states (all, completed, accepted).

- We count all negative feedback, i.e., “Not Very Confident (-1)” (F_{neg})
- We count all neutral feedback, i.e., “Somewhat Confident (0)” (F_{neu})

Nine different strategies were identified based on the observations we made from the preliminary experiments and we present them in the list below:

Strategy S_{X_1}:

- if $F_{pos} > F_{neg} + F_{neu}$ then W=W_p.
- if $F_{pos} < F_{neg} + F_{neu}$ then W=W_n.
- if $F_{pos} = F_{neg} + F_{neu}$ then W=W_{ne}.

Strategy S_{X_2}:

- if $F_{pos} > F_{neg}$ then W=W_p.
- if $F_{pos} < F_{neg}$ then W=W_n.
- if $F_{pos} = F_{neg}$ then W=W_{ne}.

Strategy S_{X_3}:

- if $F_{pos} + F_{neu} > F_{neg}$ then W=W_p.
- if $F_{pos} + F_{neu} < F_{neg}$ then W=W_n.
- if $F_{pos} + F_{neu} = F_{neg}$ then W=W_{ne}.

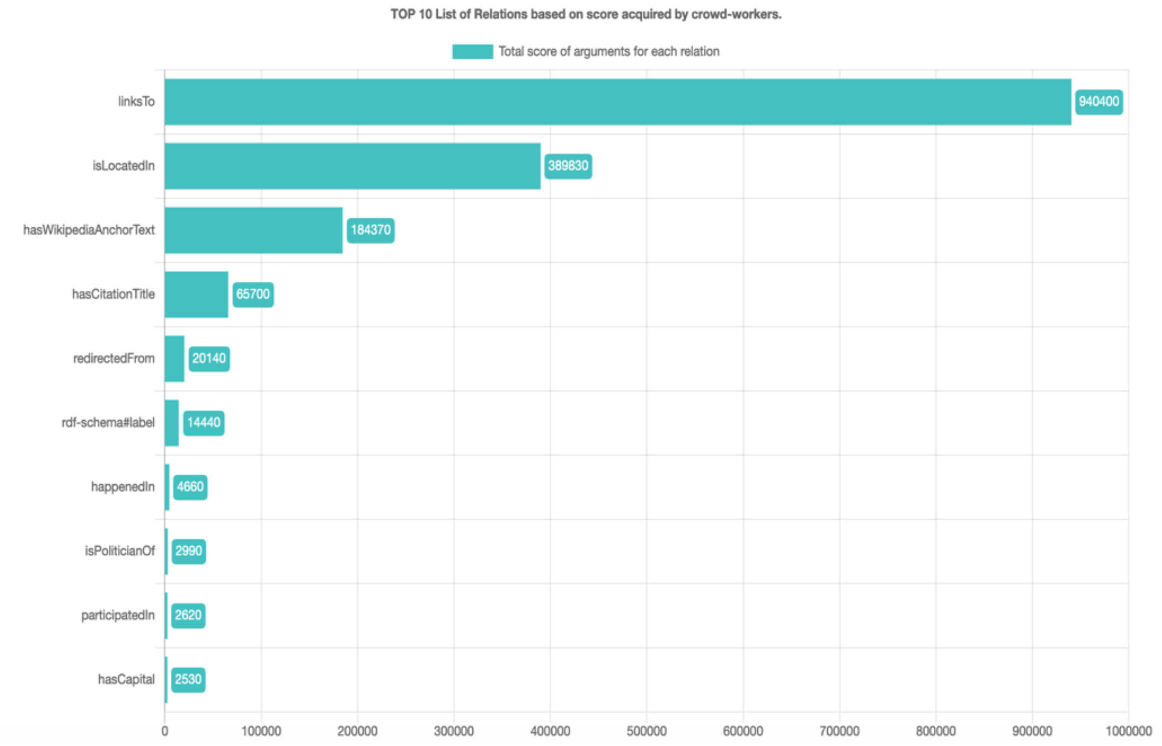


Figure 6.15 The score, i.e., the sum of weights, obtained by crowd-workers per argument relation. Top 10 results are presented.

Where $X \in \{1,2,3\}$.

For strategies S_{1_1}, S_{1_2}, and S_{1_3} we assign both positive ($W_p = 600$) and negative weights ($W_n = -600$) in a symmetrical way and for neutral evaluations the weight of the argument remains intact ($W_{ne} = 1$).

For strategies S_{2_1}, S_{2_2}, and S_{2_3} we assign positive integer weights ($W_p = 600$) to positive evaluations, for negative evaluations the weight of the argument remains intact ($W_n = 1$) and for neutral evaluations we assign a positive integer weight, less than the one assigned to positive evaluations ($W_{ne} = 100$).

For strategies S_{3_1}, S_{3_2}, and S_{3_3} we assign positive integer weights ($W_p = 600$) to positive evaluations, negative evaluations are assinged a zero weight ($W_n = 0$) and for neutral evaluations the weight of the argument remains intact ($W_{ne} = 1$).

The value of 600 and 100 were identified heuristically by applying different values of weights in the various strategies and testing them using the query answering strategies during the preliminary experiments.

An additional set of weighting strategies (SC_{X_X}) are generated from the selection of arguments that were evaluated by workers who stated in their profile that they originate or are confident in contributing for the same country as the one the argument supports. These

Table 6.10 Results from the λ selection process for each strategy.

#	Range	Strategy	Crowd_1	Crowd_2	Crowd_3	Crowd_4	λ
1	$\lambda \in \mathbb{R}_{\geq 0}$	NUMR	160	39	10	131	0.5
2	$\lambda \in \mathbb{R}_{\geq 0}$	PERCR	210	211	74	83	1.3
3	$\lambda \in \mathbb{R}_{\geq 0}$	TF-IDF	294	260	47	81	0.0102

weighting strategies follow the same rules as the ones presented earlier and differ only on the source of the arguments.

6.5.4 Experimental Setup

The next step after preparing and testing the experimental workflow and the respective components, was to lunch an experiment to test if our crowdsourcing methodology can yield better results when compared to the results of the original methodology used by GeoMantis.

We took under consideration all GeoMantis strategies and created a broad coverage dataset using the four `Crowd_npr` subsets (cf. Section 6.5.3) and the three GeoMantis strategies. More specifically, we selected stories from each of the four subsets of `Crowd_npr` using each time one of the 3 strategies, i.e., NUMR, PERCR and TF-IDF. These results to the generation of 12 subsets. For each of these 12 subsets, we retrieve stories based on a λ per strategy. In Table 6.10, the selection of the parameters is depicted for each strategy, as well as the number of stories per subset.

Next, we needed to choose a number of stories from the eval_npr dataset that are unique per subset. For that purpose we designed an automated process that randomly chooses stories per strategy that follow the four subsets constraints. The selection process was repeated until all 12 subsets chosen were unique in terms of stories and where that was not possible, we would choose the maximum possible subset. 71 unique stories were chosen which form the `Crowd_npr_diverse` dataset.

To calculate the arguments used, we applied the 3 strategies on the `Crowd_npr_diverse` dataset. The amount of arguments activated for these 71 stories to calculate A_1 and A_2 is 434,562 (178,469 unique). 63% of these arguments was selected and loaded to the crowdsourcing module for evaluation. The acceptance threshold was raised to 70% for the validation test meaning that all contributions below that threshold were not accepted.

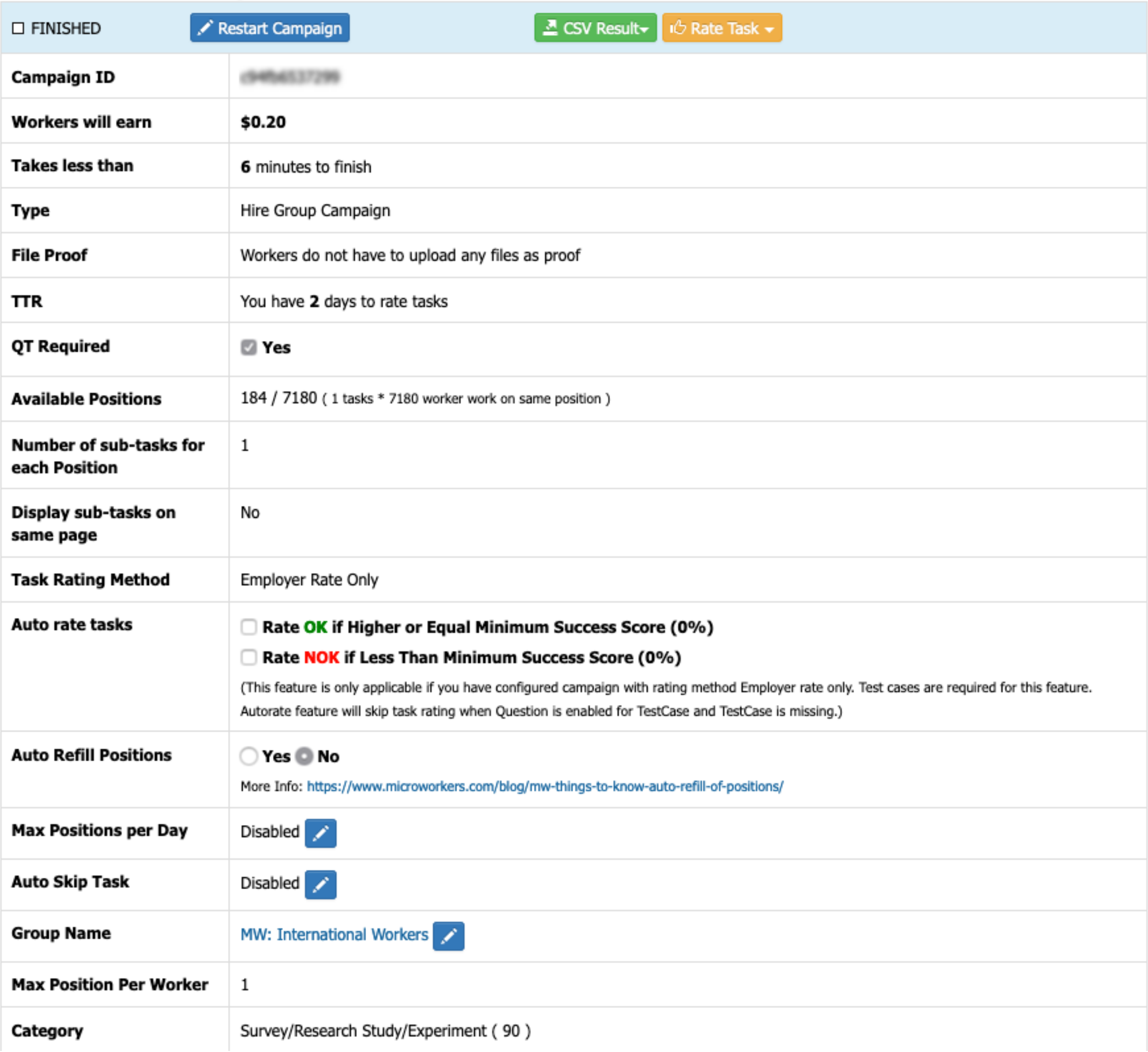
microWorkers Platform Setup

In this section we provide insights on the microWorkers platform campaign setup. To start a crowdsourcing task at the microWorkers platform a user first needs to create a campaign. There are 2 types of campaigns; the "Hire Group" campaign and the "Basic" campaign. The former type of campaign, is used for assigning jobs only to a specific group of crowd-workers which is selected before the start of the campaign. There is also the option to create groups of workers and assign tasks to that particular Group, e.g, "All International workers", "All European workers" or a custom made group of worker. The latter allows the assignment of tasks to workers originating from specific countries, having the option to include or exclude countries from the selected zone. There is no option to exclude crowd-workers from each of the selected countries. For our case, we chose the "Hire Group" campaign choosing the "All International workers" groups which included 1,346,882 crowd-workers.

Next, we needed to set the TTF (Time-To-Finish), which is the amount of time expected for a worker to complete the task. Based on the results we received from the preliminary experiment (cf. Table 6.8), it was set at 6 minutes. During the experiment setup we also needed to state the TTR (Time-To-Rate), i.e., the number of days allowed to rate tasks. Choosing a low value is a good incentive for a crowd-worker to perform the task as their payment will be processed earlier than tasks with higher TTR. We set that to 2 days, while the proposed maximum is 7. Next, we set the *Available positions* for the task to 7180, as this is an estimate of the number of crowd-workers needed to complete this task. Additionally, we added the amount each worker will earn when they successfully complete the task. We chose to pay \$0.20 for each completed task. The amount was decided after checking the average payment of other tasks with a similar TTF available at that time. We also took under consideration the results from the previous short experiment. The average payment of the other tasks was \$0.15 so we increased that to \$0.20 to add an additional incentive for crowd-workers to choose our task.

The last part of the information needed before launching the campaign is the category of the crowdsourcing task. For our experiment the chosen category is "Survey/Research Study/Experiment" which allows crowd-workers to visit an external site and complete the task. A template also needs to be created (cf. Figure 6.17) with instructions, details on the task and the type of information and a placeholder for crowd-workers to enter a verification code when they successfully complete the task. In Figure 6.16, a screenshot of the microWorkers platform setup is presented, depicting part of the information needed to launch the crowdsourcing campaign.

The instructions page (from the MicroWorkers platform) contained a link to a detailed information page to get each participant's informed consent. This page included details on the



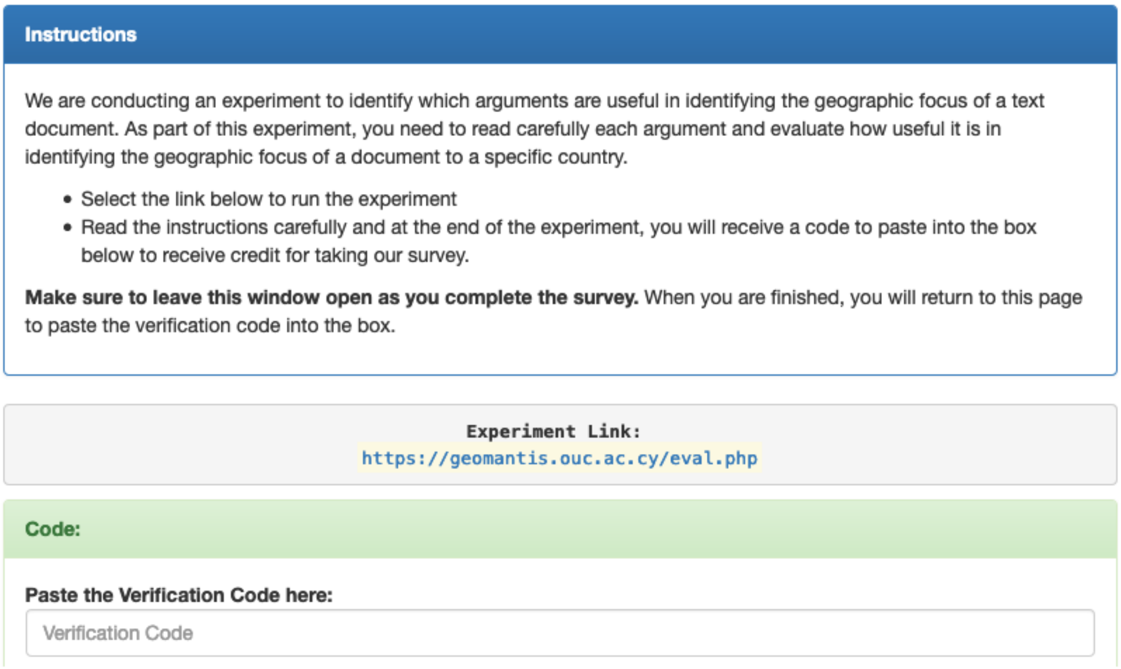
Campaign ID	[redacted]
Workers will earn	**$0.20**
Takes less than	**6** minutes to finish
Type	Hire Group Campaign
File Proof	Workers do not have to upload any files as proof
TTR	You have **2** days to rate tasks
QT Required	☑ **Yes**
Available Positions	184 / 7180 (1 tasks * 7180 worker work on same position)
Number of sub-tasks for each Position	1
Display sub-tasks on same page	No
Task Rating Method	Employer Rate Only
Auto rate tasks	☐ **Rate OK if Higher or Equal Minimum Success Score (0%)** ☐ **Rate NOK if Less Than Minimum Success Score (0%)** (This feature is only applicable if you have configured campaign with rating method Employer rate only. Test cases are required for this feature. Autorate feature will skip task rating when Question is enabled for TestCase and TestCase is missing.)
Auto Refill Positions	○ **Yes** ● **No** More Info: https://www.microworkers.com/blog/mw-things-to-know-auto-refill-of-positions/
Max Positions per Day	Disabled
Auto Skip Task	Disabled
Group Name	MW: International Workers
Max Position Per Worker	1
Category	Survey/Research Study/Experiment (90)

Figure 6.16 A screenshot of the microWorkers platform interface for launching the crowdsourcing campaign.

research task, the researcher and organization who is responsible for running the experiment, the estimated duration for completing the task, the number of arguments they need to evaluate, the payment they will receive for the task and the validation test requirements.

Crowd-workers were also compensated for the validation questions, as these were included in the 100 arguments needed to be evaluated. Crowd-workers who did not pass the validation test or did not complete the evaluation of all arguments were not compensated. This was clearly stated in both the instructions text and the informed consent page and crowd-workers could choose at any time to skip the task.

There was also a section informing the crowd-worker on what data will be collected and how these will be used for our research. More specifically, gathered data included



Instructions

We are conducting an experiment to identify which arguments are useful in identifying the geographic focus of a text document. As part of this experiment, you need to read carefully each argument and evaluate how useful it is in identifying the geographic focus of a document to a specific country.

- Select the link below to run the experiment
- Read the instructions carefully and at the end of the experiment, you will receive a code to paste into the box below to receive credit for taking our survey.

Make sure to leave this window open as you complete the survey. When you are finished, you will return to this page to paste the verification code into the box.

