KNOWLEDGE ACQUISITION THROUGH NATURAL LANGUAGE CONVERSATION AND CROWDSOURCING

Luka Bradeško

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Doc. Dunja Mladenić, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Dr. Michael Witbrock, Chair, IBM, New York, New York
Prof. Erjavec, Member, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Iztok Savnik, Member, Univerza v Novi Gorici, Nova Gorica, Slovenia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Luka Bradeško

KNOWLEDGE ACQUISITION THROUGH NATURAL LANGUAGE CONVERSATION AND CROWDSOURCING

Doctoral Dissertation

PRIDOBIVANJE STRUKTURIRANEGA ZNANJA SKOZI POGOVOR TER S POMOČJO MNOŽIČENJA

Doktorska disertacija

Supervisor: Doc. Dunja Mladenić

Ljubljana, Slovenia, April 2017

To the world...

Acknowledgments

Thank everyone who contributed to the thesis: - EU Projects - Cyc - Dave - Michael - Vanessa - Dunja - Coworkers

Abstract

The English abstract should not take up more than one page.

Povzetek

Povzetek v slovenščini naj ne bo daljši od ene strani.

Contents

List of Figures xv

List of Tables xvii

List of Algorithms xix

Abbreviations xxi

Symbols xxiii

Glossary xxv

1 Introduction 1
- 1.1 Scientific Contributions . . . 1
  - 1.1.1 Novel Approach Towards Knowledge Acquisition . . . 1
  - 1.1.2 Knowledge Acquisition Platform Implementation as Technical Contribution . . . 2
  - 1.1.3 A Shift From NL Patterns to Logical Knowledge Representation in Conversational Agents . . . 2
- 1.2 Thesis structure . . . 2

2 Background and Related Work 3
- 2.1 Labour Acquisition . . . 6
  - 2.1.1 Cyc . . . 6
  - 2.1.2 ThoughtTreasure. . . . 6
  - 2.1.3 HowNet . . . 6
  - 2.1.4 Open Mind Common Sense (OMCS) . . . 7
  - 2.1.5 Mindpixel . . . 7
- 2.2 Interaction Acquisition . . . 7
  - 2.2.1 Interactive User Interfaces . . . 7
  - 2.2.2 KRAKEN . . . 8
  - 2.2.3 User Interaction Agenda (UIA) . . . 8
  - 2.2.4 Factivore . . . 8
  - 2.2.5 Predicate Populator . . . 8
  - 2.2.6 Freebase . . . 9
  - 2.2.7 OMCommons (Open Mind Commons) . . . 9
  - 2.2.8 Games . . . 9
  - 2.2.9 20Q (20 Questions) . . . 9
  - 2.2.10 Verbosity . . . 10
  - 2.2.11 Rapport . . . 10
  - 2.2.12 Virtual Pet . . . 10
  - 2.2.13 Goal Oriented Knowledge Collection (GOKC) . . . 10

2.2.14 Collabio (Collbaorative Biography) 11
2.2.15 Interactive Natural Language Conversation 11
2.2.16 AIML(Artificial Intelligence Mark-up Language 11
2.2.17 ChatScript 13
2.2.18 CyN 13
2.3 Mining Acquisition 14
2.3.1 Populating Cyc from the Web (PCW) 14
2.3.2 Learning Reader 14
2.3.3 Never Ending Language Learner (NELL) 15
2.3.4 KnowItAll 15
2.3.5 Probase 16
2.3.6 TextRunner 16
2.3.7 ReVerb 16
2.3.8 R2A2 17
2.3.9 ConceptMiner 17
2.4 Reasoning Acquisition 17
2.5 Acquistion of Geospatial Context 17

3 Knowledge Acquisition Approach **19**
3.1 Architecture 19
3.1.1 Knowledge Base 19

4 Real World Knowledge Acquisition Implementation **23**
4.1 Cyc 23

5 Evaluation **25**

6 Conclusions **27**

References **29**

Bibliography **33**

Biography **35**

List of Figures

Figure 3.1: General Architecture of the KA system, with a simple interaction loop. 21

List of Tables

Table 2.1: Structured overview of related KA systems 5
Table 2.2: AIML Example 12
Table 2.3: AIML example of saving info to variables 12
Table 2.4: AIML Example of remembering answers on specific questions 12
Table 2.5: Simple ChatScript example 13
Table 2.6: Simple ka (remembering) example 13
Table 2.7: Simple ChatScript question/answer/remember example 13

List of Algorithms

Abbreviations

CC	Curious Cat (a name of the knowledge acquisition application and platform that is a side result of this thesis)
CSK	Common Sense Knowledge
CYC	An AI system (Inference Engine and Ontology), developed by Cycorp Inc.
CycKB	Cyc Knowledge Base (Ontology part of Cyc system)
CycL	Cyc Lanugage
GOKC	Goal-Oriented Knowledge Collection
GWAP	Games With A Purpose
JSI	Jožef Stefan Institute
KA	Knowledge Acquisition
KDML	Knowledge SDatabase Mark-up Language
MIT	Massachusetts Institute of Technology
MSR	Microsoft Research
NL	Natural Language
NLP	Natural Language Processing
NP	Noun Phrase
NTU	National Taiwan University
OIE	Open Information Extraction
POS	Part of Speech
POS:X	Abbreviatios for Part of Speech tags used by POS parsers
POS:CC	Coordinating conjunction
POS:CD	Cardinal Number
POS:DT	Determiner
POS:EX	Existential there
POS:FW	Foreign Word
POS:IN	Preposition or subordinating conjunction
POS:JJ	Adjective
POS:JJR	Adjective, comparative
POS:JJS	Adjective, superlative
POS:LS	List item marker
POS:MD	Modal
POS:NN	Noun, singular or mass
POS:NNS	Noun, plural
POS:NNP	Proper noun, singular
POS:NNPS	Proper noun, plural
POS:PDT	Predeterminer
POS:POS	Possesive ending
POS:PRP	Personal pronoun
POS:PRP$	Possesive pronoun
POS:RB	Adverb
POS:RBR	Adverb, comparative

POS:RBS	Adverb, superlative
POS:RP	Particle
POS:SYM	Symbol
POS:TO	to
POS:UH	Interjection
POS:VB	Verb, base form
POS:VBD	Verb, past tense
POS:VBG	Verb, gerund or present participle
POS:VBN	Verb, past participle
POS:VBP	Verb, non-3rd person singular present
POS:VBZ	Verb, 3rd person singular present
POS:WDT	Wh-determniner
POS:WP	Wh-pronoun
POS:WP$	Possesive wh-pronoun
POS:WRB	Wh-adverb
PTT	Taiwanese Bulletin Board System
UW	University of Washington

Symbols

$j^{\star}$ … black-body irradiance
σ … Stefan's (or Stefan-Boltzmann) constant

Glossary

OIE (Open Information Extraction is a paradigm introduced by Oren Etizoni in his TextRunner system. THe main idea of this paradigm is that the knowledge acquisition system is not pre-determined to extract some specific facts, patterns, etc, but is open-ended, extracting large set of relational tuples without any human input.

Chapter 1

Introduction

An intelligent being or machine solving any kind of a problem needs knowledge to which it can apply its intelligence while coming up with an appropriate solution. This is especially true for the knowledge-driven AI systems which constitute a significant fraction of general AI research. For these applications, getting and formalizing the right amount of knowledge is crucial. This knowledge is acquired by some sort of Knowledge Acquisition (KA) process, which can be manual, automatic or semi-automatic. Knowledge acquisition using an appropriate representation and subsequent knowledge maintenance are two of the fundamental and as-yet unsolved challenges of AI. Knowledge is still expensive to retrieve and to maintain. This is becoming increasingly obvious, with the rise of chat-bots and other conversational agents and AI assistants. The most developed of these (Siri, Cortana, Google Now, Alexa), are backed by huge financial support from their producing companies, and the lesser-known ones still result from 7 or more person-years of effort by individuals Finish

Knowledge acquisition and subsequent knowledge maintenance, are two of the fundamental and as-yet not-completely-solved challenges of Artificial Intelligence (AI).