Experiment Link:
https://geomantis.ouc.ac.cy/eval.php

Code:

Paste the Verification Code here:

Verification Code

Figure 6.17 A screenshot of the experiment template, i.e., the first screen with instructions for the crowd-worker to read, click on the GeoMantis evaluation site and then submit the task code for payment.

the worker's id (not name or username), country of origin, country they feel confident in contributing, their age group and the evaluations for each presented argument. At the end of the experiment phase, worker's id was removed and anonymized so that gathered data could not be used to identify any of the crowd-workers from the platform. Hence, the published dataset includes only the anonymized data.

Crowd-workers Analysis

The experiment was conducted from February 18, 2020 until March 12, 2020, a total of 24 days, through the microWorkers platform. A total of 8,341 crowd-workers contributed, of which 6,112 (73.28%) provided accepted contributions, i. e., contributions that passed the threshold of 70% at the validation test. For one of the crowd-workers, the contribution time exceeded 23 hours and we removed both the worker and the contributions from the accepted data list, leaving a total of 6,111 crowd-workers with accepted contributions. In Figure 6.20 the number of registrations per day and the number of contributions per day are depicted.

The majority of contributors are from Asia (e.g., Bangladesh, India, Pakistan). In Figure 6.18 the top 25 countries of crowd-workers' origin is presented. In total, crowd-workers come from 133 countries. Additionally, crowd-workers were confident in contributing for 154

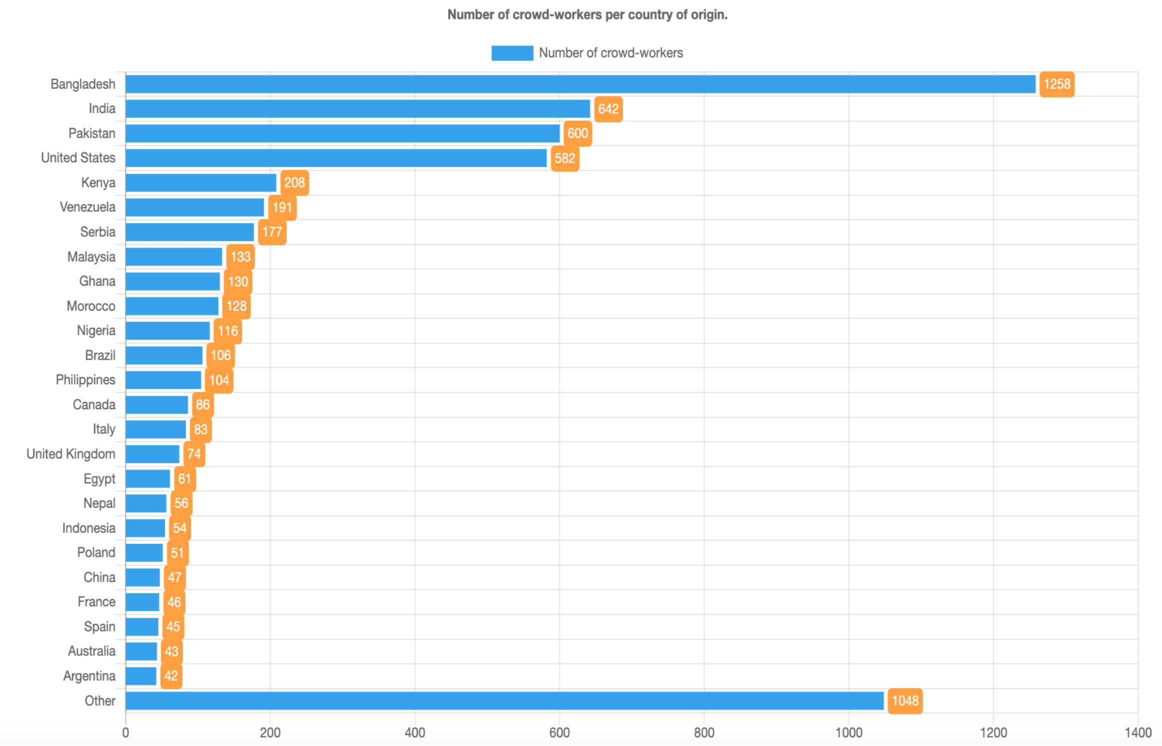


Figure 6.18 In this graph the number of crowd-workers per country is depicted.

countries. The countries for which crowd-workers felt confident in contributing is depicted in Figure 6.19. Similar to the country of origin the majority of crowd-workers are confident in contributing in countries from Asia and US. From 6,111 crowd-workers, 4,396 (71.94%) are confident in contributing for their country of origin and 1,715 (28.06%) were confident in contributing to a country other than their country of origin.

Crowd-workers completed a session, i.e., 100 argument evaluations, on average in 6 minutes with $\sigma = 8$ minutes, and for each argument evaluation they spent on average 4 seconds with $\sigma = 25$ seconds. Table 6.11 summarizes the crowd-workers contributions. In terms of age range, 76% of crowd-workers are between 18 and 35 years old.

The demographics (country of origin and age group) presented above are inline with the results from the survey conducted in the work of Martin et al. (2017) regarding the microWorkers site, showing a good understanding of who the crowdworkers are.

In terms of evaluations, the majority of arguments (59697) received 3 evaluations from crowd-workers. In Figure 6.21 the number of arguments per number of evaluations is depicted. There are also arguments which received more than 3 evaluations since the system presents sometime the same argument for evaluation to different users who are engaged in the task at the same time.

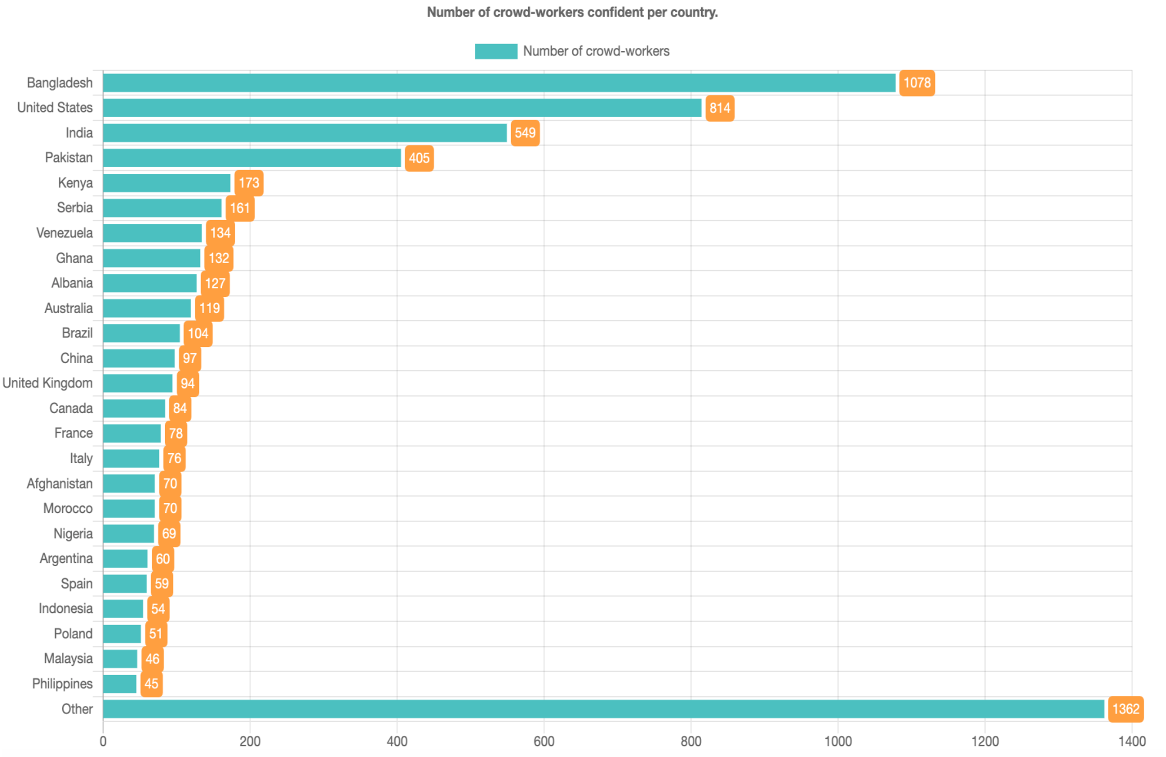
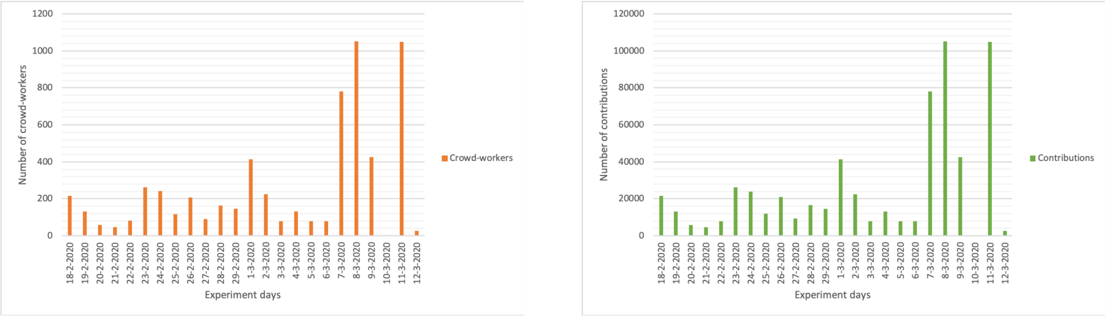


Figure 6.19 In this graph the number of crowd-workers confident in contributing for a specific country is depicted



(a) Number crowd-workers registrations per day.

(b) Number of crowd-workers' contributions per day.

Figure 6.20 Graphical representation of the number of crowd-workers and the number of contributions per experiment day. On day 10/03/2020 there was a network outage so only 3 crowd-workers registered.

Table 6.11 Crowd-worker statistics for evaluating GeoMantis on `Crowd_npr_diverse` dataset.

Number of Stories	71
Total number of arguments to evaluate	178,469
Number of Arguments evaluated	113,219 (63.44%)
Minimum number of evaluators per argument	3
Test acceptance percentage	≥70%
Time needed for evaluation	24 days
Number of Crowd Workers (all contributions)	8,341
Number of Crowd workers (completed contributions)	6,796
Number of Crowd workers (accepted contributions)	6,111
Avg time per contribution (all contributions)	6 minutes
Avg time per contribution (completed contributions)	6 minutes
Avg time per contribution (accepted contributions)	6 minutes
Avg time per evaluation (all contributions)	4 seconds
Avg time per evaluation (completed contributions)	4 seconds
Avg time per evaluation (accepted contributions)	4 seconds
Amount payed per worker	$0.20 USD

6.5.5 Experimental Results

After recording the evaluations of the arguments we added weights to each argument using one of the proposed weighting strategies presented in Section 6.5.3. Then we use the GeoMantis experimental interface to predict the geographic focus for the stories in the `Crowd_npr_diverse` dataset using the weighted query answering strategies (cf. Section 6.5.1).

The first observation is that the TF-IDF$_\text{w}$ strategy outperforms the other two (NUMR$_\text{w}$ and PERCR$_\text{w}$) and the prevailing weighted versions (S_{2_1} and S_{3_1}) of the strategies outperform the original versions when applied on the `Crowd_npr_diverse` dataset.

When the S_{1_1}, S_{2_1} and S_{3_1} weighting strategies were used, 82,951 (73.27%) arguments were positively evaluated, 3,840 (3.39%) arguments were neither positively nor negatively evaluated and 26,428 (23.34%) arguments were negatively evaluated. When the S_{1_2}, S_{2_2} and S_{3_2} weighting strategies were used, 91,336 (80.67%) arguments were positively evaluated, 9,250 (8.17%) arguments were neither positively nor negatively evaluated and 12,633 (11.16%) arguments were negatively evaluated. When the S_{1_3}, S_{2_3} and S_{3_3} weighting strategies were used, 101,729 (89.85%) arguments were positively evaluated, 1,888 (1.67%)

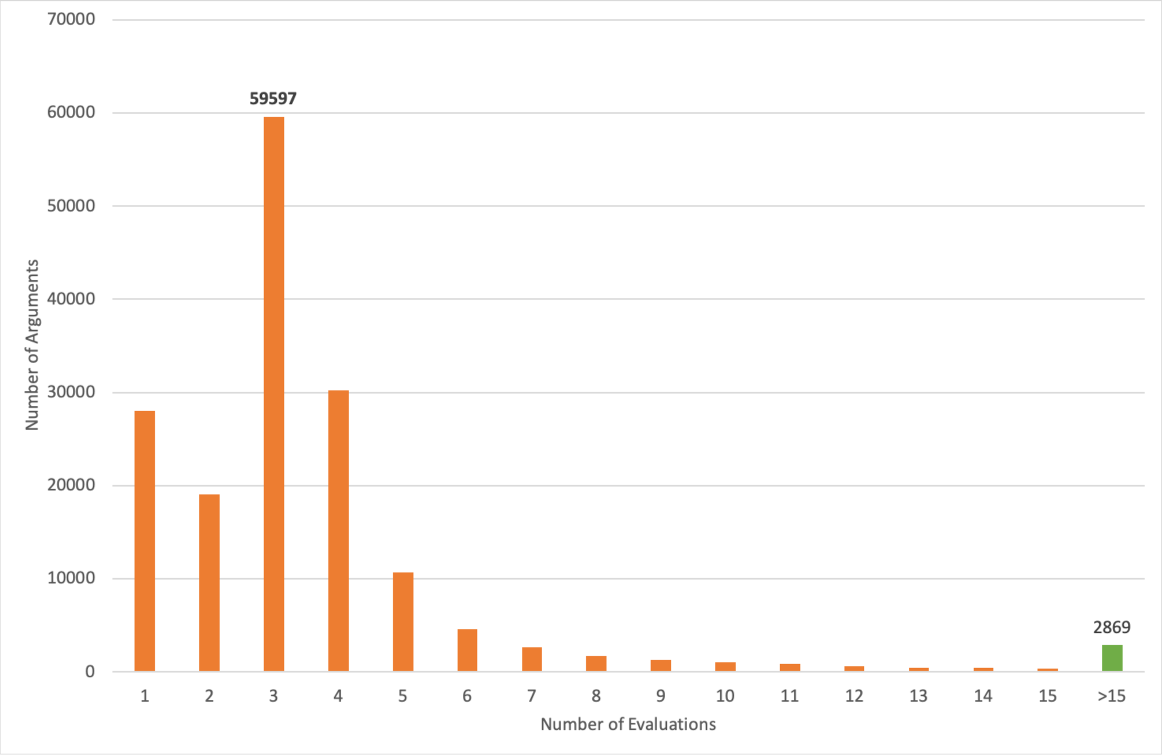


Figure 6.21 In this graph the number of arguments over the number of evaluations is depicted. In green, the number of arguments with more than 15 evaluations is presented.

arguments were neither positively nor negatively evaluated and 9,602 (8.48%) arguments were negatively evaluated.

When the SC_{1_1}, SC_{2_1} and SC_{3_1} weighting strategies were used, 16,410 (70.49%) arguments were positively evaluated, 1,156 (4.97%) arguments were neither positively nor negatively evaluated and 5,714 (24.54%) arguments were negatively evaluated. When the SC_{1_2}, SC_{2_2} and SC_{3_2} weighting strategies were used, 18,866 (81.04%) arguments were positively evaluated, 1,214 (5.21%) arguments were neither positively nor negatively evaluated and 3,200 (13.75%) arguments were negatively evaluated. When the SC_{1_3}, SC_{2_3} and SC_{3_3} weighting strategies were used, 20,543 (88.24%) arguments were positively evaluated, 608 (2.61%) arguments were neither positively nor negatively evaluated and 2,129 (9.15%) arguments were negatively evaluated.

It is also important to note that the S_{X_X} weighting strategies yield better or the same results to the SC_{X_X} strategies. The SC_{X_X} strategies use evaluations only from crowd-workers who stated that the originate or are confident in contributing to arguments supporting their stated country.

The results of the experiment show an improvement on all query answering strategies for the S_{X_X} weighting strategies, when compared to the original strategies. In Table 6.12

we present the results of the experiments per weighting strategy and per query answering strategy. The highlighted rows show the best performing weighting strategies, i.e S_{2_1} and S_{3_1}. In terms of the query answering strategies, the best performing strategy is the TF-IDF$_\mathrm{w}$, followed by PERCR$_\mathrm{w}$ and then NUMR$_\mathrm{w}$.

Additionally, we applied CLIFF-CLAVIN and Mordecai on the `Crowd_npr_diverse` dataset for comparison reasons. One can easily observe that the prevailing GeoMantis weighting strategies, i.e., S_{2_1} and S_{3_1} outperform both systems. More specifically, when we compare the results from the updated GeoMantis architecture using the S_{2_1} and S_{3_1} strategies and the TF-IDF$_\mathrm{w}$ and PERCR$_\mathrm{w}$ query answering strategies, to that of CLIFF-CLAVIN and Mordecai we observe that our system outperforms both of them. The CLIFF-CLAVIN system, returned an unidentified geographic focus for 16.90% of the news-stories. Moreover, due to the fact that CLIFF-CLAVIN does not order the results and for comparison reasons we calculated A_1 when only one country was returned and it was the correct one and A_7 (where 7 is the maximum number of countries returned for that dataset) when more than one country was returned and the correct one was among them.

6.6 An Example of Applying Argumentation Acceptance Semantics

GeoMantis is also capable to identify the geographic focus of a text document using argumentation approaches (cf. Section 6.2.3). In this section we showcase this capability of GeoMantis on identifying the geographic focus of text from the `Crowd_npr_diverse` dataset comprising 71 stories (cf. Section 6.5.2).

First, we convert triples from the selected knowledge base, i.e., YAGO, into arguments attacking a candidate country for the geographic focus of the document. The argumentation graph is similar to the one presented in Figure 6.2. The generated argumentation graph is used to compute the acceptance semantics of the argumentation framework chosen. Algorithm 4 is used for this purpose.

We generate an argumentation graph for all possible countries and compute the acceptance semantics for the outcome of Algorithm 4. If any of the accepted sets of arguments includes the candidate country then the country is a possible geographic focus for the text document.

This same algorithm can be modified to add weights to the attacks between arguments following the weighted argumentation framework. More specifically, for this example we assigned a weight of 1 on attacks between we can assign weights to each attack based on the evaluations we retrieved (cf. Section 6.5) for each argument. For attacks between countries

Table 6.12 Accuracy at A_1 and A_2 when tested on each of the 3 strategies (NUMR$_\text{w}$, PERCR$_\text{w}$, TF-IDF$_\text{w}$) for the `Crowd_npr_diverse` dataset. The left column depicts the system and weighting strategies presented in Section 6.5.3. On the top rows of the table we present the original strategies of GeoMantis (GM, v1) and the results when these are applied on the `EVAL_npr` dataset and the `Crowd_npr_diverse` dataset, comprising 1000 and 71 stories respectively, for comparison with their weighted versions (GM, v2, S_{X_X}). In the last 2 rows we present results from two widely used systems; CLIFF-CLAVIN and Mordecai, when applied on the `Crowd_npr_diverse` dataset.

System	Dataset	NUMR		PERCR		TF-IDF	
		A_1	A_2	A_1	A_2	A_1	A_2
GM (v1)	`EVAL_npr`	30.50%	47.30%	43.40%	58.60%	55.40%	68.20%
GM (v1)	`Crowd_npr_diverse`	33.80%	56.34%	50.70%	83.10%	84.51%	92.96%
		NUMR$_\text{w}$		PERCR$_\text{w}$		TF-IDF$_\text{w}$	
GM (v2, S_{1_1})	`Crowd_npr_diverse`	43.66%	64.79%	61.97%	84.51%	94.37%	95.77%
GM (v2, S_{1_2})	`Crowd_npr_diverse`	45.07%	64.79%	59.15%	88.73%	94.37%	97.18%
GM (v2, S_{1_3})	`Crowd_npr_diverse`	43.66%	63.38%	53.52%	84.51%	94.37%	97.18%
GM (v2, S_{2_1})	`Crowd_npr_diverse`	42.25%	64.79%	67.61%	90.14%	95.77%	98.59%
GM (v2, S_{2_2})	`Crowd_npr_diverse`	42.25%	63.38%	61.97%	87.32%	95.77%	98.59%
GM (v2, S_{2_3})	`Crowd_npr_diverse`	42.25%	64.79%	60.56%	87.32%	95.77%	98.59%
GM (v2, S_{3_1})	`Crowd_npr_diverse`	42.25%	64.79%	67.61%	90.14%	95.77%	98.59%
GM (v2, S_{3_2})	`Crowd_npr_diverse`	42.25%	64.79%	63.38%	88.73%	95.77%	98.59%
GM (v2, S_{3_3})	`Crowd_npr_diverse`	42.25%	64.79%	60.56%	87.32%	95.77%	98.59%
GM (v2, SC_{1_1})	`Crowd_npr_diverse`	42.25%	57.75%	47.89%	64.79%	84.51%	95.77%
GM (v2, SC_{1_2})	`Crowd_npr_diverse`	42.25%	61.97%	52.11%	73.24%	88.73%	98.59%
GM (v2, SC_{1_3})	`Crowd_npr_diverse`	38.03%	61.97%	50.70%	73.24%	90.14%	98.59%
GM (v2, SC_{2_1})	`Crowd_npr_diverse`	42.25%	60.56%	53.52%	74.65%	90.14%	98.59%
GM (v2, SC_{2_2})	`Crowd_npr_diverse`	39.44%	60.56%	54.93%	74.65%	90.14%	98.59%
GM (v2, SC_{2_3})	`Crowd_npr_diverse`	39.44%	60.56%	52.11%	73.24%	90.14%	98.59%
GM (v2, SC_{3_1})	`Crowd_npr_diverse`	42.25%	61.97%	53.52%	74.65%	90.14%	98.59%
GM (v2, SC_{3_2})	`Crowd_npr_diverse`	39.44%	60.56%	54.93%	74.65%	90.14%	98.59%
GM (v2, SC_{3_3})	`Crowd_npr_diverse`	39.44%	60.56%	54.93%	73.24%	90.14%	98.59%
		A_1		A_2		A_7	
CLIFF-CLAVIN	`Crowd_npr_diverse`	61.97%		-		74.65%	
Mordecai	`Crowd_npr_diverse`	56.33%		70.42%		76.06%	

and attacks between words and possible countries of focus we assign a weight of 1. For attacks between arguments from YAGO and words we assign a weight of 1 for neutral and negative evaluations of the arguments and a weight of 600 for positive.

In Table 6.13 we present the results of applying argumentation semantics using QAE2 on the `Crowd_npr_diverse` dataset. The accuracy (AA_1) of the system is defined as $AA_1 = \frac{N_1}{C}$, where, N_1 denotes the number of correct assignments of the target country, i.e, the target

Algorithm 4 Creating an argumentation graph from knowledge base.

```
procedure CreateArgumentationGraph(KB)
    PossibleCountryCodes ← retrievePossibleCountryCodes(KB)
    for each CountryCode in PossibleCountryCodes do
        createArgument(countryCode)
        countryWords ← RetrieveCountryWords(NOT(countryCode))
        for each countryWord in countryWords do
            createArgument(countryWord)
            createAttack(countryWord,countryCode)
            SupportTriples ← RetrieveSupportTriples(countryCode,countryWord)
            for each SupportTriple in SupportTriples do
                createArgument(SupportTriple)
                createAttack(SupportTriple,countryWord)
            end for
        end for
    end for
     // The following loop creates attacks between arguments for each possible country
    for each CountryCode in PossibleCountryCodes do
        createAttack(countryCode,PossibleCountryCodes)
    end for
    return ArgumentationGraphForCountry
end procedure
```

country's argument is included in the accepted set of arguments of the specific extension and no other country's argument is included in that set, and C denotes the number of available documents in the dataset. The accuracy (AA_{all}) of the system is defined as $AA_{all} = \frac{N}{C}$, where, N denotes the number of correct assignments of the target country, i.e, the target country's argument is included in the accepted set of arguments of the specific extension and other country's arguments could also be included in that set, and C denotes the number of available documents in the dataset. U indicates the number of stories that there are no country's arguments in any of the accepted sets of arguments of the specific extension. The QAE2 engine is able to report if a country could be accepted as a geographic focus for a document, but it cannot create an order list of countries as the QAE1 does. Results between the two engines are not comparable since QAE1 outcome is an ordered list of countries and QAE2 outcome is the existence or not of the country argument in the arguments results sets. We will not attempt a comparison between the two engines as this is not the point of this section. What is worth noting, is that i) an argumentation based engine can also provide solutions for the problem of identifying the geographic focus of a document, and ii) crowdsourcing

Table 6.13 Accuracy measured at AA_1 and AA_{all} when tested using the QAE2 for the `Crowd_npr_diverse` dataset. The left column depicts the argumentation framework used, the semantic used for computation and its parameters.

Framework	Semantic	AA_1	AA_{all}	U
AAF	Grounded	40.85%	40.85%	36 (50.70%)
AAF	Stable	43.66%	43.66%	14 (19.72%)
AAF	Complete	45.07%	45.07%	14 (19.72%)
WAF	Stable ($\alpha = 1, \gamma = 1$)	56.34%	56.34%	3 (4.23%)
WAF	Stable ($\alpha = 1, \gamma = 0$)	56.34%	56.35%	3 (4.23%)
WAF	Complete ($\alpha = 1, \gamma = 1$)	40.85%	40.85%	25 (35.21%)
WAF	Complete ($\alpha = 1, \gamma = 0$)	39.44%	39.44%	25 (35.21%)
WAF	Stable ($\alpha = 2, \gamma = 1$)	39.44%	57.55%	5 (7.04%)
WAF	Stable ($\alpha = 2, \gamma = 2$)	40.85%	59.15%	5(7.04%)
WAF	Stable ($\alpha = 2, \gamma = 0$)	38.03%	56.34%	5(7.04%)
WAF	Complete ($\alpha = 2, \gamma = 0$)	26.76%	26.76%	21 (29.58%)
WAF	Complete ($\alpha = 2, \gamma = 1$)	26.76%	26.76%	21 (29.58%)
WAF	Complete ($\alpha = 2, \gamma = 2$)	26.76%	26.76%	21 (29.58%)
WAF	Stable ($\alpha = 10, \gamma = 1$)	8.57%	85.71%	0 (0%)

methods used, i.e., using crowd-workers to evaluate arguments, can yield better results than methods which do not employ these techniques (cf. AAF vs WAF results in Table 6.13).

6.7 Discussion

Story understanding is not only about question answering but also about understanding other properties of a story. In this chapter we investigated ways to address the problem of identifying the geographic focus of a story. More specifically, we developed a system and a methodology that uses crowdsourced knowledge from popular knowledge bases to provide arguments which support the country of focus for a certain story. Moreover, we expanded this methodology with evaluations of these arguments from the crowd, providing evidences that an approach that combines techniques provides better results, as we have presented in Chapter 5. Based on the experiments described in this Chapter, we provide evidences that argumentation is indeed a good method for representing and reasoning with commonsense knowledge. Furthermore, acquired knowledge can be used to answer questions using automated reasoning systems. In particular, this knowledge can be used to answer questions when the answer is not explicitly found in the story text, providing evidence that this method is a good approach to evaluate the acquired knowledge.

The experiments we conducted and the evaluation process results, show that the methodology chosen, i.e., using general purpose ontologies, is applicable and well suited for the problem of identifying the geographic focus of documents that do not explicitly mention the target country. In this work, a number of strategies were tested and the one that presents better results, is the ordering of the list of countries according to the TF-IDF algorithm, in descending order. In terms of knowledge source, the YAGO ontology results present a greater accuracy than the ConceptNet ontology results. Moreover, the usage of named entities filtering on the document text increases the performance and the accuracy of target country identification.

Moreover, the extended version of GeoMantis, i.e., the one that utilizes crowd-workers to evaluate arguments and apply weighted strategies to identify the geographic focus of a story, yields better results than the original version. In particular the weighted TF-IDF strategy (TF-IDF_w) outperforms the other strategies and in terms of weighting strategy, the ones that use positive weights outperform the ones that use negative weights.

The field of text comprehension can benefit from the recent advances in Artificial Intelligence (Hermann et al., 2015). Researchers started growing concern in algorithm transparency and accountability, since most newly developed “intelligent” systems and algorithms are opaque black boxes where you give an input and the output is presented without actually presenting their “thinking” process. Algorithms should provide transparency (Dignum, 2017) on their methods, results, and explanations. GeoMantis is inline with that direction, since it exposes its query answering strategy and can provide explanations on why a specific geographic focus of a document was chosen, i.e., present the arguments that were activated from the story. The explanatory role of such systems, with respect to the target natural cognitive systems they take as source of inspiration, is highlighted in the work of Lieto and Radicioni (2016).

Currently, there are not many systems dedicated for the task of identifying the geographic focus of a text document. The majority of the available systems are basically geoparsers that offer focus identification as an additional feature of their primary purpose and they rely on text that has a good amount of place mentions in it. When these systems are tested on documents that have few place mentions, they perform poorly in terms of accuracy, as opposed to the high accuracy they present when tested on datasets that have mentions of locations. This limitation is waived in GeoMantis, which does not rely exclusively on place mentions to work, but uses any type of general-purpose knowledge that can be found in generic ontologies.

GeoMantis is currently able to identify country-level geographic focus, but it can be expanded to handle other levels (e.g., administrative area, city), as long as the relevant knowledge is available. The techniques used for news stories, could also apply to other types of documents such as myths, novels, legal documents, etc. This line of research can also find applications for document classification and geographic knowledge extraction from text. Moreover, it can be used with techniques for linking image and text-based contents together, for document management tasks (Cristani and Tomazzoli, 2016).

Crowdsourcing approaches like GWAPs or hybrid solutions such as the GWAP presented in Chapter 5, could also be applied in future versions of the system for fact disambiguation. The integration of other ontologies or knowledge bases with GeoMantis, like the one generated from the Never Ending Language Learner (Mitchell et al., 2015), DBpedia (Lehmann et al., 2015), Wikidata (Erxleben et al., 2014) or their combination, could also be explored.

We believe that the GeoMantis system can be used in several application scenarios, such as document searching and tagging, games (e.g., taboo game challenges), and news categorization. Its extendable architecture enables the addition of new functionality and new sources of knowledge and also the integration with other systems. GeoMantis could also be used in conjunction with other systems to return results in cases where the other systems are not able to return any.

Related Publications:

(1) Christos T. Rodosthenous and Loizos Michael. A Crowdsourcing Methodology for Improved Geographic Focus Identification of News-Stories. International Conference on Agents and Artificial Intelligence, 2021.

(2) Christos Rodosthenous and Loizos Michael. Using Generic Ontologies to Infer the Geographic Focus of Text. In Jaap van den Herik and Ana Paula Rocha. Agents and Artificial Intelligence, pages 223–246, Cham, 2019. Springer International Publishing.

(3) Christos T. Rodosthenous and Loizos Michael. GeoMantis: Inferring the Geographic Focus of Text using Knowledge Bases. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pages 111–121, Madeira, Portugal, 2018. SciTePress.

(4) Christos T. Rodosthenous and Loizos Michael. Inferring the Geographic Focus of Stories Using Crowdsourced Knowledge Bases. Presented at the 1st International Workshop on Cognition and Artificial Intelligence for Human-Centred Design (CAID 2017), Melbourne, Australia, 2017.

7

Conclusions and Future Work

> *"The real risk with AI isn't malice but competence. A superintelligent AI will be extremely good at accomplishing its goals, and if those goals aren't aligned with ours, we're in trouble. You're probably not an evil ant-hater who steps on ants out of malice, but if you're in charge of a hydroelectric green energy project and there's an anthill in the region to be flooded, too bad for the ants. Let's not place humanity in the position of those ants."*
>
> – Stephen Hawking, *reddit post (2015)*

7.1 Summary of Thesis

The ability to comprehend texts and natural language in general, is a long awaited promise of artificial intelligence. Since its early days, artificial intelligence researchers invested a great amount or resources to investigate how this could be materialized (cf. Chapter 1). They tested a number of methods such as logic-based systems, neural networks and machine learning algorithms. Each of these methods presented some promising results, but none was capable to deeply comprehend a text, i.e., to exhibit basic commonsense reasoning capabilities. Moreover, these methods were not able to "explain" how they reached to a certain conclusion. This was due to the lack of commonsense knowledge and proper ways to represent this knowledge. Human language is very expressive and ambiguous and when we communicate, we often omit a number of details that are considered as commonsense knowledge. The latter makes it extremely difficult for machines to understand a text document. To this end, in this thesis we investigated how we can acquire commonsense knowledge using visual interfaces and games, represent it in an appropriate format using argumentation and apply this knowledge for the task of answering questions on stories where their answer is not explicitly found in the story text.

In Chapter 2 we provide an overview of the current state of affairs in story understanding and text comprehension in general, highlighting previous work, systems developed, corpora available, and techniques. We introduced the reader to the terms of crowdsourcing and present how it can be applied for commonsense knowledge acquisition. The special case of Games With A Purpose is also presented, which allows implicit knowledge acquisition and lays the foundation for the work presented in the next chapters.

Next, we focus on how we can represent commonsense knowledge using techniques that match the nature of commonsense knowledge. More specifically, in Chapter 3 we give an overview of argumentation frameworks such as Dung's Abstract Argumentation Framework (AAF) and the Weighted Argumentation Framework (WAF), that are used in the following chapters to represent commonsense knowledge in the form of arguments, laying the bricks for understanding how knowledge is used in the context of story understanding. We also relate to work specific to story understanding and argumentation, by concentrating on the STory comprehension through ARgumentation (STAR) system that is able to perform automatic story comprehension.

We then present our work on Web-STAR, a platform which facilitates story understanding by providing a range of tools for: reading a story both in natural language and in symbolic format, writing questions, adding commonsense knowledge, and finally presenting the comprehension model to the user in a graphical way, benefiting from the STAR system. The Web-STAR platform is presented in Chapter 4 and can be utilized by both expert and non-expert users as it offers the ability to convert a story from natural language to logic based format and create and represent knowledge in an easy to read and understand graph. A thorough evaluation was conducted to examine how friendly the user interface is and the results showed that the interface was indeed helpful. This was one of the limitations identified during the bibliographic review in most of the story understanding systems developed so far and this is an area where our work contributes as only a handful of systems provide an easy to use interface. Moreover, the evaluation of theWeb-STAR platform provides evidence that visual interfaces are a good approach for knowledge acquisition from non-expert users. In particular, these type of interfaces also provide the means to acquire knowledge in a format that is also suitable for automated story understanding.

The results made us more determined to examine how we can gather commonsense knowledge from non-expert users by avoiding the tedious task of manually adding it and provide a meaningful way to evaluate and use the acquired knowledge. In Chapter 5 we try to tackle the problem by using crowdsourcing techniques, so we investigate how human workers can help in acquiring suitable knowledge for understanding stories. The first step to tackle the problem is the design of a platform/framework for developing GWAPs specifically for knowledge acquisition. This platform offers a number of gamification components to include in a game and a number of integrations such as the ability to connect with the STAR reasoning engine, query SPARQL endpoints from knowledge bases and other API enabled systems. In fact, the platform was successfully utilized for the development of two GWAPs and for commonsense knowledge acquisition experiments. More specifically, we presented two distinct approaches, one that was fully crowdsourced and a hybrid one, i.e., both crowdsourcing and machine aided. Both the methodology used and the outcome, showed that crowdsourcing is a good approach for acquiring knowledge specifically for story understanding but it presents better results when crowdsourcing is combined with other automated methods to check and validate the outcome. In particular, we provided evidence that this hybrid approach can be used for acquiring simple and general commonsense knowledge that can be used by automated story understanding systems, since we presented an experiment where we use this knowledge to answer questions on unseen stories where their answer is not explicitly found in the story text.