We propose and implement novel approach to automated knowledge acquisition using the user context obtained from a mobile device and knowledge based conversational crowdsourcing. The resulting system named Curious Cat has a multi objective goal, where KA is the primary goal, while having an intelligent assistant and a conversational agent as secondary goals. The aim is to perform KA effortlessly and accurately while having a conversation about concepts which have some connection to the user, allowing the system (or the user) to follow the links in the conversation to other connected topics. We also allow to lead the conversation off topic and to other domains for a while and possibly gather additional, unexpected knowledge. For illustration see the example conversation sketch in Table I, where topic changes from a specific restaurant to a type of dish. In this example case, the conversation is started by the system when user stays at the same location for 5 minutes.

1.1 Scientific Contributions

This section gives an overview of scientific and other contributions of this thesis to the knowledge acquisition approaches.

1.1.1 Novel Approach Towards Knowledge Acquisition

Traditionally KA (knowledge acquisition) approach focuses on one type of acquisition process, which can be either Labor, Interaction, Mining or Reasoning(Zang et al. 2013). In this thesis we propose a novel, previously untried approach that intervenes all aforementioned types with current user context and crowdsourcing into a coherent, collaborative and autonomous KA system. It uses existing knowledge and user context, to automatically deduce and detect missing or unconfirmed knowledge(reasoning) and uses this info to generate crowdsourcing tasks for the right audience at the right time(labor). These tasks are presented to users in natural language (NL) as part of the contextual conversation (interaction) and the answers parsed (mining) and placed into the KB after consistency checks(reasoning). The approach contribution can be summed up as a) definition of the framework for autonomous and collaborative knowledge acquisition with the help of contextual knowledge (chapter X), and b) demonstrate and evaluate the contributions of contextual knowledge and approach in general chapter X.

1.1.2 Knowledge Acquisition Platform Implementation as Technical Contribution

Implementation of the KA framework as a working real-world prototype which shows the feasibility of the approach and a way to connect many independent and complex subsystems. Sensor data, natural language, inference engine, huge pre-existing knowledge base (Cyc)(Lenat 1995), textual patterns and crowdsourcing mechanisms are connected and interlinked into a coherent interactive application (Chapter X).

1.1.3 A Shift From NL Patterns to Logical Knowledge Representation in Conversational Agents

Besides the main contributions presented above, one aspect of the approach introduces a shift in the way how conversational agents are being developed. Normally the approach is to use textual patterns and corresponding textual responses, sometimes based on some variables, and thus encode the rules for conversation. As a consequence of natural language interaction, the proposed KA framework is in some sense a conversational agent which is driven by the knowledge and inference rules and uses patterns only for conversion from NL to logic. This shows promise as an alternative approach to building non scripted conversational engines (Chapter X).

1.2 Thesis structure

The rest of the thesis is structured in to chapters covering specific topics. Chapter X introduces

Chapter 2

Background and Related Work

In this chapter we will give an overview of approaches and related works on broader knowledge acquisition research field, information extraction, crowdsourcing and geo-spatial context mining.

Knowledge Acquisition has been addressed from different perspectives by many researchers in Artificial Intelligence over decades, starting already in 1970 as a sub-discipline of AI research, and since then resulting in a big number of types and implementations of approaches and technologies/algorithms. The difficulty of acquiring and maintaining the knowledge was soon noticed and was coined as *Knowledge Acquisition Bottleneck* in 1977(Feigenbaum 1977). In more recent survey of KA approaches (Zang et al. 2013), authors categorize all of the KA approaches into four main groups, regarding the source of the data and the way knowledge is acquired:

- *Labour Acquisition.* This approach uses human minds as the knowledge source. This usually involves human (expert) ontologists manually entering and encoding the knowledge.

- *Interaction Acquisition.* As in Labour Acquisition, the source of the knowledge is coming from humans, but in this case the KA is wrapped in a facilitated interaction with the system, and is sometimes implicit rather than explicit.

- *Reasoning Acquisition.* In this approach, new knowledge is automatically inferred from the existing knowledge using logical rules and machine inference.

- *Mining Acquisition.* In this approach, the knowledge is extracted from some large textual corpus or corpora.

We believe this categorization most accurately reflects the current state of machine (computer) based knowledge acquisition, and we decided to use the same classification when structuring our related work, focusing more on closely related approaches and extending where necessary. According to this classification, our work presented in this thesis, fits into a hybrid approach combining all four groups, with main focus on interaction and reasoning. We address the problem by combining the labour and interaction acquisition (users answering questions as part of NL interaction aimed at some higher level goal, such as helping the user with various tasks), adding unique features of using user context and existing knowledge in combination with reasoning to produce a practically unlimited number of potential interaction acquisition tasks, going into the field of crowd-sourcing by sending these generated tasks to many users simultaneously.

Previous works that can compare with our solution is divided into the systems that exploit existing knowledge (generated anew during acquisition or pre-existing from before

Fix this, refe
to chapters i
to specifici w

in other sources) (**Witbrock2003**; **Kvo2010**; **Mitchel2015**; Singh et al. 2002; Forbus et al. 2007; Sharma et al. 2010), reasoning (**Witbrock2003**; Speer 2007; Speer, Lieberman, et al. 2008; Y.-L. Kuo et al. 2010), crowdsourcing (**Singh2002**; Speer, Krishnamurthy, et al. 2009; Y.-L. Kuo et al. 2010; Saulo D. S. Pedro et al. 2012; S D S Pedro et al. 2013), acquisition through interaction (**Pedro2012**; Speer, Krishnamurthy, et al. 2009; S D S Pedro et al. 2013), acquisition through labour(add, probably rather refer to subsections) () and natural language conversation(**Pedro2012**; **Witbrock2003**; Speer 2007; Speer, Krishnamurthy, et al. 2009; Y.-L. Kuo et al. 2010).

Test referencing table (see Table 2.1).