Towards the end of this thesis, we focus on utilizing existing crowdsourced commonsense knowledge which is included in knowledge bases and ontologies such as ConceptNet and YAGO. We focus on understanding stories and more specifically on identifying the geographic focus of a news story for the special case where no explicit mention of the place is present in the text. This time we shift our efforts to the application of the crowdsourced knowledge. We developed a system called GeoMantis which includes mechanisms and strategies that can identify the geographic focus of a text at the country level. We also performed a number of experiments to propose a suitable method and mechanism that produces good results in terms of accuracy and outperform the results from other similar systems. We went a step further and designed a crowdsourcing evaluation method where arguments supporting a country of mentioned are evaluated by the crowd using monetary incentives and a modified, weighted version of the GeoMantis strategies are applied, producing even better results in terms of accuracy. Furthermore, we showcase an argumentation based engine included in our system and present the results and how these change when crowdsourcing techniques are applied in our methodology.

7.2 Implications and Applications

The outcome of this thesis includes both implementations of systems and empirical investigation of methods and strategies suitable for acquiring and applying commonsense knowledge

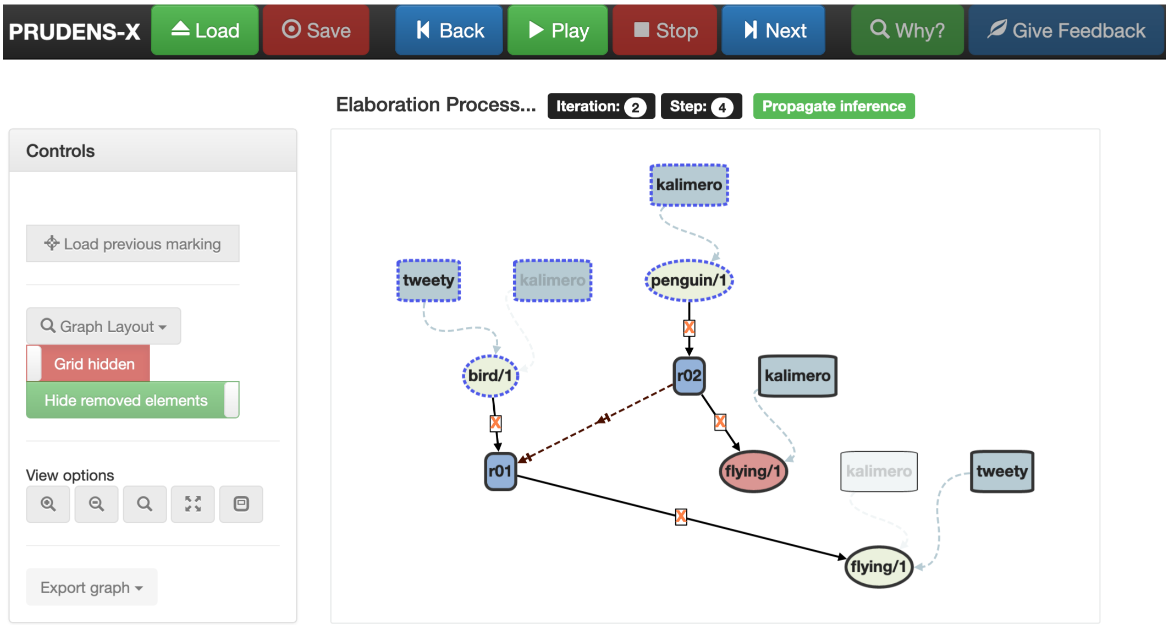


Figure 7.1 A screenshot of the PRUDENS-X platform depicting the graphical representation of the knowledge, the grounding and the result of the algorithm at each step. At the top, the controls of the interface are depicted allowing users to view each step of the algorithm and interact with the system.

for understanding stories. The main applications developed during this thesis are the Web-STAR IDE, the Crowdsourcing Platform, two Games With A Purpose called "Knowledge Coder" and "Robot Trainer" and GeoMantis.

Experiments with the Web-STAR IDE and the GWAPs provide evidence that visual interfaces and games in particular are appropriate methods for acquiring commonsense knowledge. Furthermore, we showed that argumentation is an appropriate method to represent knowledge, as it can handle both causality and implication and can handle the existence of preferences between activated knowledge. More than one frameworks were utilized for that purpose, such as the STAR's logic-based framework, Dung's AAF and Weighted AF used by GeoMantis for identifying the geographic focus. For evaluating and using the acquired knowledge we experimented with question answering on stories, where the answers of those questions were not explicitly found in the story text but were inferred.

For the acquisition of knowledge, crowdsourcing is indeed a good approach to gather vast amount of knowledge and evaluate it. In our experiments we exploited both pure crowdsourcing methods and hybrid ones, the latter of which exhibited better results than pure crowdsourcing methods. This is also verified with experiments where we applied crowdsourced knowledge that resided in knowledge bases for the task of understanding where a story is geographically focused. More specifically, retrieved triples from knowledge bases were converted to arguments for supporting a possible geographic focus of a country and a number of query answering strategies were applied to predict the geographic focus of each story. Furthermore, when these arguments were evaluated by crowd-workers, the results were better in terms of accuracy when compared to the original, not crowdsourced strategy.

In terms of crowdsourcing approaches used in our experiments, we have utilized both implicit crowdsourcing methods such as the two GWAPs developed and explicit crowdsourcing using paid crowdsourcing for evaluating arguments. Each of the two methods have their benefits and caveats. For example GWAPS can increase the retention/engagement period of the users, hence the number of contributions for the specific task as long as they keep updating with new game missions, tasks and rewards. The caveat with implicit crowdsourcing is the lengthy development time for each game and its advertising to appropriate channels. Paid crowdsourcing on the other hand is much more easy for getting started as a researcher does not need to pay much attention on creating intuitive interfaces since each crowd-worker is compensated for their time on that specific task. The downsides of this approach are the continuous costs for paying crowd-workers and the quality of the work performed by crowd-workers. In work of Martin et al. (2017) a number of ethical considerations are reported regarding the treatment of crowd-workers by the paid crowdsourcing platforms, such as Amazon Mechanical Turk and microWorkers.

Additionally, this work also makes use of easy to use web interfaces that can be used by non-experts to utilize story understanding capabilities. This is very useful as the area of story understanding is not limited to developers and logic experts. Developed tools give access to reasoning engines and special symbolic syntax formats to non-expert users that would otherwise require a long learning process before being able to use these tools. The Crowdsourcing platform for instance was used to conduct an experiment for gathering commonsence knowledge in natural language using a specific template (Diakidoy et al., 2017). It was also utilized for a crowdsourcing experiment in language learning for the V-TREL (Lyding et al., 2019; Rodosthenous et al., 2020) architecture where knowledge from ConceptNet was retrieved to generate vocabulary exercises and the contributed answers were used to extend ConceptNet.

Existing work on knowledge acquisition using crowdsourcing and GWAPs is limited in acquiring knowledge in natural language, which cannot be directly used or evaluated by automated story understanding systems. Moreover, these systems evaluate acquired knowledge using agreement methods only. Another shortcoming of existing GWAPs, is that most of them just use gamification methods to disguise the input form-style templates as games and much more work is needed to provide the fun element of a game. Our approach facilitates the acquisition of knowledge, its evaluation using both the crowd and machine methods, and the evaluation of acquired knowledge on real tasks.

7.2.1 Sustainability

One of the main advantages of the presented work is that it provides a sustainable method for knowledge acquisition. The contributed methods and systems can support future endeavors for building systems with commonsense abilities since the systems built are not "closed silos" but interconnect with other systems using webservices and can be easily expanded to include knowledge from other sources.

Commonsense knowledge acquisition is a continuous process and methods using handcrafted knowledge are not viable in the long term as they are time consuming and costly. Games on the other hand are sustainable methods, as long as they are interesting and engage players. In our case, the Robot Trainer game (cf. Chapter 5) can be easily expanded with more stories for players to continue contributing knowledge. Based on the current deployment of the game, players found it interesting (nearly 800 registrations) and as for the GeoMantis system, it can be used also in different domains than that of identifying the geographic focus. In particular, one can retrieve knowledge about sports, cooking, etc. and answer related questions. In sports for example, the GeoMantis system can read stories from sport news sites and identify the team or teams they refer to, and the sport they focus on, without an explicit mention of the relevant information in the story text. Knowledge of the form "Rafael_Nadal `hasOccupation` Tennis_Player" and "Louis_Armstrong_Stadium `instanceOf` tennis_venue" can be used to identify the sport a story refers to where the name of the player or the name of the venue is found in the text.

The developed tools are also delivered using an open source license which allows other researchers in the field to expand research on other domains using or expanding the already developed systems. Furthermore, acquired datasets are publicly available for researchers to download and reuse them for their own research.

7.3 Future Work

Work on acquiring commonsense knowledge suitable for story understanding is ongoing. The fact that numerous researchers and institutions restated their interest in the field, shows how important this line of research is. Future work could include the use of different type of knowledge bases and more specifically the new ones that were recently published

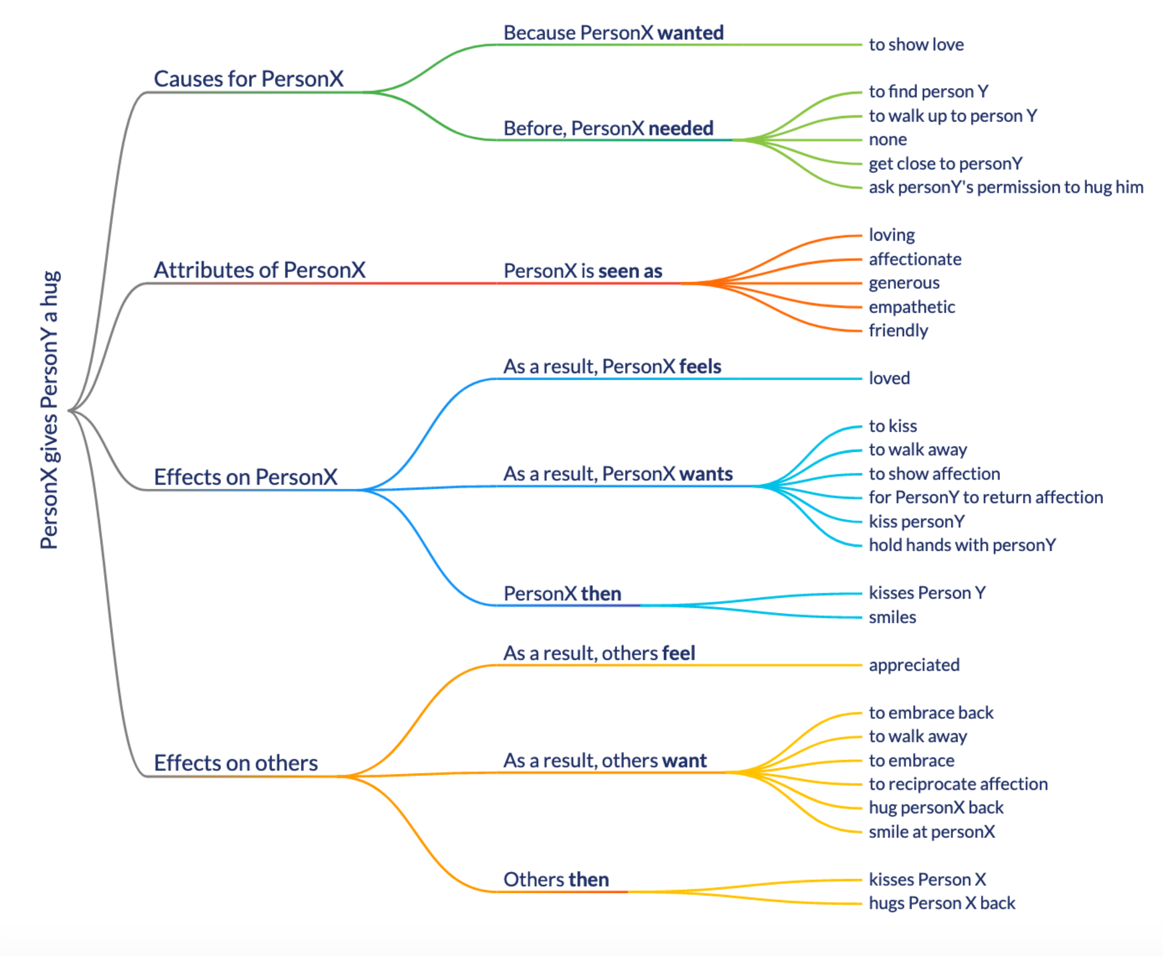


Figure 7.2 A screenshot of the knowledge graph retrieved from ATOMIC when the word "hug" is used for searching.

(e.g., ATOMIC), both to enhance the Web-STAR background knowledge editors and the GeoMantis's knowledge retrieval workflow.

In particular, an interesting extension to the Web-STAR IDE would be the automatic retrieval of knowledge from the ATOMIC knowledge graph, based on the story text. For example, in a story where a person calls another person, the system could propose knowledge rules of the form: `hug(PersonX,PersonY)` causes `feel_appreciated(PersonY)` based on the retrieved knowledge from ATOMIC[1] (cf. Figure 7.2).

In terms of the interface used in the Web-STAR IDE and more specifically the graphical representation of knowledge, we already started reusing the visual components for a web-based interface for a coaching system (Michael, 2017, 2019), i.e., a system that facilitates interaction between an advice-taker and an advice-giver (cf. Figure 7.1). The underlying

[1] https://mosaickg.apps.allenai.org/kg_atomic

system is based on argumentation semantics and the interface allows the visualization of both the reasoning process and the presentation of the results, giving the opportunity to novice users to gain insights on the reasoning mechanism of the system and to get human readable results.

Moreover, one could also use a GWAP to evaluate arguments for the GeoMantis argument evaluation system, instead of using paid crowdsourcing. An idea could be to design a taboo-like game where players would choose the arguments to show to their co-player. That way, arguments that are useful for identifying the country will be implicitly evaluated and players will have fun will performing the task. In the work of Feyisetan et al. (2015) paid crowdsourcing is combined with GWAPs for enhancing workers' contributions and a similar combination could be explored both for GeoMantis and the two GWAPs.

Additionally, other hybrid approaches could be attempted for designing a GWAP, such as using NLP tools for some tasks (e.g., noun and verb identification) and then ask players to verify the results. This could lead to possibly more interesting game missions, hence more game time and engagement from players.

Our work in this thesis, could also find other applications, such as the acquisition of diversified knowledge. Commonsense knowledge, as we have explained earlier, is not strict, but it based on a person's beliefs, rules of thumb and statements that are not always true. This knowledge could vary based on cultural differences, geographic location, cultural stereotypes and it is important to build systems for acquiring knowledge that is diverse. In that spirit, there is early work on diversifying the ATOMIC knowledge base from Acharya et al. (2020) where the authors try to identify cultural differences between two national groups, one from the United States and one from India on rituals, such as birth, marriage, funerals, etc. The methodology we presented, in particular the use of GWAPs could be adapted to accommodate the acquisition of diversified knowledge by taking into account the geographic location and religious beliefs of the contributor, and by verifying the contributed knowledge with contributors from the same geographic location or religion. Knowledge that would otherwise get negative evaluations from other contributors who do not share the same culture or religion will be evaluated positively and make it to the knowledge base, allowing automated systems to answer questions by taking into account this diversified knowledge. Furthermore, knowledge rule preferences would also find use in such a case.

In a similar fashion, the need for diversity in social interactions that transcend geographical and cultural backgrounds is also stressed out in the WeNet - Internet of Us project[2], where the aim of the project is to provide a diversity-aware, machine-mediated paradigm of social relations by developing an online platform (D'Ettole et al., 2020).

[2] The project website is available at: https://www.internetofus.eu

There is a continuous debate on which approach is best for incorporating commonsense knowledge in story understanding systems, a symbolic approach or a deep learning approach. The former approach can provide explainable results and it is data-efficient, but it is sensitive to noise and cannot be applied directly to a text in natural language. The latter approach can be applied in natural language texts, it is resilient to noise, but it is a black box method where the system is not able to explain why it derived to a certain output. There is work on NeuroSymbolic systems, i.e., systems that combine symbolic and neural networks, which use some of the best performing language models and ingest knowledge from commonsense knowledge bases to boost their performance. In the work of Ma et al. (2019) such an approach is presented using BERT and knowledge from ConceptNet. The authors tested this in a number of scenarios and found that under certain conditions, this approach can indeed boost performance. Future work can be directed to ingesting acquired knowledge from the presented GWAPs to a language model and test if this can yield better results in terms of accuracy. In a similar fashion, our work on identifying the geographic focus of a story can also be benefited from such an approach.

To conclude this thesis, we have achieved all the goals set at the start of this work and answered the three research questions using evidences from experiments with human participants, using tools that were developed towards that goal. Since the start of this work, a number of new systems came to surface by the research community, showing the increasing need for ingesting commonsense knowledge in AI systems. Newly developed systems using state-of-the-art language models and deep learning show impressive performance but still lack the ability of human-like understanding, especially in high-level cognitive tasks. From my point of view, the future of AI should be a reconciliation of methods which are cognitively compatible with human thinking and understanding, with machine learning approaches.

Bibliography

Abbott, H. P. (2008). *The Cambridge Introduction to Narrative*. Cambridge University Press.

Acharya, A., Talamadupula, K., and Finlayson, M. A. (2020). Towards An atlas of cultural commonsense for machine reasoning.

Aesop (2009). *Aesop's Fables*, volume 1 of *Dover Children's Thrift Classics*. Dover Publications.

Agre, P. and Chapman, D. (1987). Pengi: an Implementation of a Theory of Activity. In *Aaai.Org*, pages 272–286, Seattle, Washington, USA.

Allen, P. (2018). Paul Allen Home Page.

Amazon Web Services Inc. or its affiliates. (2019). AWS Cloud9 IDE.

Amgoud, L., Cayrol, C., Lagasquie-Schiex, M. C., and Livet, P. (2008). On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23(10):1062–1093.

Amitay, E., Har'El, N., Sivan, R., and Soffer, A. (2004). Web-a-Where: Geotagging Web Content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280.