Table 2.1: Structured overview of related KA systems

System	Parent	Reference	Category	Source	Representation	Prior K.	Crowds.	Context
Cyc project (Cycorp)	/	(Lenat 1995)	Labour	K. Exp.	CycL	/	/	/
ThoughtTrasure(Signiform)	/	(Mueller 2003)	Labour	K. Exp.	LAGS	/	/	/
HowNet (Keen.)	/	(Dong et al. 2010)	Labour	K. Exp.	KDML	/	/	/
OMCS/ConceptNet (MIT)	/	(Singh et al. 2002)	Labour	Public	ConceptNet	/	✓	/
KRAKEN (Cycorp)	Cyc	(Panton et al. 2002)	Interaction	D. Exp	CycL	✓	/	/
UIA (Cycorp)	Cyc	(Witbrock, Baxter, et al. 2003)	Interaction	D. Exp	CycL	✓	/	/
Factivore (Cycorp)	Cyc	(Witbrock, Matuszek, et al. 2005)	Interaction	D. Exp	CycL	✓	/	/
Predicate Populator (Cycorp)	Cyc	(Witbrock, Matuszek, et al. 2005)	Interaction	D. Exp	CycL	✓	/	/
CURE (Cycorp)	Cyc	(Witbrock 2010)	Interaction	D. Exp	CycL	✓	/	/
OMCommons (MIT)	OMCS	(Speer 2007)	Interaction	Public	ConceptNet	✓	✓	/
Freebase (Metaweb/Google)	/	(Bollacker et al. 2008)	Interaction	Public	RDF	/	/	/
20 Questions (MIT)	OMCS	(Speer, Krishnamurthy, et al. 2009)	Game	Public	ConceptNet	/	/	/
Verbosity (CMU)		(L. V. Ahn et al. 2006)	Game	Public	/	/	✓	/
Rapport (NTU)	ConceptNet	(Y.-l. Kuo et al. 2009)	Game	public	ConceptNet	/	✓	/
Virtual Pet (NTU)	ConceptNet	(Y.-l. Kuo et al. 2009)	Game	public	ConceptNet	/	✓	/
GOKC (NTU)	ConceptNet	(Y.-L. Kuo et al. 2010)	Game	Public	ConceptNet	✓	✓	/
Collabio (MS)	/	(Bernstein et al. 2010)	Game	Public	/	/	✓	/
AIML (Alice foundation)	/	(R. S. Wallace 2003)	Chatbot	/	AIML	/	/	/
Chatscript (Brilligunderstanding)	/	(Wilcox 2011)	Chatbot	/	ChatScript	/	/	/
CyN (Daxtron Labs)	Cyc+AIML	(Wilcox 2011)	Chatbot	/	AIML+Cyc	✓	/	/
PCW (Cycorp)	Cyc	(Matuszek, Witbrock, et al. 2004)	Mining	Web Search	AIML+Cyc	✓	/	/
Learning Reader (NU)	Cyc	(Forbus et al. 2007)	Mining	Web	CycL	✓	/	/
NELL (CMU)	/	(Mitchell et al. 2015)	Mining	Web	Predicate l.	✓	✓	/
KnowItAll(UW)	/	(Etzioni, Popescu, et al. 2004)	Mining	Web Search	text	/	/	/
Probase (MSR)	/	(Wu et al. 2012)	Mining	Web	Propriatery	/	/	/
TextRunner (UW)	KnowItAll	(Soderland et al. 2007)	Mining	Web	text	/	/	/
ReVerb (UW)	TextRunner	(Fader et al. 2011)	Mining	Web	text	/	/	/
R2A2 (UW)	ReVerb	(Etzioni, Fader, et al. 2011)	Mining	Web	text	/	/	/
ConceptMiner (MIT)	ConceptNet	(Eslick 2006)	Mining	Web Search	ConceptNet	✓	/	/

2.1 Labour Acquisition

This category consists of KA approaches which rely on explicit human work to collect the knowledge. A number of expert (or also untrained) ontologists or knowledge engineers is employed to codify the knowledge by hand into the given knowledge representation (formal language). Labour acquisition is the most expensive acquisition type, but it gives a high quality knowledge. It is often a crucial initial step in other KA types as well, since it can help to have some pre-existing knowledge to be able to check the consistency of the newly acquired knowledge. Labour Acquisition is often present in other KA types, even if not explicitly mentioned, since it is implicitly done when defining internal workings and structures of other KA processes. While we checked other well known systems that are result of Labour Acquisition, Cyc (mentioned below) is the most comprehensive of them and was picked as a starting point and main background knowledge and implementation base for this work.

2.1.1 Cyc

The most famous and also most comprehensive and expensive knowledge acquired this way, is Cyc KB, which is part of Cyc AI system (Lenat 1995). It started in 1984 as a research project, with a premise that in order to be able to think like humans do, the computer needs to have knowledge about the world and the language like humans do, and there is no other way than to teach them, one concept at a time, by hand. Since 1994, the project continued through Cycorp Inc. company, which is still continuing the effort. Through the years Cyc Inc. employed computer scientists, knowledge engineers, philosophers, ontologists, linguists and domain experts, to codify the knowledge in the formal higher order logic language CycL (Matuszek, Cabral, et al. 2006a). As of 2006 (Matuszek, Cabral, et al. 2006b), the effort of making Cyc was 900 non-crowdsourced human years which resulted in 7 million assertions connecting 500,000 terms and 17,000 predicates/relations (Zang et al. 2013), structured into consistent sub-theories (Microtheories) and connected to the Cyc Inference engine and Natural Language generation. Since the implemtentation of our approach is based on Cyc, we give a more detailed description of the KB and its connected systems in section 4.1 on page 23. Cyc Project is still work in progress and continues to live and expand through various research and commercial projects.

2.1.2 ThoughtTreasure.

Approximately at the same time(1994) as Cyc Inc. company was formed, Eric Mueller started to work on a similar system, which was inspired by Cyc and is similar in having a combination of common sense knowledge concepts connected to their natural language presentations. The main differentiator from Cyc is, that it tries to use simpler representation compared to first-order logic as is used in Cyc. Additionally, some parts of ThoughtTreasure knowledge can be presented also with finite automata, grids and scripts (Mueller 1999; Mueller 2003). In 2003 the knowledge of this system consisted of 25,000 concepts and 50,000 assertions. ThoughtTreasure was not so successfull as Cyc and ceased all developments in 2000 and was open-sourced on Github in 2015. link as footnote.

2.1.3 HowNet

started in 1999 and is an on-line common-sense knowledge base unveiling inter-conceptual relationships and inter-attribute relationships of concepts as connoting in lexicons of the Chinese and their English equivalents. As of 2010 it had 115,278 concepts annotated with Chinese representation, 121,262 concepts with English representation, and 662,877 knowledge base records including other concepts and attributes (Dong et al. 2010). HowNet knowledge is stored in the form of concept relationships and attribute relationships and is formally structured in KDML (Knowledge Database Mark-up Language), consisting of concepts (called semens in KDML) and their semantic roles.

2.1.4 Open Mind Common Sense (OMCS)

is a crowdsourcing knowledge acquisition project that started in 1999 at the MIT Media Lab(Singh et al. 2002). Together with initial seed and example knowledge, the system was put online with a knowledge entry interface, so the entry was crowd-sourced and anyone interested could enter and codify the knowledge. OMCS supported collecting knowledge in multiple languages. It's main difference from the systems described above (Cyc, HowNet, ThoughtTreasure) is, that it used deliberate crowdsourcing and that it's knowledge base and representation is not strictly formal logic, but rather inter-connected pieces of natural language statements. As of 2013 (Zang et al. 2013), OMCS produced second biggest KB after Cyc, consisting of English (1,040,067 statements), Chinese (356,277), Portuguese (233,514), Korean (14,955), Japanese (14,546), Dutch (5,066), etc. Initial collection was done by specifying 25 human activities, where each activity got it's own user interface for free form natural language entry and also pre-defined patterns like "A hammer is for ________", where participants can enter the knowledge. Although OMCS started to build KB from scratch it shares a similarity to our CC system in a sense that it is using crowdsourcing and also natural language patterns with empty slots to fill in missing parts. OMCS was later used in many other KA approaches as a prior knowledge, similar way as we use Cyc. After a few versions, OMCS was taken from public access and merged with multiple KBs and KA approaches into an ConceptNet KB[1] (Speer, Chin, et al. 2016), which is now (in 2017) part of Linked Open Data (LOD) and maintained as open-source project.

2.1.5 Mindpixel

.

write this:
https://en.w

2.2 Interaction Acquisition

Similarly as with Labour KA, interaction Acquisition gets the knowledge from human minds, but in this case the acquisition is an intended side effect, while users are interacting with the software as part of some other activity/task, or as part of a motivation scheme, such as knowledge acquisition games. Besides games, the interaction could be some other user interface for solving specific tasks, or a Natural Language Conversation. This type of acquisition is most strongly correlated with the approach described in this thesis, since Curious Cat uses points (gaming), to motivate users and it interacts with user in NL, while discussing various topics (concepts). It uses the conversation to set up the context and acquire (remember) user's responses and places them properly in to the KB. Sometimes the acquired knowledge is paraphrased and presented back to user to show the 'understanding', which was first tried in OSMC (section 2.1, (Singh 2002)), but there only in non-conversational way as part of the input forms.