Andogah, G., Bouma, G., and Nerbonne, J. (2012). Every Document has a Geographical Scope. *Data and Knowledge Engineering*, 81-82:1–20.

Atkinson, K., Baroni, P., Giacomin, M., Hunter, A., Prakken, H., Reed, C., Simari, G., Thimm, M., and Villata, S. (2017). Towards Artificial Argumentation. In *AI Magazine 38, 3 (2017), 25–36.*

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., and Nardi, D. (2003). *The DescriptionLogic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Bal, M. and van Boheemen, C. (2009). *Narratology: Introduction to the Theory of Narrative*. University of Toronto Press.

Balai, E., Gelfond, M., and Zhang, Y. (2013). *Towards Answer Set Programming with Sorts*, pages 135–147. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bangor, A., Kortum, P., and Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of usability studies*, 4(3):114–123.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India. Morgan Kaufmann Publishers Inc.

Barnes, J. (1995). *The Cambridge Companion to Aristotle*. Cambridge Companions to Philosophy. Cambridge University Press.

Barnum, C. M. (2001). *Usability Testing and Research*. Allyn & Bacon, Inc., Needham Heights, MA, USA, 1st edition.

Baroni, P., Caminada, M., and Giacomin, M. (2011). An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, 26(4):365–410.

Bench-Capon, T. and Dunne, P. E. (2007). Argumentation in Artificial Intelligence. *Artificial Intelligence*, 171(10-15):619–641.

Bench-Capon, T. J. M. (2003). Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448.

Besnard, P. and Hunter, A. (2008). *Elements of Argumentation*, volume 47. MIT press Cambridge.

Bex, F. and Bench-Capon, T. (2014). Understanding narratives with argumentation. In *Frontiers in Artificial Intelligence and Applications*, volume 266, pages 11–18.

Bex, F. J. and Verheij, B. (2010). Story schemes for argumentation about the facts of a crime. In *2010 AAAI Fall Symposium Series*.

Bistarelli, S., Rossi, F., and Santini, F. (2016). ConArg: A Tool for Classical and Weighted Argumentation. In *COMMA*.

Bistarelli, S. and Santini, F. (2011). ConArg: A Constraint-Based Computational Framework for Argumentation Systems. In *IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011, Boca Raton, FL, USA, November 7-9, 2011*, pages 605–612.

Blackmon, M. H., Polson, P. G., Kitajima, M., and Lewis, C. (2002). Cognitive Walkthrough for the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 463–470, Minneapolis, Minnesota, USA. ACM New York, NY, USA.

Bogaard, T., Wielemaker, J., Hollink, L., and van Ossenbruggen, J. (2017). *SWISH DataLab: A Web Interface for Data Exploration and Analysis*, pages 181–187. Springer International Publishing, Cham.

Bower, G. H. (1976). Experiments on Story Understanding and Recall. *Quarterly Journal of Experimental Psychology*, 28(4):511–534.

Brabham, D. C. (2008). Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence*, 14(1):75–90.

Bragg, J., Mausam, and Weld, D. S. (2016). Optimal Testing for Crowd Workers. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 966–974, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Brandes, U., Eiglsperger, M., Lerner, J., and Pich, C. (2013). Graph Markup Language (GraphML). In Tamassia, R., editor, *Handbook of graph drawing visualization*, Discrete mathematics and its applications, pages 517–541. CRC Press, Boca Raton [u.a.].

Brewka, G. and Woltran, S. (2010). Abstract dialectical frameworks. In *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*.

Brooke, J. (1996). SUS - A Quick and Dirty Usability Scale. *Usability evaluation in industry*, 189(194):4–7.

Brooks, P. (1992). *Reading for the Plot: Design and Intention in Narrative*. Harvard University Press.

Brun, G., Dominguès, C., and Paris-est, U. (2015). TEXTOMAP : Determining Geographical Window for Texts. In *Proceedings of the 9th Workshop on Geographic Information Retrieval*, GIR '15, pages 7–8, New York, NY, USA. ACM.

Buhrmester, M., Kwang, T., and Gosling, S. D. (2011a). Amazon's Mechanical Turk. *Perspectives on Psychological Science*, 6(1):3–5.

Buhrmester, M., Kwang, T., and Gosling, S. D. (2011b). Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1):3–5.

Byrne, R. M. J. (1989). Suppressing valid inferences with conditionals. *Cognition*, 31(1):61–83.

Cabrio, E., Villata, S., and Palmero Aprosio, A. (2017). A RADAR for information reconciliation in Question Answering systems over Linked Data 1. *Semantic Web*, 8(4):601–617.

Cambria, E., Rajagopal, D., Kwok, K., and Sepulveda, J. (2015). GECKA: Game Engine for Commonsense Knowledge Acquisition. In *Proceedings of the 28th International Flairs Conference*, pages 282–287.

Caminada, M. (2006). On the Issue of Reinstatement in Argumentation. In Fisher, M., van der Hoek, W., Konev, B., and Lisitsa, A., editors, *Logics in Artificial Intelligence*, pages 111–123, Berlin, Heidelberg. Springer Berlin Heidelberg.

Carroll, N. (2001). On the Narrative Connection. *Beyond Aesthetics: Philosophical Essays*, pages 118–133.

Cayrol, C. and Lagasquie-Schiex, M. C. (2005). On the Acceptability of Arguments in Bipolar Argumentation Frameworks BT - Symbolic and Quantitative Approaches to Reasoning with Uncertainty. pages 378–389, Berlin, Heidelberg. Springer Berlin Heidelberg.

Celino, I., Contessa, S., Corubolo, M., Dell'Aglio, D., Della Valle, E., Fumeo, S., and Krüger, T. (2012). Linking Smart Cities Datasets with Human Computation – The Case of UrbanMatch. In Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J. X., Hendler, J., Schreiber, G., Bernstein, A., and Blomqvist, E., editors, *The Semantic Web – ISWC 2012*, pages 34–49, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chabierski, P., Russo, A., Law, M., and Broda, K. (2017). Machine Comprehension of Text Using Combinatory Categorial Grammar and Answer Set Programs. In *Proceedings of the 13th International Symposium on Commonsense Reasoning (COMMONSENSE 2017)*, volume 2052, London, UK. CEUR Workshop Proceedings, CEUR-WS.org.

Chamberlain, J., Bartle, R., Kruschwitz, U., Madge, C., and Poesio, M. (2017). Metrics of Games-With-A-Purpose for NLP Applications. *Proceedings of the Games4NLP: Using Games and Gamification for Natural Language Processing*.

Charniak, E. (1972). Toward a Model of Children's Story Comprehension. Technical Report AITR-266, Cambridge, MA, USA.

Charniak, E. (1977a). A Framed Painting: The Representation of a Common Sense Knowledge Fragment. *Cognitive Science*, 1(4):355–394.

Charniak, E. (1977b). Ms. Maloprop, a Language Comprehension Program. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 1–7, Cambridge, USA. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Charwat, G., Dvořák, W., Gaggl, S. A., Wallner, J. P., and Woltran, S. (2015). Methods for solving reasoning problems in abstract argumentation – A survey. *Artificial Intelligence*, 220:28–63.

Chaturvedi, S., Peng, H., and Roth, D. (2017). Story Comprehension for Predicting What Happens Next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614, Copenhagen, Denmark. Association for Computational Linguistics.

Chen, W., Quan, X., and Chen, C. (2018). Gated Convolutional Networks for Commonsense Machine Comprehension. In Cheng, L., Leung, A. C. S., and Ozawa, S., editors, *Proceedings of the 25th International Conference on Neural Information Processing (ICONIP 2018)*, pages 297–306, Siem Reap, Cambodia. Springer International Publishing.

Chittilappilly, A. I., Chen, L., and Amer-Yahia, S. (2016). A Survey of General-Purpose Crowdsourcing Techniques. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2246–2266.

Clark, P. and Harrison, P. (2009). Large-scale Extraction and Use of Knowledge From Text. In *Proceedings of the fifth international conference on Knowledge capture - K-CAP '09*, pages 153–160, Redondo Beach, California, USA.

Codeanywhere Inc. (2017). Codeanywhere IDE.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., and Players, F. (2010). Predicting Protein Structures With a Multiplayer Online Game. *Nature*, 466:756.

Corcoglioniti, F., Rospocher, M., and Palmero Aprosio, A. (2016). Frame-Based Ontology Population with PIKES. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3261–3275.

Coste-Marquis, S., Konieczny, S., Marquis, P., and Ouali, M. A. (2012). Weighted attacks in argumentation frameworks. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Cristani, M. and Tomazzoli, C. (2016). A Multimodal Approach to Relevance and Pertinence of Documents. In Fujita, H., Ali, M., Selamat, A., Sasaki, J., and Kurematsu, M., editors, *Trends in Applied Knowledge-Based Systems and Data Science: 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*, pages 157–168. Springer International Publishing, Cham.

Cullingford, R. E. (1978). *Script Application: Computer Understanding of Newspaper Stories*. PhD thesis, New Haven, CT, USA.

Dasseville, I. and Janssens, G. (2015). A web-based IDE for IDP. In *1st International Workshop on User-Oriented Logic Programming (IULP2015)*.

Davis, E. and Marcus, G. (2015). Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Communications of the ACM*, 58(9):92–103.

de Alencar, R. O. and Jr, C. A. D. (2011). *Geotagging Aided by Topic Detection with Wikipedia*, pages 461–477. Springer Berlin Heidelberg, Berlin, Heidelberg.

Denecker, M. and Ternovska, E. (2008). A Logic of Nonmonotone Inductive Definitions. *ACM Trans. Comput. Logic*, 9(2):14:1—-14:52.

D'Ettole, G., Bjørner, T., and De Götzen, A. (2020). How to Design Potential Solutions for a Cross-country Platform that Leverages Students' Diversity: A User-Centered Design Approach – and Its Challenges. In Marcus, A. and Rosenzweig, E., editors, *Design, User Experience, and Usability. Case Studies in Public and Personal Interactive Systems*, pages 415–426, Cham. Springer International Publishing.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Diakidoy, I.-A., Kakas, A., Michael, L., and Miller, R. (2014). Story Comprehension Through Argumentation. In Parsons, S., Oren, N., Reed, C., and Cerutti, F., editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, pages 31–42, Scottish Highlands, UK. IOS Press.

Diakidoy, I.-A., Kakas, A., Michael, L., and Miller, R. (2015). STAR: A System of Argumentation for Story Comprehension and Beyond. In *Working Notes of the 12th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2015)*, pages 64–70.

Diakidoy, I.-A., Michael, L., and Kakas, A. (2017). Knowledge Activation in Story Comprehension. *Journal of Cognitive Science*, 18(4).

D'Ignazio, C., Bhargava, R., Zuckerman, E., and Beck, L. (2014). CLIFF-CLAVIN: Determining Geographic Focus for News Articles. In *Proceedings of the NewsKDD: Data Science for News Publishing*.

Dignum, V. (2017). Responsible Autonomy. In *Proceedings of the Twenty -Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*, pages 4698–4704.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing Systems on the World-Wide Web. *Commun. ACM*, 54(4):86–96.

Dolan, C. P. (1989). *Tensor Manipulation Networks: Connectionist and Symbolic Approaches to Comprehension, Learning, and Planning*. PhD thesis, Los Angeles, CA, USA.

Domeshek, E., Jones, E., and Ram, A. (1999). Understanding Language Understanding. chapter Capturing, pages 73–105. MIT Press, Cambridge, MA, USA.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357.

Dunne, P. E., Hunter, A., McBurney, P., Parsons, S., and Wooldridge, M. (2011). Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486.

Dyer, M. G. (1983). *In-depth Understanding: a Computer Model of Integrated Processing for Narrative Comprehension*. MIT Press, Cambridge, MA, USA.

Eason, K. D. (2005). *Information Technology And Organisational Change*. Taylor & Francis.

Eclipse Foundation (2017). Eclipse Che IDE.

Egly, U., Gaggl, S. A., and Woltran, S. (2008). ASPARTIX: Implementing Argumentation Frameworks Using Answer-Set Programming BT - Logic Programming. pages 734–738, Berlin, Heidelberg. Springer Berlin Heidelberg.

Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., and Vrandečić, D. (2014). *Introducing Wikidata to the Linked Data Web*, volume 8796 of *Lecture Notes in Computer Science*, pages 50–65. Springer International Publishing, Cham.

Estellés-Arolas, E. and González-Ladrón-de Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200.

Estellés-Arolas, E., Navarro-Giner, R., and González-Ladrón-de Guevara, F. (2015). *Crowdsourcing Fundamentals: Definition and Typology*, pages 33–48. Springer International Publishing, Cham.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised Named-Entity Extraction From the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134.

Evans, J. S. B. T. (2002). Logic and human reasoning: An assessment of the deduction paradigm. *Psychological bulletin*, 128(6):978.

Fan, X. and Toni, F. (2014). On Computing Explanations in Abstract Argumentation. In *ECAI*.

Fellbaum, C. (2010). *Wordnet*. Springer Netherlands, Dordrecht.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., and Mueller, E. T. (2013). Watson: Beyond Jeopardy! *Artificial Intelligence*, 199-200:93–105.

Feyisetan, O., Simperl, E., Van Kleek, M., and Shadbolt, N. (2015). *Improving Paid Microtasks through Gamification and Adaptive Furtherance Incentives*, pages 333–343. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE.

Forster, E. M. (2010). *Aspects of the Novel*. RosettaBooks.

Fort, K., Guillaume, B., and Chastant, H. (2014). Creating Zombilingo, a Game With A Purpose for dependency syntax annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 2–6. ACM.

Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O., and Bader, G. D. (2016). Cytoscape.js: A Graph Theory Library for Visualisation and Analysis. *Bioinformatics*, 32(2):309–311.

Freeley, A. J. and Steinberg, D. L. (2013). *Argumentation and debate*. Cengage Learning.

Friedman, M. (1974). Explanation and Scientific Understanding. *The Journal of Philosophy*, 71(1):5–19.

Garris, R., Ahlers, R., and Driskell, J. (2002). Games, Motivation, and Learning: A Research and Practice Model. *Simulation & gaming*, 33(4):441–467.

Geiger, D. and Schader, M. (2014). Personalized task recommendation in crowdsourcing information systems - Current state of the art. *Decision Support Systems*, 65(C):3–16.

Geiger, D., Seedorf, S., Nickerson, R., and Schader, M. (2011). Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes. In *Proceedings of the Seventeenth Americas Conference on Information Systems*, number JANUARY, pages 1–11.

Genette, G., Sheridan, A., and Logan, M.-R. (1982). *Figures of Literary Discourse*. European Perspectives: A Series in Social Thought and Cultural Criticism. Columbia University Press.

Gerrig, R. J. (2005). The Scope of Memory-Based Processing. *Discourse Processes*, 39(2-3):225–242.

Goldman, S. R., Graesser, A. C., and Broek, P. V. D. (1999). *Narrative Comprehension, Causality, and Coherence: Essays in Honor of Tom Trabasso*. Taylor & Francis.

Gordon, A. S. (2016). Commonsense Interpretation of Triangle Behavior. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, Arizona.

Gordon, J. and Schubert, L. (2011). Discovering Commonsense Entailment Rules Implicit in Sentences. In ... *TextInfer 2011 Workshop on Textual Entailment*, number 2003, pages 59–63, Edinburgh, Scotland, UK.

Gordon, J. and Schubert, L. K. (2010). Quantificational Sharpening of Commonsense Knowledge. In *Proceedings of the 24th 2010 AAAI Fall Symposium Series*, Arlington, Virginia. Association for the Advancement of Artificial Intelligence (AAAI) Publications.

Haase, K. (1996). FramerD: Representing Knowledge in the Large. *IBM Syst. J*, 35(3-4):381–397.

Halterman, A. (2017). Mordecai: Full Text Geoparsing and Event Geocoding. *The Journal of Open Source Software*, 2(9).

Hamari, J. and Eranti, V. (2011). Framework for Designing and Evaluating Game Achievements. In *Proceedings of DiGRA 2011 Conference: Think Design Play*, page 20, Utrecht, Netherlands.

Hayes, P. and McBride, B. (2004). RDF Semantics. W3C Recommendation. *World Wide Web Consortium*.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Haykin, S. (2009). *Neural Networks and Learning Machines*. Number v. 10 in Neural networks and learning machines. Prentice Hall.

Hees, J., Roth-Berghofer, T., Biedert, R., Adrian, B., and Dengel, A. (2011). BetterRelations: Using a Game to Rate Linked Data Triples. In Bach, J. and Edelkamp, S., editors, *KI 2011: Advances in Artificial Intelligence*, pages 134–138, Berlin, Heidelberg. Springer Berlin Heidelberg.

Herdagdelen, A. and Baroni, M. (2010). The Concept Game: Better Commonsense Knowledge Extraction by Combining Text Mining and a Game with a Purpose. *AAAI Fall Symposium on Commonsense Knowledge, Arlington*, (2006):52–57.

Herdağdelen, A. and Baroni, M. (2012). Bootstrapping a Game With a Purpose for Commonsense Collection. *ACM Transactions on Intelligent Systems and Technology*, 3(4):1–24.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 1–13.

High, R. (2012). The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works. *IBM Corporation, Redbooks*.

Hill, F., Bordes, A., Chopra, S., and Weston, J. (2015). The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. In *International Conference on Learning Representations*, volume abs/1511.0.

Hirschman, L., Light, M., Breck, E., and Burger, J. D. (1999). Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 325–332, College Park, Maryland, USA. Association for Computational Linguistics.

Hobbs, J. R. (1991). SRI International: Description of the TACITUS System As Used for MUC-3. In *Proceedings of the 3rd Conference on Message Understanding*, MUC3 '91, pages 200–206, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hobbs, J. R. and Martin, P. (1987). Local Pragmatics. Technical report, SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. (1993a). Interpretation as Abduction. *Artificial Intelligence*, 63(1-2):69–142.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. (1993b). Interpretation as Abduction. *Artificial Intelligence*, 63(1):69–142.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-kelham, E., Melo, G. D., and Weikum, G. (2011). YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In *Proceedings of the 20th International Conference on World Wide Web*, pages 229–232.

Howe, J. (2006). Crowdsourcing: A Definition.

Huang, L., Le Bras, R., Bhagavatula, C., and Choi, Y. (2019). Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Hung, S. H., Lin, C. H., and Hong, J. S. (2010). Web Mining for Event-based Commonsense Knowledge Using Lexico-syntactic Pattern Matching and Semantic Role Labeling. *Expert Systems with Applications*, 37(1):341–347.

ICEcoder Ltd (2017). ICEcoder IDE.

Imani, M. B., Chandra, S., Ma, S., Khan, L., and Thuraisingham, B. (2017). Focus Location Extraction From Political News Reports With Bias Correction. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1956–1964.

Iwanska, L. M. and Shapiro, S. C. (2000). *Natural language processing and knowledge representation: language for knowledge and knowledge for language*. MIT Press, Cambridge, MA, USA.

Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J. L., III, H. D., and Davis, L. S. (2016). The Amazing Mysteries of the Gutter: Drawing Inferences Between Panels in Comic Book Narratives. *CoRR*, abs/1611.0.

Jakobovits, H. and Vermeir, D. (1999). Robust semantics for argumentation frameworks. *Journal of logic and computation*, 9(2):215–261.

John, B. E. and Packer, H. (1995). Learning and Using the Cognitive Walkthrough Method: A Case Study Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 429–436, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Johnson-Laird, P. N. and Steedman, M. (1978). The psychology of syllogisms. *Cognitive Psychology*, 10(1):64–99.

Johnston, A. B. and Burnett, D. C. (2012). *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*. Digital Codex LLC, USA.

Kakas, A. (2019). Informalizing Formal Logic. *Informal Logic*, 39(2):169–204.

Kakas, A. and Michael, L. (2016). Cognitive Systems: Argument and Cognition. *IEEE Intelligent Informatics Bulletin*, 17(1):14–20.

Kakas, A., Michael, L., and Toni, F. (2016). Argumentation: Reconciling Human and Automated Reasoning. *CEUR Workshop Proceedings*, 1651:43–60.

Karimzadeh, M., Pezanowski, S., MacEachren, A. M., and Wallgrün, J. O. (2019). GeoTxt: A Scalable Geoparsing System for Unstructured Text Geolocation. *Transactions in GIS*, 23(1):118–136.

Katz, B. (1997). Annotating the World Wide Web Using Natural Language. In *Computer-Assisted Information Searching on Internet*, RIAO '97, pages 136–155, Paris, France, France. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., Zarour, E., Players, P., Sarmenta, L., Blanchette, M., and Waldispühl, J. (2012). Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment. *PLOS ONE*, 7(3):1–9.

Kim, G. L., Lawley, L., and Schubert, L. (2019). Towards Natural Language Story Understanding with Rich Logical Schemas. In *Proceedings of the 6th Workshop on Natural Language and Computer Science*, pages 11–22, Gothenburg, Sweden. Association for Computational Linguistics.

Kintsch, W. (1988). The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, 95(C):163–182.

Kirkels, Yvonne E. M. and Post, G. (2013). Crowdvoting, a method tested in favour of entrepreneurship. pages 1–9, Manchester. The International Society for Professional Innovation Management (ISPIM).

Kline, R. B. and Seffah, A. (2005). Evaluation of Integrated Software Development Environments: Challenges and Results from Three Empirical Studies. *International Journal of Human-Computer Studies*, 63(6):607–627.

Kluyver, T., Ragan-kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90.

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. (2017). The NarrativeQA Reading Comprehension Challenge.

Kowalski, R. and Sadri, F. (2016). Programming in Logic Without Logic Programming. *Theory and Practice of Logic Programming*, 16(3):269–295.

Kowalski, R. and Sergot, M. (1989). *A Logic-Based Calculus of Events*, pages 23–55. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kucera, H. and Francis, W. (1979). A Standard Corpus of Present-day Edited American English, for Use With Digital Computers (Revised and Amplified from 1967 Version).

Kuo, Y.-L. and Hsu, J. Y.-J. (2011). Resource-bounded Crowd-sourcing of Commonsense Knowledge. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2470–2475. AAAI Press.

Kuo, Y.-l., Lee, J.-C., Chiang, K.-y., Wang, R., Shen, E., Chan, C.-w., and Hsu, J. Y.-j. (2009). Community-based Game Design: Experiments on Social Games for Commonsense Data Collection. In *Proceedings of the 1st ACM SIGKDD Workshop on Human Computation (HCOMP 2009)*, pages 15–22, Paris, France. Association for Computing Machinery (ACM).

Lafourcade, M., Joubert, A., and Le Brun, N. (2015). GWAPs for Natural Language Processing. *Games with a Purpose (Gwaps)*, pages 47–72.

Landa, J. A. G. and Onega, S. (2014). *Narratology: An Introduction*. Longman Critical Readers. Taylor & Francis.

Landauer, T. (1986). How Much do People Remember? Some Estimates of the Quantity of Learned Information in Long-term Memory. *Cognitive Science*, 10(4):477–493.

Lassila, O. and Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999.

Law, E. and von Ahn, L. (2011). *Human Computation*. Morgan & Claypool Publishers, 1st edition.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., and Others (2015). DBpedia–a Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Lehnert, W. G. (1977). *The Process of Question Answering*. PhD thesis, New Haven, CT, USA.

Leidner, J. L. and Lieberman, M. D. (2011). Detecting Geographical References in the Form of Place Names and Associated Spatial Natural Language. *SIGSPATIAL Special*, 3:5–11.

Lenat, D. (1995). CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38.

Lenat, D. (2019). Cyc Technology Overview White Paper.

Lenat, D. B. and Guha, R. V. (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

Lewis, J. R., Utesch, B. S., and Maher, D. E. (2015). Measuring Perceived Usability: The SUS, UMUX-LITE, and AltUsability. *International Journal of Human–Computer Interaction*, 31(8):496–505.

Li, Z., Ding, X., and Liu, T. (2019). Story Ending Prediction by Transferable BERT. In *IJCAI*.

Lieberman, H. (2008). Usable AI Requires Commonsense Knowledge. *ACM Conference on Computers and Human Interaction*, pages 1–5.

Lieberman, H., Smith, D. A., and Teeters, A. (2007). Common Consensus: A Web-Based Game for Collecting Commonsense Goals. In *Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, Honolulu, Hawaii, USA.

Lieto, A. and Radicioni, D. P. (2016). From Human to Artificial Cognition and Back: New Perspectives on Cognitively Inspired AI Systems. *Cognitive Systems Research*, 39:1–3.

Liu, C., Zhang, H., Jiang, S., and Yu, D. (2018). DEMN: Distilled-Exposition Enhanced Matching Network for Story Comprehension. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*, Hong Kong. Association for Computational Linguistics.

Lyding, V., Rodosthenous, C., Sangati, F., ul Hassan, U., Nicolas, L., König, A., Horbacauskiene, J., and Katinskaia, A. (2019). v-trel: Vocabulary Trainer for Tracing Word Relations - An Implicit Crowdsourcing Approach. In Angelova, G., Mitkov, R., Nikolova, I., and Temnikova, I., editors, *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*, pages 675–684, Varna, Bulgaria.

Ma, K., Francis, J., Lu, Q., Nyberg, E., and Oltramari, A. (2019). Towards Generalizable Neuro-Symbolic Systems for Commonsense Question Answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32, Hong Kong, China. Association for Computational Linguistics.

Macefield, R. (2009). How to Specify the Participant Group Size for Usability Studies: A Practitioner's Guide. *Journal of Usability Studies*, 5(1):34–45.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2015). YAGO3: A Knowledge Base from Multilingual Wikipedias. *Proceedings of CIDR*, pages 1–11.

Manning, C. D., Bauer, J., Finkel, J., Bethard, S. J., Surdeanu, M., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *An Introduction to Information Retrieval*, volume 1. Cambridge University Press.

Marcopoulos, E., Reotutar, C., and Zhang, Y. (2017). An Online Development Environment for Answer Set Programming. *2nd International Workshop on User-Oriented Logic Paradigms (IULP 2017)*.

Markotschi, T. and Johanna, V. (2010). GuessWhat?! – Human Intelligence for Mining Linked Data. In *Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data KIELD at the International Conference on Knowledge Engineering and Knowledge Management EKAW*, volume 1, pages 1–12.

Martin, D., Carpendale, S., Gupta, N., Hoßfeld, T., Naderi, B., Redi, J., Siahaan, E., and Wechsung, I. (2017). Understanding the Crowd: Ethical and Practical Matters in the Academic Use of Crowdsourcing. In Archambault, D., Purchase, H., and Hoßfeld, T., editors, *Evaluation in the Crowd. Crowdsourcing and Human-Centered Experiments*, pages 27–69, Cham. Springer International Publishing.

Maslan, N., Roemmele, M., and Gordon, A. S. (2015). One Hundred Challenge Problems for Logical Formalizations of Commonsense Psychology. In *Proceedings of the 12th International Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford, California, USA. AAAI Publications.

McCarthy, J. (1959). Programs With Common Sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, pages 75–91. London.

McCarthy, J. (1989). *Artificial Intelligence, Logic and Formalizing Common Sense*, pages 161–190. Springer Netherlands, Dordrecht.

McCarthy, J. (1990). An Example for Natural Language Understanding and the AI Problems it Raises.

McCharty, J., Minsky, M. L., Rochester, N., and Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence.

McCorduck, P. (2004). *Machines Who Think: A Personal Inquiry Into The History and Prospects of Artificial Intelligence*. Ak Peters Series. A.K. Peters.

McNamara, D. S. and Magliano, J. (2009). Toward a Comprehensive Model of Comprehension. *Psychology of Learning and Motivation*, 51:297–384.

Mekler, E. D., Brühlmann, F., Opwis, K., and Tuch, A. N. (2013). Disassembling Gamification: The Effects of Points and Meaning on User Motivation and Performance. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1137–1142, New York, NY, USA. Association for Computing Machinery.

Melo, F. and Martins, B. (2016). Automated Geocoding of Textual Documents: A Survey of Current Approaches. *Transactions in GIS*, 21(1):3–38.

Mercier, H. and Sperber, D. (2011). Why Do Humans Reason? Arguments for an Argumentative Theory. *Behavioral and Brain Sciences*, 34(02):57–74.

Merriam-Webster Online (2009). Merriam-Webster Online Dictionary.

Miaschi, A. and Dell'Orletta, F. (2020). Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online. Association for Computational Linguistics.

Michael, L. (2008). *Autodidactic Learning and Reasoning*. PhD thesis, Cambridge, MA, USA.

Michael, L. (2009). Reading Between the Lines. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1525–1530.

Michael, L. (2013a). Machines with Websense. In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2013)*.

Michael, L. (2013b). Story Understanding... Calculemus. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*, Ayia Napa, Cyprus.

Michael, L. (2016). Cognitive Reasoning and Learning Mechanisms. In *Proceedings of the (BICA 2016) 4th International Workshop on Artificial Intelligence and Cognition (AIC 2016)*, volume 1895, pages 2–23. CEUR-WS.org.

Michael, L. (2017). The Advice Taker 2.0. In *Proceedings of the 13th International Symposium on Commonsense Reasoning (Commonsense 2017)*, volume 2052. CEUR-WS.org.

Michael, L. (2019). Machine Coaching. In *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI @ IJCAI 2019)*, pages 80–86.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. (2018). Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

Miller, R. and Shanahan, M. (2002). *Some Alternative Formulations of the Event Calculus*, pages 452–490. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Mishra, B. D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2015). Never-Ending Learning. In *AAAI Conference on Artificial Intelligence*, pages 2302–2310.

Modgil, S. and Prakken, H. (2014). The ASPIC+ Framework for Structured Argumentation: A Tutorial. *Argument & Computation*, 5(1):31–62.

Mönch, C., Grimen, G., and Midtstraum, R. (2006). Protecting Online Games Against Cheating. In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games - NetGames '06*, page 20, Singapore.

Monteiro, B. R., Davis, C. A., and Fonseca, F. (2016). A Survey on the Geographic Scope of Textual Documents. *Computers and Geosciences*, 96:23–34.

Moravec, H. P. (2000). *Robot: Mere Machine to Transcendent Mind*. Oxford University Press on Demand.

Morgan, M. S. (2017). Narrative Ordering and Explanation. *Studies in History and Philosophy of Science Part A*, 62:86–97.

Morschheuser, B., Hamari, J., and Koivisto, J. (2016). Gamification in Crowdsourcing: A Review. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 4375–4384.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. (2016). A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Mueller, E. T. (1998). *Natural Language Processing With Thought Treasure*.

Mueller, E. T. (2000). Prospects for In-depth Story Understanding by Computer. *Cognitive Systems Research*, 23:307–340.

Mueller, E. T. (2003). Story Understanding Through Multi-representation Model Construction. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9*, HLT-NAACL-TEXTMEANING '03, pages 46–53, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mueller, E. T. (2004). Understanding Script-Based Stories Using Commonsense Reasoning. 5(4):307–340.

Mueller, E. T. (2006a). *Commonsense Reasoning*.

Mueller, E. T. (2006b). Story Understanding. In *Encyclopedia of Cognitive Science*. John Wiley & Sons, Ltd.

Mueller, E. T. (2007). Modelling Space and Time in Narratives About Restaurants. *Literary and Linguistic Computing*, 22(1):67–84.

Mueller, E. T. (2015). *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition.

Najmi, E., Malik, Z., Hashmi, K., and Rezgui, A. (2016). ConceptRDF: An RDF Presentation of ConceptNet Knowledge Base. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 145–150.

Narayanan, S. (1997). Knowledge-based Action Representations for Metaphor and Aspect (KARMA). *Computer Science Division, University of California at Berkeley dissertation*.

Nguyen, N. (2014). Microworkers Crowdsourcing Approach, Challenges and Solutions. In *Proceedings of the 3rd International ACM Workshop on Crowdsourcing for Multimedia (CrowdMM '14)*, CrowdMM '14, page 1, Orlando, Florida, USA. Association for Computing Machinery (ACM).

Nguyen, Q. V. H., Duong, C. T., Nguyen, T. T., Weidlich, M., Aberer, K., Yin, H., and Zhou, X. (2017). Argument Discovery via Crowdsourcing. *The VLDB Journal*, 26(4):511–535.

Norvig, P. (1987). A Unified Theory of Inference for Text Understanding. Technical report, Berkeley, CA, USA.

Norvig, P. (1989). Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science*, 13(4):569–620.

Ohlsson, S., Sloan, R. H., Turán, G., and Urasky, A. (2013). Verbal IQ of a Four-Year Old Achieved by an AI System. In *Proceedings of the 17th AAAI Conference on Late-Breaking Developments in the Field of Artificial Intelligence*, pages 89–91.

Orkin, J. and Roy, D. (2007). The Restaurant Game: Learning Social Behavior and Language From Thousands of Players Online. *Journal of Game Development*, 3(December):39–60.

Ostermann, S., Modi, A., Roth, M., Thater, S., and Pinkal, M. (2018a). MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan.

Ostermann, S., Roth, M., Modi, A., Thater, S., and Pinkal, M. (2018b). SemEval-2018 Task 11: Machine Comprehension Using Commonsense Knowledge. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval2018)*, pages 747–757, New Orleans, Louisiana. Association for Computational Linguistics.

Otani, N., Kawahara, D., Kurohashi, S., Kaji, N., and Sassano, M. (2016). Large-Scale Acquisition of Commonsense Knowledge via a Quiz Game on a Dialogue System. In *Proceedings of the Open Knowledge Base and Question Answering (OKBQA) Workshop*, pages 11–20. The COLING 2016 Organizing Committee.

Pansanato, L., Rivolli, A., and Pereira, D. (2015). An Evaluation with Web Developers of Capturing User Interaction with Rich Internet Applications for Usability Evaluation. *International Journal of Computer Science and Application*, 4(2):51–60.

Paulheim, H. (2018). How much is a Triple? Estimating the Cost of Knowledge Graph Creation. In *International Semantic Web Conference*.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L., and Ducceschi, L. (2013). Phrase Detectives: Utilizing Collective Intelligence for Internet-scale Language Resource Creation. *ACM Trans. Interact. Intell. Syst.*, 3(1):3:1—-3:44.

Prince, G. (1982). *Narratology: The Form and Functioning of Narrative*. Janua Linguarum. Series Maior. De Gruyter Mouton.

Prince, G. (2003). *A Dictionary of Narratology*. U of Nebraska Press.

Purves, R. S., Clough, P., Jones, C. B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A. K., Vaid, S., and Yang, B. (2007). The Design and Implementation of SPIRIT: A Spatially Aware Search Engine for Information Retrieval on the Internet. *International Journal of Geographical Information Science*, 21(7):717–745.

Quercini, G., Samet, H., Sankaranarayanan, J., and Lieberman, M. D. (2010). Determining the Spatial Reader Scopes of News Sources Using Local Lexicons. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, pages 43–52.

Quilitz, B. and Leser, U. (2008). *Querying Distributed RDF Data Sources with SPARQL*, pages 524–538. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rahwan, I. and Simari, G. R. (2009). *Argumentation in Artificial Intelligence*. Springer Publishing Company, Incorporated, 1st edition.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. (ii).

Ram, A. and Cox, M. (1994). Introspective Reasoning Using Meta-Explanations for Multistrategy Learning. *Machine Learning: A Multistrategy Approach*, 4:349.

Read, S. (2016). Aristotle ' s Theory of the Assertoric Syllogism. (2009):1–23.

Reddy, S., Chen, D., and Manning, C. D. (2019). CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Reiter, R. (1991). The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression. In Lifschitz, V., editor, *Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy*, chapter The Frame, pages 359–380. Academic Press Professional, Inc., San Diego, CA, USA.

Richardson, M., Burges, C. J. C., and Renshaw, E. (2013). MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. *Empirical Methods in Natural Language Processing (EMNLP)*, (October):193–203.

Rick Altman (2008). *A Theory of Narrative*. Columbia University Press.

Ricoeur, P. (1980). Narrative Time. *Critical Inquiry*, 7(1):169–190.