2.2.1 Interactive User Interfaces

Interactive user interfaces are the most common representation of interaction acquisition, where the user interface is constructed in a way to help user enter the data and thus make the acquisition much faster and cheaper. Historically, these systems were developed to help the labour acquisition systems, or on top of them, after parent systems reached some sort of maturity and initial knowledge stability. This is the reason why all of these systems rely or are build on top of labour acquisition (section 2.1) or mining acquisition (section 2.3) systems.

---

[1] http://conceptnet.io/

2.2.2 KRAKEN

system was a knowledge entry tool which allows domain experts to make meaningful additions to CYC knowledge base, without the training in the areas of artificial intelligence, ontology development, or knowledge representation(Panton et al. 2002). It was developed as part of DARPA's Rapid Knowledge Formation (RKF) project in 2000. As its goal was to allow knowledge entry to non-trained experts, it started to use natural language entry and is as this, a first pre-cursor to Curious Cat system and a seed idea for it. It consists of creators, selectors, modifiers of Cyc KB building blocks, tools for consistency checks and tools for using existing knowledge to infer new things to ask. This tool, together with it's derived solutions was later re-written and integrated into Cyc as CURE system (see below). While KRAKEN and later CURE already used Natural Language generation and parsing, and started with the idea of natural language dialogue for doing the KA, the interaction, it was missing user context (user's had to select or search the concept of interest), and also crowdsourcing aspects. Kraken was also missing rules for explicit question asking. The questions were all related to the selected concept and given as a list of natural language forms.

2.2.3 User Interaction Agenda (UIA)

was a web based user interface for KRAKEN KA tool(Panton et al. 2002; Witbrock, Baxter, et al. 2003). It worked inside a browser and it worked as responsive web-app (in 2001) by automatically triggering refresh functionality of the browser. It consisted of a menu of tools that is organized according to the recommended steps of the KE process, text entry box (query, answer, statement), center screen for the main interaction with the current tool, and a summary with a set of colored steps needed to complete current interaction. Similarly as KRAKEN itself, this interface was later improved and integrated into main Cyc system as part of CURE tool.

2.2.4 Factivore

was a Java Applet user interface for an extended KRAKEN system, meant for quick facts entering (Witbrock, Matuszek, et al. 2005). On the back-end it used the same mechanisms and logical templates, while in the front-end it only allowed facts entering, as opposed to UIA, which also allowed rules (which ended up as not being useful).

2.2.5 Predicate Populator

is a similar tool as *Factivore*, which instead of only collecting instances, allows to add general knowledge about classes. For example, instead of describing facts for a specific restaurant, it can collect general knowledge that is true for all restaurants (Witbrock, Matuszek, et al. 2005). The context of the KA in this case, is given by class concept, a predicate and a web-site which is parsed into CycL concepts. These are then filtered out if they do not match argument constraints of the predicate and then shown to user for selection. As part of the validation, this tool had some problems with correctly acquired knowledge. One of the proposed solutions (never implemented), was to start using volunteers to vote about the correctness. This is already a pre-cursor idea for crowd-sourced voting mechanisms that we used in Curious Cat.

2.2.6 Freebase

started in 2007(Bollacker et al. 2008) and was a large (mostly instance based) crowd-sourced graph database for structured general human knowledge. Initially it was acquired from multiple public sources, mostly Wikipedia. The initial seed was then constantly updated and corrected by the community. On the user interface side, Freebase provides an AJAX/Web based UI for humans and an HTTP/JSON based API for software access. For finding knowledge and also software based editing, it uses Metaweb Query Language (MQL). A company behind freebase was bought by Google in 2010 and incorporated into a Google Knowledge Graph. In 2016 Freebase was incorporated into the Wikidata platform and shut down by Google and is no longer maintained.

2.2.7 OMCommons (Open Mind Commons)

is an interactive interface to OMCS which can respond with a feedback to user answers and maintain dialogue (Speer 2007). This is similar approach as we do with Curious Cat and shows understanding of the knowledge users enter. The mechanisms behind is by using inference engine to make analogical inferences based on the existing knowledge and new entry. Then it generates some relevant questions and asks user to confirm them. For example, as given from the original paper, *OMCommons* asks: "A bicycle would be found on the street. Is this common sense?". This is then displayed to the user with the justification for the question: "A bicycle is similar to a car. I have been told that a car would be found on the street". Users then click on "Yes/No" buttons to confirm or reject the inferred statement. The interactive interface also allows its users to refine the knowledge entered by other users and see the ratings. Users can also explore what new inferences are result of their new contributions.

2.2.8 Games

Games are a specific sub-section of interaction acquisition, where the actual acquisition is hidden or transformed into much more enjoyable process, maximizing the entertainment of the users. This type of KA was first officially introduced by Luis von Ahn in 2006 (L. von Ahn 2006; Luis von Ahn et al. 2008) under the name 'Games with Purpose' paradigm.

2.2.9 20Q (20 Questions)

is a game with intentional knowledge acquisition task which focuses to the most salient properties of concepts. The game itself is a standard 20 questions game which aims to make one player figure out the concept of discussion by asking yes/no questions and then infer from the answers what the concept could be. The only difference is that the player which is asking is a computer based on OMCS knowledge base. It generates questions in NL, and according to what a player answers, it attempts to guess the concept. To decide what questions to ask, it uses statistical classification methods (Speer, Krishnamurthy, et al. 2009), to discover the most informative attributes of concepts in OMCS KB. After the user answers all the questions, including whether the detected concept was right or not, the concept and the answers will be assigned to proper cluster and thus the characteristics of the object are learned.

2.2.10 Verbosity

. Similarly as Q20 above, Verbosity is a spoken game for two persons randomly selected online. It was inspired by Taboo board game(Hasbro n.d.) which required players to state common sense facts without mentioning the secret concept. While having similar gameplay as aforementioned board game, Verbositywas developed with the intent to collect common sense knowledge (L. V. Ahn et al. 2006). One player (narrator), gets a secret word concept and needs to give hints about the word to the other player (guesser), who must figure out the word that is described the hints. The hints take the form of sentence templates with blanks to be filled in. For example, if the word is "CAR", the narrator could say "it has wheels." In the experiments, a total of 267 people played the game and collected 7,871 facts. While these facts were mostly a good quality and it was proven that the game can be used successfully, these facts were natural language snippets and were not incorporated into any kind of structure or formal KB.

2.2.11 Rapport

is a KA game based on Chinese OMCS questions, but implemented as a Facebook game to make use of the social connections inside social network. The Game helps users to make new friends or enhance connections with their existing social network by asking and answering questions and matching the answers to other users(Y.-l. Kuo et al. 2009). This game aims to enhance the experience and community engagement and thus functionality of aforementioned *Verbosity* game, by employing simultaneous interaction between all the players versus only 1 to 1 interaction between 2 community members. For evaluation, the answers where multiple users answered the same were considered valid. This game had a similarity with Curious Cat in a sense that it employed the voting mechanism for the same answers, and the repetitive questioning of the same question to multiple users. Authors found out that the agreement between same answers of the repetitive question and voting is 80% or more after at least 2 repetitions of the same question. In 6 months, *Rapport* collected 14,001 unique statements from 1,700 users. Normalized, this is 8.2 answers per user.

2.2.12 Virtual Pet

is a similar game as *Rapport* in a sense that it uses *OMCS* patterns, is in Chinese and is developed by the same authors (Y.-l. Kuo et al. 2009). Instead of Facebook platform, *Virtual Pet* uses PTT (Taiwanese bulleting board system in Chinese language). Instead of direct interaction between the users themselves, users interact with virtual pet and can ask it questions and answer it's questions. In the back-end, the questions the pet asks, are actually questions from other users. This game in 6 months collected 511,734 unique pieces of knowledge from 6,899 users. Normalized this is 74,1 answers per user. While this game attracted much more answers than *Rapport*, the quality of the answers was slightly lower. Authors argue that the reasons behind both is, that users didn't interact directly, but through the virtual pet, so they were less careful whether answers are correct or not.