Riloff, E. and Thelen, M. (2000). A Rule-based Question Answering System for Reading Comprehension Tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests As Evaluation for Computer-based Language Understanding Sytems - Volume 6*, ANLP/NAACL-ReadingComp '00, pages 13–19, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rodosthenous, C., Lyding, V., Sangati, F., König, A., ul Hassan, U., Nicolas, L., Horbacauskiene, J., Katinskaia, A., and Aparaschivei, L. (2020). Using Crowdsourced Exercises for Vocabulary Training to Expand ConceptNet. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 307–316, Marseille, France. European Language Resources Association.

Rodosthenous, C. and Michael, L. (2016). A Hybrid Approach to Commonsense Knowledge Acquisition. In Pearce, D. and Pinto, S. H., editors, *Proceedings of the 8th European Starting AI Researcher Symposium*, pages 111–122. IOS Press.

Rodosthenous, C. and Michael, L. (2019). Using Generic Ontologies to Infer the Geographic Focus of Text. In van den Herik, J. and Rocha, A. P., editors, *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018)*, pages 223–246, Funchal, Madeira, Portugal. Springer International Publishing.

Rodosthenous, C. and Michael, L. (2021). A Crowdsourcing Methodology for Improved Geographic Focus Identification of News-stories. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, pages 680–687. INSTICC, SciTePress.

Rodosthenous, C. T. and Michael, L. (2014). Gathering Background Knowledge for Story Understanding through Crowdsourcing. In *Proceedings of the 5th Workshop on Computational Models of Narrative (CMN 2014)*, volume 41, pages 154–163, Quebec, Canada. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Rodosthenous, C. T. and Michael, L. (2018a). A Platform for Commonsense Knowledge Acquisition Using Crowdsourcing. In Zdravkova, K., Fort, K., and Bédi, B., editors, *Supplementary Proceedings of the enetCollect WG3 & WG5 Meeting 2018*, pages 24–25, Leiden. CEUR.

Rodosthenous, C. T. and Michael, L. (2018b). GeoMantis: Inferring the Geographic Focus of Text using Knowledge Bases. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, pages 111–121. INSTICC, SciTePress.

Rodosthenous, C. T. and Michael, L. (2018c). Web-STAR: A Visual Web-based IDE for a Story Comprehension System. *Theory and Practice of Logic Programming*, pages 1–43.

Roth, P. A. (1989). How Narratives Explain. *Social Research*, 56(2):449–478.

Sabou, M., Scharl, A., and Föls, M. (2013). Crowdsourced Knowledge Acquisition: Towards Hybrid-Genre Workflows. *Int. J. Semant. Web Inf. Syst.*, 9(3):14–41.

Safranski, K. (2017). Codiad Web Based, Cloud IDE.

Saldanha, E.-A. D. and Kakas, A. (2019). Cognitive Argumentation for Human Syllogistic Reasoning. *KI - Künstliche Intelligenz*, 33(3):229–242.

Sandhaus, E. (2008). The New York Times Annotated Corpus LDC2008T19. DVD. *Linguistic Data Consortium, Philadelphia*.

Sap, M., Bras, R. L., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., and Choi, Y. (2019). ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *ArXiv*, abs/1811.0.

Schank, R. and Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding: An Enquiry Into Human Knowledge Structures*. Artificial Intelligence Series. Psychology Press.

Schank, R. C. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology*, 3(4):552–631.

Schank, R. C. (1973). *Causality and Reasoning*. Istituto per gli Studi Semantici e Cognitivi Castagnola: Working papers. Istituto per gli studi semantici e cognitivi.

Schank, R. C. and Abelson, R. P. (1975). Scripts, Plans, and Knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 151–157, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Schank, R. C., Goldman, N., Riager, C. J., and Rissbeck, C. (1973). Margie Memory, Analysis, Response Generation, and Inference on English. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI'73, pages 255–261, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Schank, R. C. and Riesbeck, C. K. (1981). *Inside Computer Understanding: Five Programs Plus Miniatures*. Artificial Intelligence Series. Taylor & Francis.

Scharl, A., Sabou, M., and Föls, M. (2012). Climate Quiz: A Web Application for Eliciting and Validating Knowledge from Social Networks. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web*, WebMedia '12, pages 189–192, New York, USA. Association for Computing Machinery (ACM).

Schmidt, G. B. and Jettinghoff, W. M. (2016). Using Amazon Mechanical Turk and Other Compensated Crowdsourcing Sites. *Business Horizons*, 59(4):391–400.

Schubert, L. (2002). Can we Derive General World Knowledge From Texts? In *Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 94–97, San Diego, California. Morgan Kaufmann Publishers Inc.

Schubert, L. and Hwang, C. H. (1989). An Episodic Knowledge Representation for Narrative Texts. Technical Report 345.

Schulz, C. and Dumitrache, D. (2016). The ArgTeach Web-Platform. In *COMMA*.

Shapiro, S. C. and Rapaport, W. J. (1995). An Introduction to a Computational Reader of Narratives. *Deixis in narrative: A cognitive science perspective*, pages 79–105.

Sharma, A. and Forbus, K. D. (2010). Graph-based Reasoning and Reinforcement Learning for Improving Q/A Performance in Large Knowledge-Based Systems. In *Proceedings of the 2010 AAAI Fall Symposium Series*, pages 96–101, Arlington, Virginia.

Siebert, S. and Stolzenburg, F. (2019). CoRg: Commonsense Reasoning Using a Theorem Prover and Machine Learning. In Benzm\"uller, C., Parent, X., and Steen, A., editors, *Selected Student Contributions and Workshop Papers of LuxLogAI 2018*, volume 10 of *Kalpa Publications in Computing*, pages 20–26, Luxembourg. EasyChair.

Silva, M. J., Martins, B., Chaves, M., Afonso, A. P., and Cardoso, N. (2006). Adding Geographic Scopes to Web Resources. *Computers, Environment and Urban Systems*, 30(4):378–399.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the Game of Go With Deep Neural Networks and Tree Search. *Nature*, 529(7587):484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550:354.

Singh, P. (2002). The Public Acquisition of Commonsense Knowledge. Technical report, Stanford, California.

Singh, P., Barry, B., and Liu, H. (2004). Teaching Machines About Everyday Life. *BT Technology Journal*, 22(July):227–240.

Siorpaes, K. and Hepp, M. (2008a). Games with a Purpose for the Semantic Web. *IEEE Intelligent Systems*, 23(3):50–60.

Siorpaes, K. and Hepp, M. (2008b). OntoGame: Weaving the Semantic Web by Online Games. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *The Semantic Web: Research and Applications*, pages 751–766, Berlin, Heidelberg. Springer Berlin Heidelberg.

Smullyan, R. M. (1968). *First-order Logic*. Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer-Verlag.

Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, San Francisco, California.

Speer, R. and Havasi, C. (2013). *ConceptNet 5: A Large Semantic Network for Relational Knowledge*, pages 161–176. Springer Berlin Heidelberg, Berlin, Heidelberg.

Storks, S., Gao, Q., and Chai, J. Y. (2019). Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches. *arXiv preprint arXiv:1904.01172*.

Storring, G. (1908). *Experimentelle Untersuchungen über einfache Schlussprozesse*, volume 11. W. Engelmann.

Strass, H., Wyner, A., and Diller, M. (2019). EMIL: Extracting Meaning from Inconsistent Language: Towards Argumentation Using a Controlled Natural Language Interface. *International Journal of Approximate Reasoning*, 112:55–84.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). YAGO: A Large Ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-to-End Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2440–2448, Cambridge, MA, USA. MIT Press.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2019). CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Teitler, B. E., Lieberman, M. D., Panozzo, D., Sankaranarayanan, J., Samet, H., and Sperling, J. (2008). NewsStand: A New View on News. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–18.

Thaler, K., Simperl, E., Siorpaes, K., and Aifb, S. (2011). SpotTheLink: A Game for Ontology Alignment. *Proc. 6th Conference for Professional Knowledge Management WM*, 6:246–253.

Tindale, C. W. (2007). *Fallacies and argument appraisal*. Cambridge University Press.

Toni, F. (2014). A Tutorial on Assumption-based Argumentation. *Argument & Computation*, 5(1):89–117.

Toulmin, S. E. (2003). *The uses of argument*. Cambridge university press.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, 59(236):433–460.

Tversky, B. (1993). Cognitive Maps, Cognitive Collages, and Spatial Mental Models. In Frank, A. U. and Campari, I., editors, *Spatial Information Theory A Theoretical Basis for GIS: European Conference, COSIT'93 Marciana Marina, Elba Island, Italy September 19–22, 1993 Proceedings*, pages 14–24. Springer Berlin Heidelberg, Berlin, Heidelberg.

UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., and Pustejovsky, J. (2013). SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. *Second joint conference on lexical and computational semantics (* SEM)*, 2(SemEval):1–9.

Van Eemeren, F. H., Jackson, S., and Jacobs, S. (2015). *Argumentation*, pages 3–25. Springer International Publishing, Cham.

Van Pelt, C. and Sorokin, A. (2012). Designing a Scalable Crowdsourcing Platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 765–766, New York, NY, USA. ACM.

Velleman, J. D. (2003). Narrative Explanation. *The Philosophical Review*, 112(1):1–25.

Verheij, B. (1996). Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. In *In Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop*, pages 357–368. Universiteit.

Vickrey, D., Bronzan, A., Choi, W., Kumar, A., Turner-Maier, J., Wang, A., and Koller, D. (2008). Online Word Games for Semantic Data Collection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 533–542, Stroudsburg, PA, USA. Association for Computational Linguistics.

von Ahn, L. (2013). Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 1–2. ACM.

von Ahn, L., Blum, M., Hopper, N. J., and Langford, J. (2003). CAPTCHA: Using Hard AI Problems for Security. In Biham, E., editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 294–311, Berlin, Heidelberg. Springer Berlin Heidelberg.

von Ahn, L. and Dabbish, L. (2004). Labeling Images with a Computer Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA. ACM.

von Ahn, L. and Dabbish, L. (2008). Designing Games With a Purpose. *Communications of the ACM*, 51(8):57.

von Ahn, L., Kedia, M., and Blum, M. (2006a). Verbosity: A Game for Collecting Common-Sense Facts. In *Proceedings of the 25th SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*, page 75, Montréal, Québec. Association for Computing Machinery (ACM).

von Ahn, L., Liu, R., and Blum, M. (2006b). Peekaboom: A Game for Locating Objects in Images. In *Proceedings of the 24th SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*, pages 55–64, Montréal, Québec, Canada. Association for Computing Machinery (ACM).

Waitelonis, J., Ludwig, N., Knuth, M., and Sack, H. (2011). WhoKnows? Evaluating linked data heuristics with a quiz that cleans up DBpedia. *Interactive Technology and Smart Education*, 8(4):236–248.

Wang, A., Hoang, C. D. V., and Kan, M. Y. (2013). Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation*, 47(1):9–31.

Wang, Z., Wang, H., Wen, J.-R., and Xiao, Y. (2015). An Inference Approach to Basic Level of Categorization. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 653–662, Melbourne, Australia. Association for Computing Machinery (ACM).

Wason, P. C. (1968). Reasoning about a Rule. *Quarterly Journal of Experimental Psychology*, 20(3):273–281.

Wason, P. C. and Johnson-Laird, P. N. (1972). *Psychology of Reasoning: Structure and Content*. A Harvard Paperback. Harvard University Press.

Watanabe, K. (2018). Newsmap. *Digital Journalism*, 6(3):294–309.

Wellner, B., Ferro, L., Grieff, W., and Hirschman, L. (2006). Reading Comprehension Tests for Computer-based Understanding Evaluation. *Natural Language Engineering*, 12(4):305–334.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., Van Merriënboer, B., Joulin, A., and Mikolov, T. (2016). Towards AI-complete question answering: A set of prerequisite toy tasks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

Weston, J., Chopra, S., and Bordes, A. (2015). Memory networks. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. Technical report, New York, NY, USA.

Wielemaker, J., Lager, T., and Riguzzi, F. (2015). SWISH: SWI-Prolog for Sharing. In *1st International Workshop on User-Oriented Logic Programming (IULP2015)*.

Wielemaker, J., Riguzzi, F., Kowalski, R. A., Lager, T., Sadri, F., and Calejo, M. (2019). Using SWISH to realize interactive web-based tutorials for logic-based languages. *Theory and Practice of Logic Programming*, 19(2):229–261.

Wielemaker, J., Schrijvers, T., Triska, M., and Lager, T. (2012). SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96.

Wilensky, R. (1976). Using Plans to Understand Natural Language. In *Proceedings of the 1976 Annual Conference*, ACM '76, pages 46–50, New York, NY, USA. ACM.

Wilensky, R. (1977). PAM: A Program That Infers Intentions. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, page 15, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wilensky, R. (1978). *Understanding Goal-based Stories*. PhD thesis, New Haven, CT, USA.

Winston, P. H. (2011). The Strong Story Hypothesis and the Directed Perception Hypothesis. In Langley, P., editor, *Technical Report FS-11-01, Papers from the AAAI Fall Symposium*, pages 345–352, Menlo Park, CA. AAAI Press.

Winston, P. H. (2012a). The Next 50 Years: a Personal View. *Biologically Inspired Cognitive Architectures*, 1.

Winston, P. H. (2012b). The Right Way. *Advances in Cognitive Systems*, 1:23–36.

Winston, P. H. (2014). The Genesis Story Understanding and Story Telling System: A 21st Century Step Toward Artificial Intelligence. Memo 019, Center for Brains Minds and Machines, MIT.

Winston, P. H. (2015). Model-based Story Summary. In Finlayson, M. A., Miller, B., Lieto, A., and Ronfard, R., editors, *6th Workshop on Computational Models of Narrative (CMN 2015)*, volume 45 of *OpenAccess Series in Informatics (OASIcs)*, pages 157–165, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Witbrock, M. J., Matuszek, C., Brusseau, A., Kahlert, R. C., Fraser, C. B., and Lenat, D. B. (2005). Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc. Technical report.

Wolf, L., Knuth, M., Osterhoff, J., and Sack, H. (2011). RISQ! Renowned Individuals Semantic Quiz: A Jeopardy Like Quiz Game for Ranking Facts. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 71–78, New York, NY, USA. ACM.

Woodruff, A. G. and Plaunt, C. (1994). GIPSY: Georeferenced Information Processing SYstem. *Journal of the American Society for Information Science*, 45:645–655.

Wu, W., Li, H., Wang, H., and Zhu, K. Q. (2012). Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. Association for Computing Machinery (ACM).

Yu, J. (2016). Geotagging Named Entities in News and Online Documents. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1321–1330.

Zang, L. J., Cao, C., Cao, Y. N., Wu, Y. M., and Cao, C. G. (2013). A Survey of Commonsense Knowledge Acquisition. *Journal of Computer Science and Technology*, 28(4):689–719.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2018a). From Recognition to Cognition: Visual Commonsense Reasoning. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, Salt Lake City.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018b). SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium. Association for Computational Linguistics.

Zubizarreta, Á., de la Fuente, P., Cantera, J. M., Arias, M., Cabrero, J., García, G., Llamas, C., and Vegas, J. (2009). Extracting Geographic Context from the Web: GeoReferencing in MyMoSe. In *Advances in Information Retrieval*, pages 554–561.

Appendix A

Cognitive Walkthrough Tasks

Introduction

Thank you participating in this research. The Web-STAR IDE is a web-based IDE that facilitates the use of the STAR system for automated story comprehension. It provides an interface to represent stories and the world knowledge required to comprehend them within the STAR system. Web-STAR also provides a Public Stories Repository for sharing publicly a user's STAR stories and opening a discussion about them.

In short, the IDE takes as input:

1) a story with questions in either Natural Language or in symbolic format (STAR syntax)
2) world knowledge in the form of rules in either graphical format or in symbolic format

and responds with the comprehension model, i.e., the way the story and its concepts are shaped through time, what holds and what does not at each time-point and answers to the questions posed.

<u>Acceptance statement</u>

By proceeding with the following tasks, you agree that we will capture data (screen capture, recording of your actions, answers to questions) for research purposes only and more specifically for the evaluation of the Web-STAR IDE.

TASK No.: **1**
Title: Create an account

Description:

Create a new account to the Web-STAR IDE. Activate the new account and log in to the system.

Goals:

1) Create an account

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Create a new account by filling in your details
3) Activate your account
4) Log in to the Web-STAR IDE
5) Log out

TASK No.: **2**
Title: Follow the guided tour

Description:

Follow the guided tour to learn the basic functionality of the system.

Goals:

1) Learn the various areas of the IDE, its main features and the options available

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Start the guided tour and go through it
4) Log out

TASK No.: **3a**
Title: Write a new story in natural language

Description:

Write a new story in natural language and add questions. Convert the story to STAR syntax. Add the background knowledge using the visual editor. Save the story.

Goals:

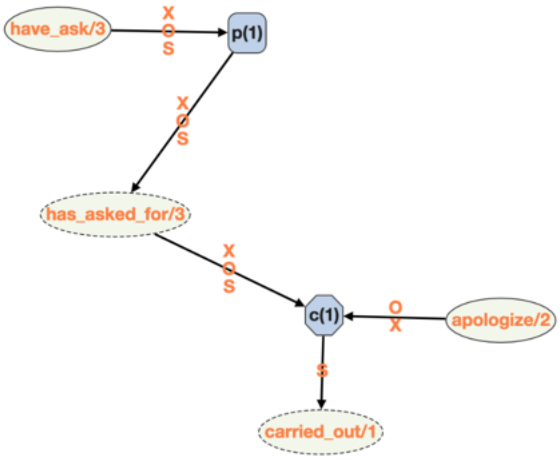
1) Understand where the different parts of the story should be placed in the IDE
2) Understand the structure of a story
3) Test the conversion process from natural language to STAR syntax (symbolic format)
4) Test the visual editor functionality to add background knowledge

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Write the following story in natural language:

Story in Natural Language
Bob called Mary. She did not want to answer the phone. Bob had asked her for a favor. She had agreed to do the favor. She answered the phone. She apologized to Bob. Was Mary embarrassed? Was the favor carried out?

4) Convert the story to STAR syntax
5) Add the background knowledge for the story using the visual editor:



Fluents: carried_out, has_asked_for

Actions: have_ask, apologize

6) Convert the background knowledge from visual format to STAR syntax
7) Save the story as “cw_task3a_nl”
8) Log out

TASK No.: **3b**
Title: Write a new story in STAR syntax

Description:

Write a new story in STAR syntax, add questions and save the story. Moreover, add the relevant background knowledge for comprehending the story in STAR syntax.

Goals:

1) Understand where the different parts of the story should be placed in the IDE
2) Understand the structure of a story
3) Test the source code editor functionality to add background knowledge

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Write the following story in STAR syntax (symbolic format):

Story in Symbolic Format

```
session(s(0),[],all).
session(s(1),[q(1),q(2)],all).

s(0) :: is_favor(favor1) at always.
s(0) :: is_person(bob) at always.
s(0) :: is_person(mary) at always.
s(0) :: is_phone(phone1) at always.

s(1) :: call(bob,mary) at 3.
s(1) :: -do_want(mary,answer(phone1)) at 4.
s(1) :: have_ask(bob,mary,favor1) at 1.
s(1) :: have_agreed(mary,do(favor1)) at 2.
s(1) :: answer(mary,phone1) at 5.
s(1) :: apologize(mary,bob) at 6.

q(1) ?? is_embarrassed(mary) at 7.
 q(2) ?? carried_out(favor1) at 8.
```

4) Write the background knowledge needed to comprehend the story above in STAR syntax:

Background knowledge in STAR syntax (symbolic format)

```
fluents([

  do_want(_,_),

  is_embarrassed(_),

  carried_out(_),

  has_asked_for(_,_,_),

  has_agreed_to(_,_)

]).


p(01) :: have_ask(X,O,S) implies has_asked_for(X,O,S).

p(02) :: have_agreed(O,do(S)) implies has_agreed_to(O,S).

c(01) :: has_asked_for(X,O,S), has_agreed_to(O,S), apologize(O,X) causes -carried_out(S).

p(03) ::  has_asked_for(X,O,S), -carried_out(S) implies is_embarrassed(O).

c(02) :: has_asked_for(X,O,S), has_agreed_to(O,S), -carried_out(S), call(X,O), is_phone(P) causes -do_want(O,answer(P)).
```

5) Convert the background knowledge in visual format
6) Save the story as “cw_task3b_star”
7) Log out

TASK No.: **4**
Title: Load a story and initiate the comprehension process

Description:

Choose a demo story, load it and initiate the comprehension process.

Goals:

1) Understand where you can find stories created by others

2) Load a story

3) Initiate the story comprehension process

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the story titled “**Penguins**” from the demo stories
4) Initiate the story comprehension process (Start Reading)
5) Examine the comprehension model in the output area and find the answer the system gave to the multiple choice question posed in the story at session 2

   What answer the system gave to the multiple choice question posed in the story at session 2:

   a) accepted choice: [penguin at 9], accepted choice: [bird at 9], rejected choice: [flying at 9]
   b) rejected choice: [penguin at 9], accepted choice: [bird at 9], accepted choice: [flying at 9]
6) Log out

TASK No.: **5a**
Title: Modify the background knowledge using the visual format editor

Description:

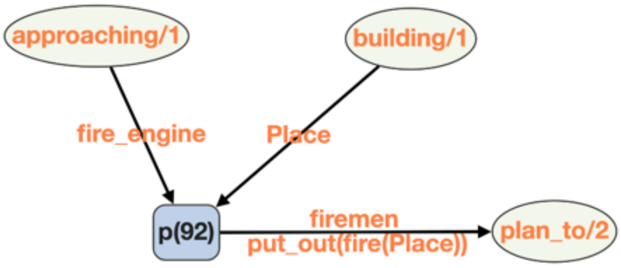
Load a story, add a new background knowledge rule, update an existing one and remove an existing rule. Initiate the story comprehension process.

Goals:

1) Understand how you can add a rule to the background knowledge
2) Understand how you can delete a rule from the background knowledge
3) Understand how you can edit a rule in the background knowledge

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the story titled "**The house**" from the public story repository
4) Add the rule using the Background Knowledge in Visual Format editor:



5) Delete the rule **p(8)** using the Background Knowledge in Visual Format editor
6) In rule **p(92)** change the argument name from **Place** to **Special_place** using the Background Knowledge in Visual Format editor
7) Convert the Background Knowledge in STAR syntax
8) Initiate the story comprehension process (Start Reading)
9) Examine the comprehension model and find the answer the system gave to the multiple choice question posed in the story at session 2

   What answer the system gave to the multiple choice question posed in the story at session 2:

   a. rejected choice: [on_fire(the_house) at 1]
   b. accepted choice: [on_fire(the_house) at 1]
10) Log out

TASK No.: **5b**
Title: Modify the background knowledge using the source code editor

Description:

Load a story, add a new background knowledge rule, update an existing one and remove an existing rule. Initiate the story comprehension process.

Goals:

1) Understand how you can add a rule to the background knowledge
2) Understand how you can delete a rule from the background knowledge
3) Understand how you can edit a rule in the background knowledge

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the story titled "**The house**" from the public story repository
4) Add the rule using the background knowledge in STAR syntax source code editor:

   p(11) :: approaching(fire_engine), building(Place) implies plan_to(firemen, put_out(fire(Place))).

5) Delete the rule **p(8)** using the background knowledge in STAR syntax source code editor
6) In rule **p(11)** change the argument name from **Place** to **Special_place** using the background knowledge in STAR syntax source code editor
7) Initiate the story comprehension process (Start Reading)
8) Examine the comprehension model and find the answer the system gave to the multiple choice question posed in the story at session 1

   What answer the system gave to the multiple choice question posed in the story at session 1:

   a) accepted choice: [on_fire(the_house) at 1]
   b) rejected choice: [on_fire(the_house) at 1]
9) Log out

TASK No.: **6**
Title: Filter the output of the comprehension process

Description:

Load a story, initiate the story comprehension process and filter the output to present only the concepts that have changes while the story unfolds.

Goals:

1) Understand how to filter the comprehension model
2) Extract information from the comprehension model

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the story titled "**The Cat**" from the public story repository
4) Initiate the story comprehension process (Start Reading)
5) Examine the comprehension model
6) Apply a filter to the visual output of the story comprehension process to show only the concepts that have changes
7) Log out

TASK No.: **7**
Title: Share a story

Description:

Load a story and share it in the public repository.

Goals:

1) Share a story with the community

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the "**Penguins**" demo story from the story browser window
4) Share the story with the community
5) Log out

TASK No.: 8
Title: Comment on a user's story

Description:

Users must find a story in the public story repository and add a comment on that story.

Goals:

1) Understand the ability to comment on a story and start a discussion

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load the "**babl project (Demo 3)**" story from the public repository
4) Add a short comment on the story to start or continue the discussion
5) Log out

TASK No.: **9**
Title: Initiate the collaboration tool

Description:

Initiate the collaboration functionality and send the link to another person. Use the feedback option to send the link to the developers of the IDE.

Goals:

1) Use the collaboration tool to work with another user

Steps:

1) Navigate your browser to http://cognition.ouc.ac.cy/webstar
2) Log in to the Web-STAR IDE
3) Load any story
4) Initiate the collaboration tool
5) Send the link through the feedback form of the system
6) Log out

Appendix B

Web-STAR IDE Evaluation Questionnaire

Web-STAR evaluation Questionnaire

Introduction

Thank you participating in this research. The Web-STAR IDE is a web-based IDE that facilitates the use of the STAR system for automated story comprehension. It provides an interface to represent stories and the world knowledge required to comprehend them within the STAR system. Web-STAR also provides a Public Stories Repository for sharing publicly a user's STAR stories and opening a discussion about them.

In short, the IDE takes as input:

1. a story with questions in either Natural Language or in symbolic format (STAR syntax)
2. world knowledge in the form of rules in either graphical format or in symbolic format

and responds with the comprehension model, i.e., the way the story and its concepts are shaped through time, what holds and what does not at each timepoint and answers to the questions posed.

Acceptance statement

By proceeding with the following tasks, you agree that we will capture data (screen capture, recording of your actions, answers to questions) for research purposes only and more specifically for the evaluation of the Web-STAR IDE.

There are 41 questions in this survey

General (demographics)

[]Please type your experiment ID *

Please write your answer here:

[]Please select your gender *

Please choose **only one** of the following:

○ Female

○ Male

[]Please select your age group *

Choose one of the following answers

Please choose **only one** of the following:

○ 18-24 years old

○ 25-34 years old

○ 35-44 years old

○ 45-54 years old

○ 55-64 years old

○ 65-74 years old

○ 75 years or older

[]What is the highest degree or level of school you have completed? *

Choose one of the following answers

Please choose **only one** of the following:

○ Less than a high school diploma
○ High school degree or equivalent (e.g. GED)
○ Some college, no degree
○ Associate degree (e.g. AA, AS)
○ Bachelor’s degree (e.g. BA, BS)
○ Master’s degree (e.g. MA, MS, MEd)
○ Professional degree (e.g. MD, DDS, DVM)
○ Doctorate (e.g. PhD, EdD)
○

If you’re currently enrolled in school, please indicate the highest degree you have *received*

[]What is your current employment status? *

Choose one of the following answers

Please choose **only one** of the following:

○ Employed full time (40 or more hours per week)
○ Employed part time (up to 39 hours per week)
○ Unemployed and currently looking for work
○ Unemployed and not currently looking for work
○ Student
○ Retired
○ Homemaker
○ Self-employed
○ Unable to work
○

[]Your degree is relevant to: *

Check all that apply

Please choose **all** that apply:

□ Computer Science
□ Psychology
□ Philosophy
□ Storytelling or Narratology
□ Linguistics

□ Law

□ Other: ____________

General (development specific)

[]Have you ever used an Integrated Development Environment (IDE) before for software development or programming? *

Please choose **only one** of the following:

○ Yes

○ No

[]Please specify what applies for each of the following IDEs: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '7 [B01]' (Have you ever used an Integrated Development Environment (IDE) before for software development or programming?)

Please choose the appropriate response for each item:

	1: I have never heard about it	2: I have heard about it but I have never used it	3: I have heard about it	4: I have used it before	5: It is on of the IDEs I mostly use
Microsoft Visual Studio	○	○	○	○	○
NetBeans	○	○	○	○	○
Eclipse	○	○	○	○	○
Cloud9	○	○	○	○	○
Codiad	○	○	○	○	○
ICEcoder	○	○	○	○	○
Codeanywhere	○	○	○	○	○
Eclipse Che	○	○	○	○	○

[]Please specify what applies for each of the following programming languages: *

Please choose the appropriate response for each item:

	1: I have never heard about it	2: I have heard about it but I have never used it	3: I have heard about it	4: I have used it before	5: It is one of the programming languages I use frequently
c	○	○	○	○	○
c++	○	○	○	○	○
JAVA	○	○	○	○	○
javascript	○	○	○	○	○
PHP	○	○	○	○	○
Python	○	○	○	○	○
Perl	○	○	○	○	○
Prolog	○	○	○	○	○
Lisp	○	○	○	○	○

[]Are you familiar with the notion of automated story understanding by machines? *

Please choose **only one** of the following:

○ Yes

○ No

[]Have you ever used a story understanding system? *

Please choose **only one** of the following:

○ Yes

○ No

[]Have you ever used the STAR story understanding system? *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '11 [B05]' (Have you ever used a story understanding system?)

Please choose **only one** of the following:

○ Yes

○ No

Task 1

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The process of creating a new account is easy.	○	○	○	○	○	○
The process of creating a new account is the same as with the other systems I use.	○	○	○	○	○	○
The process of activating the new account is easy.	○	○	○	○	○	○
The process of activating the new account is the same as with the other systems I use.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

Task 2

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find and start the guided tour.	○	○	○	○	○	○
The duration of the guided tour is appropriate for learning the basics of the IDE.	○	○	○	○	○	○
After completing the guided tour, I feel confident in using the IDE.	○	○	○	○	○	○

Task 3a

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to write the story in natural language.	○	○	○	○	○	○
The automatic conversion of the story from natural language to STAR syntax is easy .	○	○	○	○	○	○
It is easy to add the background knowledge of the story using the visual editor.	○	○	○	○	○	○
The automatic conversion of the background knowledge in visual format to STAR syntax is easy.	○	○	○	○	○	○
It is easy to save the story.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '16 [TSK03a2]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 3b

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to write the story in STAR syntax.	○	○	○	○	○	○
It is more efficient to write the story in natural language and then convert it to STAR syntax than writing the story directly using the STAR syntax.	○	○	○	○	○	○
It is easy to add the background knowledge in the source code editor	○	○	○	○	○	○
It is easy to convert the background knowledge from STAR syntax to visual format.	○	○	○	○	○	○
It is easier to understand the background knowledge rules in visual format than in STAR syntax.	○	○	○	○	○	○
It is easy to save the story.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or

disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '19 [TSK03b2]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 4

[]
What answer the system gave to the multiple choice question posed in the story at session 2?

*

Choose one of the following answers

Please choose **only one** of the following:

○ accepted choice: [penguin at 9], accepted choice: [bird at 9], rejected choice: [flying at 9]

○ rejected choice: [penguin at 9], accepted choice: [bird at 9], accepted choice: [flying at 9]

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find a story and load it.	○	○	○	○	○	○
The Load Story window is easy to use.	○	○	○	○	○	○
It is easy to find how to initiate the story comprehension process.	○	○	○	○	○	○
The system provides constant feedback on the comprehension process status.	○	○	○	○	○	○
It is easy to find the answer to the question using the visual output panel (left panel).	○	○	○	○	○	○
It is easy to find the answer to the question using the raw output panel (right panel).	○	○	○	○	○	○

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The visual output panel presents the story concepts and questions in an understandable way.	○	○	○	○	○	○
The raw output panel presents the various story concepts and questions in an understandable way.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '23 [TSK042]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 5a

[]
What answer the system gave to the multiple choice question posed in the story at session 2?
*

Choose one of the following answers

Please choose **only one** of the following:

○ rejected choice: [on_fire(the_house) at 1]

○ accepted choice: [on_fire(the_house) at 1]

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to add a rule using the Background Knowledge in Visual Format editor.	○	○	○	○	○	○
It is easy to delete a rule using the Background Knowledge in Visual Format editor.	○	○	○	○	○	○
It is easy to edit a rule using the Background Knowledge in Visual Format editor.	○	○	○	○	○	○
The controls available in the Background Knowledge in Visual Format editor are easy to use.	○	○	○	○	○	○

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to understand what is the functionality of the controls in the Background Knowledge in Visual Format editor toolbar.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '27 [TSK05a2]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 5b

[]
What answer the system gave to the multiple choice question posed in the story at session 1?
*

Choose one of the following answers

Please choose **only one** of the following:

○ accepted choice: [on_fire(the_house) at 1]

○ rejected choice: [on_fire(the_house) at 1]

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to add a rule using the Background Knowledge source code editor.	○	○	○	○	○	○
It is easy to delete a rule using the Background Knowledge source code editor.	○	○	○	○	○	○
It is easy to edit a rule using the Background Knowledge source code editor.	○	○	○	○	○	○
The controls available in the Background Knowledge source code editor toolbar are easy to use.	○	○	○	○	○	○
It is easy to understand what is the functionality of the controls in the Background Knowledge source code editor.	○	○	○	○	○	○

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is more efficient to use the Background Knowledge in Visual Format editor to modify the background knowledge than the Background Knowledge source code editor.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '31 [TSK05b2]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 6

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find the filtering functionality.	○	○	○	○	○	○
It is easy to apply the filter on the comprehension model.	○	○	○	○	○	○
The filters available can help me extract information from the comprehension model.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '34 [TSK062]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 7

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find a demo story and load it.	○	○	○	○	○	○
The story browser window is easy to use.	○	○	○	○	○	○
It is easy to find how to share a story.	○	○	○	○	○	○
The feedback messages from the system while performing the task are helpful.	○	○	○	○	○	○

[]I have used the online help facility to perform this task. *

Please choose **only one** of the following:

○ Yes

○ No

[]Please answer the degree at which you agree or disagree with the following statements: *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '37 [TSK072]' (I have used the online help facility to perform this task.)

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
The help available from the system to perform this task is adequate.	○	○	○	○	○	○

Task 8

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find a story in the public story repository.	○	○	○	○	○	○
It is easy to comment on a story.	○	○	○	○	○	○
Comments added by others are clearly presented on the screen.	○	○	○	○	○	○

Task 9

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	Does not apply
It is easy to find how to initiate the collaboration functionality.	○	○	○	○	○	○
The collaboration functionality could be useful for teaching logic programming.	○	○	○	○	○	○
The collaboration functionality is useful for collaborative creating of stories.	○	○	○	○	○	○
The collaboration functionality is useful for collaborative designing of knowledge.	○	○	○	○	○	○
The feedback messages from the system to perform the task are helpful.	○	○	○	○	○	○

System Usability Scale

[]Please answer the degree at which you agree or disagree with the following statements: *

Please choose the appropriate response for each item:

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
I think that I would like to use the Web-STAR IDE frequently.	○	○	○	○	○
I found the Web-STAR IDE unnecessarily complex.	○	○	○	○	○
I thought the Web-STAR IDE was easy to use.	○	○	○	○	○
I think that I would need the support of a technical person to be able to use the Web-STAR IDE.	○	○	○	○	○
I found the various functions in the Web-STAR IDE were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in the Web-STAR IDE.	○	○	○	○	○
I would imagine that most people would learn to use the Web-STAR IDE very quickly.	○	○	○	○	○
I found the Web-STAR IDE very awkward to use.	○	○	○	○	○
I felt very confident using the Web-STAR IDE.	○	○	○	○	○
I needed to learn a lot of things before I could get going with the Web-STAR IDE.	○	○	○	○	○

Thank you for participating.

Submit your survey.
Thank you for completing this survey.