2.2.13 Goal Oriented Knowledge Collection (GOKC)

. This game builds on the findings and approach of *Virtual Pet* KA game. The main improvement is to try and actually make use of the new knowledge inside a given domain (picked by the initial seed questions), to infer new questions. With this the authors tried to fix a drawback of *Virtual Pet*, that through time, the questions and answers become saturated, and the number of new questions and answers falls exponentially through time, with respect to the number of already collected knowledge peaces. This approach is also aligned with the CC approach, which uses existing+ context and new knowledge, to drive the questions. First part of the *GOKC* paper describes analysis of the knowledge collected by *Virutal Pet* game. The second part is a description and evaluation of GOKC KA approach, where authors did 1 week experiment to show that the approach works. During that week the system inferred created 755 new questions, out of which, 12 were reported as bad. Out of these questions 10,572 answers were collected where 9,734 were voted as good. This results in the 92,07% precision. Compared to the game without question expansion (*Virtual Pet*), which has precision of 80.58%, this is an improvement.

2.2.14 Collabio (Collbaorative Biography)

. This is also a Facebook based game, with the intention to collect user's tags. While the gathered knowledge is more a set of person's tags than knowledge, it served as an inspiration to *Rapport* and *Virtual Pet.* During the experiment, *Collabio* users tagged 3,800 persons with accurate tags with information that cannot be found otherwise(Bernstein et al. 2009; Bernstein et al. 2010).

2.2.15 Interactive Natural Language Conversation

Natural Language Knowledge Acquisition methods are special case of Interaction Acquisition systems. While almost all of the approaches already described above (under Interactive User Interfaces and Games subsections) use natural language to some extent, the language processing used is based on relatively small amount of textual patterns, or statements which are not necessary connected into a conversation. Common denominator of these systems is that they intentionally try to acquire knowledge and then use natural language statements to do this. As a side effect and as motivation for users, sometimes consequent questions and answers give a feeling of conversation. On the other side chat-bots, start with the intention to maintain an interesting conversation with the users, and have to do knowledge acquisition only to remember facts and parts of the past conversations to be able to be smart enough, so users do not lose interest. Starting with Eliza(Weizenbaum 1966), these systems evolved, mostly directed by Turning Tests(**Turing?**), implemented as Loebner competitions, trying to pass it[2]. Through the measure of these tests(Bradeško et al. 2012), among a few propriatery chat-bots, two technologies evolved (*AIML, ChatScript*) to be general enough and can be used for conversational engine (chat-bot) construction and also NL knowledge acquisition.

2.2.16 AIML(Artificial Intelligence Mark-up Language

is an XML based scripting language. It allows developers of chat-bots, to construct a pre-defined natural language patterns and their responses. These definitions are then fed into an AIML engine, which can match user inputs with the patterns and figure out what response to write. AIMLs syntax consists mostly of input rules (categories) with appropriate output. The pattern must cover the entire input and is case insensitive. It is possible to use a wildcard (*) which binds to one or more words. The simplest example of AIML pattern with appropriate response is presented in Table 2.2. This pattern detects user's questions like "Do you have something on the menu?" and responds with "We have everything on the menu."

AIML allows recursive calls to its own patterns, which allows for some really complicated and powerful patterns, covering many examples of input. Regarding the knowledge

---

[2] http://www.loebner.net

Table 2.2: AIML Example

```
<Category>
   <pattern> Do you have * on the menu </pattern>
   <template>
      We have everything on the menu.
   </template>
</Category>
```

acquisition, AIML has an option to store parts of the textual patterns as variables and thus store information for later.

Table 2.3: AIML example of saving info to variables

```
<category>
   <pattern>I just ate *</pattern>
   <template>
      Nice choice! <set name = "food"> <star/></set>
   </template>
</category>
<category>
   <pattern>I am hungry</pattern>
   <template>
      Eat another <get name = "food"/>?
   </template>
</category>
```

The AIML example on Table 2.3 can remember keywords following "I just ate" pattern, like "I just ate pizza". If user at some point later says "I am hungry", the bot is able to respond with "Eat another pizza". In a combination with "<that>" tag, which matches previous computer's response, AIML can be used to construct specific knowledge acquisition questions (Table 2.4). The given example is using AIML 1.0, which was later improved with AIML2.0(R. Wallace 2013) which introduced the *<Learn>* tag, but mechanism stayed mostly the same.

Table 2.4: AIML Example of remembering answers on specific questions

```
<category>
   <pattern>*</pattern>
   <that>What did you order</that>
   <template>
      Was it good? <set name = "menuItem"><star/></set>
   </template>
</category>
```

While AIML language with appropriate engine can remember specific facts, the mechanism is purely keyword based and cannot really count as structured knowledge. Additionally, since it only remembers direct facts, it would be really hard to construct an acquisition of all types of food for example. AIML based chatbots were winning Loebner's competitions in the years from 2000 to 2004, but were later outcompeted by Chatscript based bots and propriatery solutions.

2.2.17 ChatScript

is an NLP expert system consisting of textual patterns rules. It was designed by Bruce Wilcox (Wilcox 2011) and besides patterns it has mechanisms for defining concepts, triple store for facts, own inference engine POS tagger and parser. From the measure of how close the system is to pass the Turing Test as measured by Loebner's competitions, *ChatScript* surpassed *AIML*, and is its successor, since both systems are open sourced. It was designed purposely to be simpler to use and have more powerful tools for NLP and knowledge acquisition which is integral part chatbot systems. A simple example from AIML (Table 2.2) can be re-written in much shorter form as ChatScript rule (Table 2.5).

Table 2.5: Simple ChatScript example

```
?: (do you have * on the menu) We have everything on the menu.
```

Similarly the example from Table 2.3 can be writen as:

Table 2.6: Simple ka (remembering) example

```
s: (I just ate _*) Nice choice!
s: (I am hungry) East another _0?
```

Similarly, example from Table 2.4 in ChatScript looks like:

Table 2.7: Simple ChatScript question/answer/remember example

```
t: What did you order?
        a: (_*) \$menuItem=_0 Was it good?
```

While the above examples repeats the functionality of AIML, ChatScript is more powerful and can remember facts in the shape of (subject verb object) and act on them. This is done with using *ĉreatefact* and *f̂indfact* functions.

2.2.18 CyN

CyN is an AIML interpreter implementation with additional functionality to be able to access Cyc inference engine and KB for both, storing the knowledge and also for querying(Coursey 2004). This was done by introduction of new AIML tags:

- <*cycterm*> Translates an English word/phrase into a Cyc symbol.
- <*cycsystem*> Executes a CycL statement and returns the result.
- <*cycrandom*> Executes a CycL query and returns one response at random.
- <*cycassert*> Asserts a CycL statement.
- <*cycretract*> Retracts a CycL statement.

- <*cyccondition*> Controls the flow execution in a category template.
- <*guard*> Processes a template only if the CycL expression is true

2.3 Mining Acquisition

This category of KA systems try to make use of big text corpus-es available online or otherwise on some digital media. Because the core idea of writing is to share information, there is a lot of knowledge in the texts that can be extracted and converted into a structured knowledge that can later be used by computers. Due to wast size and availability of the data on the internet, mining is most often done on the web resources. Since most of the data format from these corpuses is text, these techniques are particularly strong in the using various NLP techniques, which are often combined with existing knowledge to correct mistakes and check consistency.

2.3.1 Populating Cyc from the Web (PCW)

Since whole idea of Cyc system is to gain enough knowledge through manual work, to be able to learn on itself after some point, the Cyc team is looking int into other means of knowledge acquisition which can automatize or speed-up the KA process. One of the approaches is by mining facts from the Internet by issuing appropriate search engine queries(Matuszek, Witbrock, et al. 2004).

Because Cyc KB is really big, the first step of this approach is to select appropriate part of the kb (concepts and related queries) which are in the interest of the system. For initial experiments a set of 134 binary predicates was selected. These predicates were then used to scan the KB and find the missing knowledge, which was converted to CycL queries. These queries were then converted to NL and queried on a web search engine. The results are then converted back to CycL through NL to logic engine of Cyc. After the conversion these are converted back to NL and re-searched on the web, to check whether the results still hold, and then as the last step, the results are checked for consistency (whether they can be asserted into Cyc). From the initial 134 predicates, the system generated 348 queries, 4290 searches. It found 1016 facts, out of which 4 were rejectec due to inconsistency, 566 rejected by search engine (not same results), 384 were already known to Cyc, and finally 61 new consistent facts were detected as valid. After human review, the findings were that only 32 facts were actually correct.

2.3.2 Learning Reader

Learning Reader(Forbus et al. 2007) is a prototype knowledge mining system that combines NLP, large KB (CycKB) and analogical inference into an automated knowledge extraction system that works on simplified language texts. The system uses Direct Memory Access Parsing (DMAP(Martin et al. 1986)) to parse text and convert it into CycL concepts which are then checked by the inference engine whether they can form correct CycL statements. The prototype consisted of 30,000 NDAP patterns (a quick approximation would be to imagine CyN patterns- chapter 2.2.18). After parsing and syntax checks, found CycL sentences were checked by the inference within CycKB, whether knowledge is new or not. Only new logical statements were then asserted. Additional feature of this prototype system is that it includes question answering mechanism, which can be used by evaluators to check what the system learned. This same mechanism is also used to try to generate new facts (elaborate) and also questions based on newly acquired knowledge. The experiment on 62 written stories improved the recall from 10% to 37% and kept the accuracy as 99.7%

compared to original 100%. By using additional inference and conjecture based inference, the recall raised to 60% while accuracy dropped to 90.8%.

2.3.3 Never Ending Language Learner (NELL)

NELL(Mitchell et al. 2015) is a text mining KA system running 24/7 with the goal to extract knowledge, use this knowledge to improve itself and extract more knowledge. NELL was started in January 2010 and as of 2015 acuired 80 million confidence weighted new beliefs. NELL consists of many different learning tasks (for different types of knowledge), where each task also consist of the performance metrics, so the system can asses itself and check if the learning task itself is also improving through time. In 2015 it consisted of 2500 learning tasks. Some of learning task examples:

- Category Classification
- Relation Classification
- Entity Resolution
- Inference Rules among belief triples

After learning tasks, there is a *Coupling Constraings* component, which combines results of learning tasks. The potentially useful knowledge gets asserted into the KB as candidates, where the assertions are checked by knowledge integrator module which integrates the assertions into the KB, or rejects them.

After some initail initial KB had been gathered, the CMU text-mining knowledge NELL also started to apply a crowdsourcing approach(Saulo D. S. Pedro et al. 2012), using natural language questions to validate its KB. In a similar fashion as Curious Cat, NELL can use newly acquired knowledge, to formulate new representations and learning tasks. There is, however, a distinct difference between the approaches of NELL and Curious Cat. NELL uses information extraction to populate its KB from the web, then sends the acquired knowledge to Yahoo Answers, or some other Q/A site, where the knowledge can be confirmed or rejected. By contrast, Curious Cat formulates its questions directly to users (and these questions can have many forms, not just facts to validate), and only then sends the new knowledge to other users for validation. Additionally, Curious Cat is able to use context to target specific users who have a very high chance of being able to answer a question.

2.3.4 KnowItAll

KnowItAll(Etzioni, Popescu, et al. 2004) is a domain independent web fact extraction system that s specific search engine queries to find new instances of specific classes. It starts it extraction with a small seed of class names and NL patterns like "NP1 such as NP2". The classes and patterns are then used to find instances, new classes and also new extraction phrases by analyzing the results of the web search engines. For example, Googling: "Cities such as *", will return a lot of statements with instances of cities. After these cities are extracted, the names can be used for furhter Googling and by analyzing the phrases in which these cities appear, new patterns can be found, and so on. As part of the experiment, KnowItAll ran for 5 days and extracted over 50,000 instances of cities, states, countries, actors and films. To asses the correctnes of the extractions, the system can fill-in the instances into the various patterns and check the hit-count returned by the search engine./ This then compares by the hit-count of the instance itself and uses this to asses the probability of the instance really belonging to the detected class. For example:

comparing the hit-count of "Ljubljana", "Cities such as Ljubljana" and "Planets such us Ljubljana", the system can figure out that Ljubljana is most likely indeed an instance of the class city.

KnowItAll was the first of the systems that inspired an *Open Information Extraction (Open IE)* paradigm(Etzioni, Fader, et al. 2011) which resulted in many other IE systems such us TextRunner, ReVerb and R2A2. The main ide of this paradigm is to avoid hand labeled examples and domain specific verbs and nouns when approaching textual patterns which can lead to open (without specifying the targets) knowledge extraction on a web scale.

2.3.5 Probase

Probase is a probabilistic taxonomy of concepts and instances consisting of 2.7 million of concepts extracted from 1.68 billion of web pages (Wu et al. 2012). The main difference between Probase and other KBs is that Probase is probabilistic as opposed of "black and white" KB. On the other hand, even if it has much more concepts, it is sparse in the knowledge, since it only uses isA relation (taxonomy). Probase was compared to other taxonomies such as WordNet, YAGO and Freebase in the sense of recall and precition of isA relations. Probase was found to be most comprehensive (biggest recall), while losing at precision measure against YAGO (92.8% vs 95%).

Probase was later renamed as *Microsoft Concept Graph*, and has accessible API[3] which in 2017 consists of 5,401,933 concepts, 12,551,613 instances and 87,603,947 isA relations.

2.3.6 TextRunner

TextRunner is a successor of KnowItAll system (Soderland et al. 2007) and is the first to introduce Open Information Extraction (OIE) paradigm, which's main idea is that it is open-ended and can extract information autonomously without any human intervention which would fix the system to some specific domain or set of concepts/relations. *TextRunner* was ran through over 9 million of web pages, and compared to *KnowItAll* reduced the error rate for 33% on comparable set of extractions. Throughout the experiments, *TextRunner* collected 11mio of high probability tuples and 1 mio concrete facts. *TextRunner* consists of three components.

Self-Supervised Learner is a component started first which takes a small corpus of documents as an input and then outputs a classifier that can detect candidate extractions and classify them as trustworthy or not.

Single-Pass Extractor is a component that makes a single pass through the full corpus and extracts tuples for all possible relations. These tuples are then sent to the classifier trained before, which then marks them as trustworthy or not. Only trustworthy tuples are retained.

Redundancy-Based Assesor is the last step which assigns a probability to each retained tuple, based on the probabilistic model or redundancy (Downey et al. 2005).

2.3.7 ReVerb

With the experiments done with *TextRunner* and *WAE*, it became obvious that OIE systems have a lot of noise and inconistencies in the results. For this reason two syntactical and lexical constraints were introduced in *ReVerb* OIE system(Fader et al. 2011). This helps with removing the incoherent extractions such as "recalled began" which was extracted from sentence "They recalled that Nungesser began his career as precint leader", or uninformative extractions like "Faust, made a deal" extracted from "Faust made a deal with the devil".(Fader et al. 2011). When started, *ReVerb* first identifies relation phrases that match the constraints, then if finds approriate pair of appropriate noun phrase rguments for each identified phrase. The resulting extractions are then given a confidence score using logistic regression classifier.

---

[3] https://concept.research.microsoft.com

2.3.8 R2A2

R2A2 is another improvement in OIE paradigm, since previous systems assumed that relation arguments are only simple noun phrases. Analysis of *ReVerb* errors showed that 65% of errors is on the arguments side (the relation was ok). To fix this, *R2A2* system goes somehow into the direction of kb based KA sytems like Curious Cat with argument constraings.(Etzioni, Fader, et al. 2011). The difference is that *R2A2* is not using hard logic and inference, but rather statistical classifier to detects class constraints (bounds) of the arguments. Compared to *ReVerb*, *R2A2* has much higher precision and recall.

2.3.9 ConceptMiner

ConceptMiner is a KA system built by Ian Scott Eslick as part of his master thesis(Eslick 2006), with the main hipothesis that the seed knowledge collected from vounteers can be then used to bootstrap automatic knowledge acquisition. *ConceptMiner* specifically focuses on binary semantic relationships such as cause, effect, intent and time. The system relies on the prior volunteer knowledge from *ConceptNet* and tests its hipothesis with experimental extractions of knowledge aroun three semantic relations: desire, effect and capability.

2.4 Reasoning Acquisition

adad dada

2.5 Acquistion of Geospatial Context

adad

Chapter 3

Knowledge Acquisition Approach

This chapter introduces the terms, defines formal structure and steps that form our proposed KA approach. First it introduces the general architecture and steps involved in the process(ref to chapter). In the second part, it formalizes the upper ontology and logical constructs required for the KA approach (ref to chapter). After that, each of the crucial steps is described in more detail through examples and additions to the base logical structure defined earlier.

3.1 Architecture

In this section we present the general architecture and workflow of the proposed system depicted also on 3.1, where arrows represent the workflow, squared boxes separate logical sub-systems and different colors representing functionality groups (see the figure legend).

We can see that the system and its user interaction loop are built around the knowledge base in the center (marked in purple and letter A in Figure 3.1). Around the KB, is an integrated Inference engine that can perform inference over the knowledge from the KB. This is represented with the red color and letter B. Tightly connected to the knowledge base and inference engine is a crowdsourcing module, which adds and removes knowledge from the KB based on its consistency among multiple users (Green color and letter F). At the entry and exit point of the systems workflow, there are natural language/logic converters, which are used for communication with the users (blue letter E). Besides the NL endpoints, the system also have a functional endpoint and support, which is used to be able to bring in additional language independent states, such as locations, structured knowledge, etc. In addition to this, the functional part of the application also brings in additional machine learning algorithms and support, and also serves as a glue for all the components, taking care of the interaction between submodules (represented with orange color and letter D). All the modules are triggered either through context (also internal like timer), when it changes, which then causes system to send a request to the user, or through user request directly. This is represented with the arrows, where the blue arrows represent natural language interaction and the orange one structured or functional interaction, where the phone part of the system is interacting automatically without direct user involvement.

3.1.1 Knowledge Base

Internally KB has three components. The main part, which should in any real implementation of the system also be the biggest, is the common-sense knowledge and its upper ontology over which we operate. This part of the system contributes the most to the ability to check the answers for consistency. The more knowledge already exists, the easier becomes to assess the answers. The second part is the user Context KB, which stores the contextual knowledge about the user. This covers the knowledge that the user has provided about himself (section 4.4.2) and the knowledge obtained by mining raw mobile sensors (section 4.4.1). This is represented as the orange arrow, pointing into the context part of the KB. The sensor based context allows the system to proactively target the right users at the right time and thus improve the efficiency and accuracy and also stickiness of the KA process. The third KB part, is the meta-knowledge and KA rules that drive the dialog and knowledge acquisition process (section 4.3.3). Although in our implementation we used Cyc KB and tested Umko KB, the approach is not fixed to any particular knowledge base. But it needs to be expressive enough to be able to cover the intended knowledge acquisition tasks and meta-knowledge needed for the system?s internal workings. After the KB, the second most important part of the architecture is an inference engine (in Fig. 2 marked in red and letter B), which is tightly connected to the knowledge base. The inference engine needs to be able to operate with the concepts, assertions and rules from the KB and should also be capable of meta-reasoning about the knowledge base?s internal knowledge structures. As the individual components (indicated with red color in Fig. 2) suggest, the inference engine is used for: ? Checking the consistency of the users? answers (e.g., can you order a car in a restaurant if it?s not food?). ? Placement of new knowledge inside the KB. ? Querying the KB to answer possible questions. ? Using knowledge and meta-rules to produce responses based on the user and her/his context input (similar in function to the scripts in script-based conversational agents).

Fig. 2. General Architecture of the KA system, with a simple interaction loop At both ends of the stacked chain in Fig. 2, there are natural language processing components (marked in blue and with letter E), which are responsible for logic-to-language and language-to-logic conversion (sections 2.4 and 4.5). These are crucial if we want to interact with users in a natural way and thus avoid the need for users to be experts in first order logic. This module and its components are described in more detail in section 4.5. Besides the main interaction loop, which implicitly uses crowdsourcing while it interacts with the users, there is an additional component (marked in green and with letter F). This ?crowdsourcing and voting? component handles and decides, which elements of knowledge (logical assertions) can be safely asserted and made ?visible? to all the users and which are questionable and should stay visible only to the authors of the knowledge. If the piece of knowledge is questionable, the system marks it as such and then the question formulation process will check with other users whether it?s true or not. This is described in more detail in section 4.7. In addition to logic-based components presented above, there is a functional driver system (marked in orange), which glues everything together, forwards the results of inference to the NL converters, accepts and asserts the context into the KB, handles the synchronization between the instances of the systems, etc.

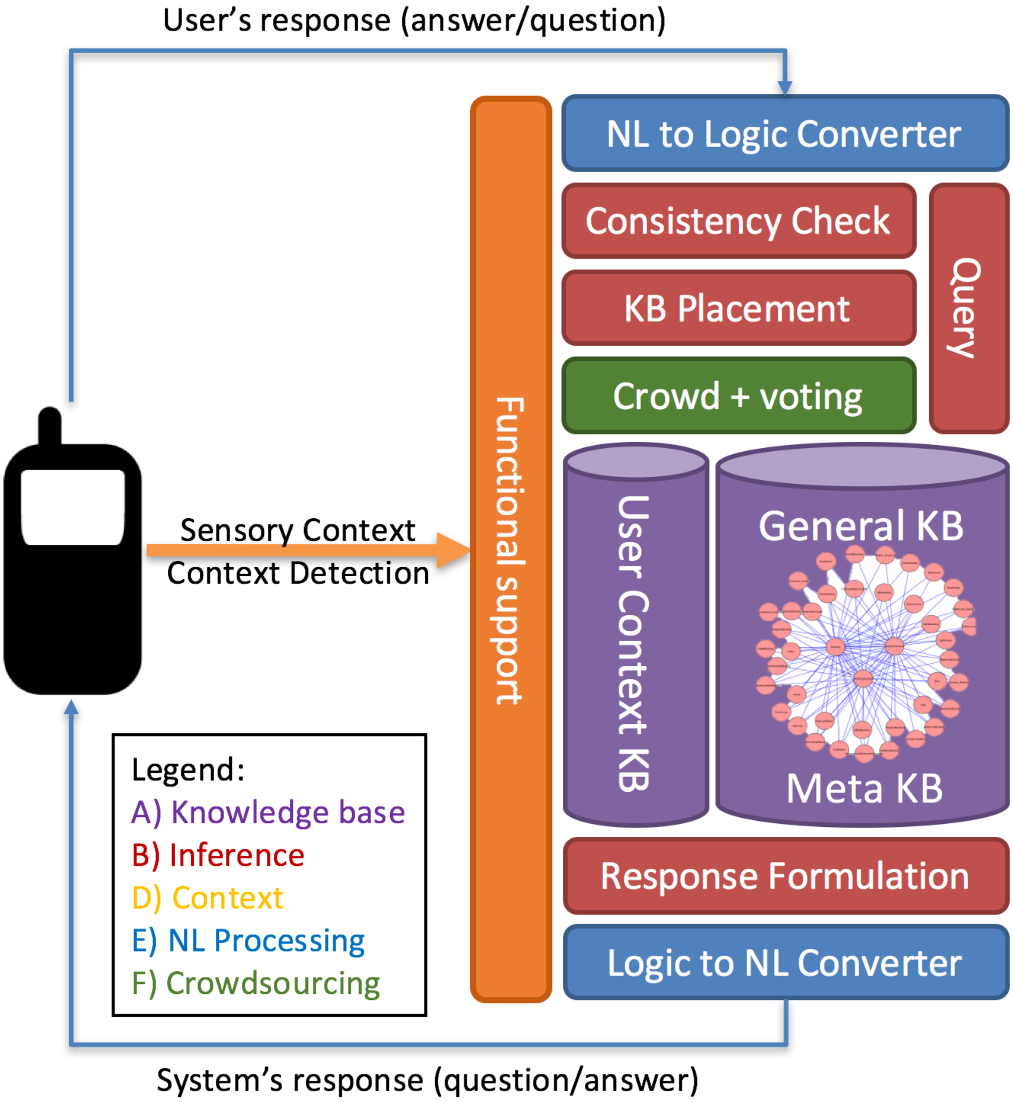
User's response (answer/question)
NL to Logic Converter
Consistency Check
KB Placement
Crowd + voting
Query
Functional support
Sensory Context
Context Detection
User Context KB
General KB
Meta KB
Response Formulation
Logic to NL Converter
Legend:
A) Knowledge base
B) Inference
D) Context
E) NL Processing
F) Crowdsourcing
System's response (question/answer)

Figure 3.1: General Architecture of the KA system, with a simple interaction loop.

Chapter 4

Real World Knowledge Acquisition Implementation

4.1 Cyc

TBW

Chapter 5

Evaluation

TBW

chapters that

Chapter 6

Conclusions

We came to the following conclusions …

References

Ahn, L. von (2006). “Games with a Purpose.” In: *Computer* 39.6, pp. 92–94. ISSN: 0018-9162. DOI: `10.1109/MC.2006.196`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1642623`.

Ahn, Luis Von, Mihir Kedia, and Manuel Blum (2006). “Verbosity : A Game for Collecting Common-Sense Facts.” In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 75–78. ISBN: 1595931783.

Ahn, Luis von and Laura Dabbish (2008). “Designing games with a purpose.” In: *Communications of the ACM* 51.8, p. 57. ISSN: 00010782. DOI: `10.1145/1378704.1378719`.

Bernstein, Michael et al. (2009). “Collabio: a game for annotating people within social networks.” In: *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST ’09)*, pp. 97–100. ISSN: 00325910. DOI: `10.1145/1622176.1622195`. URL: `http://dl.acm.org/citation.cfm?id=1622195`.

— (2010). “Personalization via friendsourcing.” In: *ACM Transactions on Computer-Human Interaction* 17.2, pp. 1–28. ISSN: 10730516. DOI: `10.1145/1746259.1746260`.

Bollacker, Kurt et al. (2008). “Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge.” In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’08. New York, NY, USA: ACM, pp. 1247–1250. ISBN: 978-1-60558-102-6. DOI: `10.1145/1376616.1376746`. URL: `http://doi.acm.org/10.1145/1376616.1376746`.

Bradeško, Luka and Dunja Mladenić (2012). “A Survey of Chabot Systems through a Loebner Prize Competition.” In: *Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies*, pp. 34–37. ISBN: ISBN 978-961-264-048-4.

Coursey, Kino (2004). “LIVING IN CYN : MATING AIML AND CYC TOGETHER WITH PROGRAM N.” In:

Dong, Zhendong, Qiang Dong, and Changling Hao (2010). “HowNet and Its Computation of Meaning.” In: *Coling 2010* August, pp. 53–56. DOI: `10.1142/9789812774675`.

Downey, Doug, Oren Etzioni, and Stephen Soderland (2005). “A probabilistic model of redundancy in information extraction.” In: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1034–1041. ISSN: 10450823. DOI: `10.1016/j.artint.2010.04.024`.

Eslick, Ian Scott (2006). “Searching for Commonsense.” Doctoral dissertation.

Etzioni, Oren, Anthony Fader, et al. (2011). “Open Information Extraction: The Second Generation.” In: *Proc. Int. Joint Conf. Artificial Intell.*

Etzioni, Oren, Ana-maria Popescu, et al. (2004). “Web-Scale Information Extraction in KnowItAll (Preliminary Results).” In: *WWW 2004*.

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). “Identifying relations for open information extraction.” In: *Proceedings of the Conference on ...* pp. 1535–1545. ISSN: 1937284115. DOI: `10.1234/12345678`. arXiv: `arXiv:1411.4166v4`. URL: `http://dl.acm.org/citation.cfm?id=2145596%7B%5C%%7D5Cnhttp://dl.acm.org/`

citation.cfm?id=2145596%7B%5C%%7D5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.226.1089%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf%7B%5C%%7D5Cnhttp://www.cs.washington.edu/research/projects/aiweb/media/papers/etzioni-ijca.

Feigenbaum, E. A. (1977). “The Art of Artificial Intelligence: Themses and Case Studies of Knowledge Engineering.” In: *Proceedings of the 5th International Joint Conference of Artificial itelligence*, pp. 1014–1029.

Forbus, Kenneth D et al. (2007). “Integrating Natural Language , Knowledge Representation and Reasoning , and Analogical Processing to Learn by Reading Learning Reader : The System.” In: *Proceedings of AAAI-07: Twenty-Second Conference on Artificial Intelligence*. Vancouver,BC.

Hasbro (n.d.). *Taboo board game*. URL: https://www.hasbro.com/common/documents/dad288731c4311ddbd0b0800200c9a66/2BF862075056900B1021F6D7061EDCC7.pdf.

Kuo, Yen-Ling and Jane Yung-jen Hsu (2010). “Goal-Oriented Knowledge Collection.” In: *AAAI Fall Symposium: Commonsense Knowledge*, pp. 64–69. URL: http://www.aaai.org/ocs/index.php/FSS/FSS10/paper/viewPDFInterstitial/2278/2605.

Kuo, Yen-ling et al. (2009). “Community-Based Game Design: Experiments on Social Games for Commonsense Data Collection.” In: *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP)*, pp. 15–22. ISSN: 978-1-60558-193-4. DOI: 10.1145/1600150.1600154. URL: http://dl.acm.org/citation.cfm?id=1600150.1600154.

Lenat, Douglas Bruce (1995). “Cyc: A Large-Scale Investment in Knowledge Infrastructure.” In: *Communications of the ACM* 38.22.

Martin, E and I Riesbeck (1986). “Uniform Parsing and Inferencing for Learning.” In: *Proceedings of AAAI-86*, pp. 257–261.

Matuszek, Cynthia, John Cabral, et al. (2006a). “An Introduction to the Syntax and Content of Cyc.” In: *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*. AAAI Press.

— (2006b). “An Introduction to the Syntax and Content of Cyc.” In: *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*. AAAI Press.

Matuszek, Cynthia, Michael Witbrock, et al. (2004). “Searching for Common Sense : Populating Cyc ^{TM} from the Web.” In: *Search*.

Mitchell, T et al. (2015). “Never-Ending Learning.” In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.

Mueller, Erik T (1999). *A database and lexicon of scripts for ThoughtTreasure*. Vol. 1999. CogPrints ID cog00000555 http://cogprints.soton.ac.uk, Article No. 0003004.

— (2003). *ThoughtTreasure: A natural language/commonsense platform*. URL: http://alumni.media.mit.edu/%7B~%7Dmueller/papers/tt.html (visited on 01/01/2017).

Panton, Kathy et al. (2002). “Knowledge Formation and Dialogue Using the KRAKEN Toolset.” In: *Proceedings of the Fourteenth National Conference on Innovative Applications of Artificial Intelligence*, pp. 900–905.

Pedro, S D S, A P Appel, and E R Hruschka Jr (2013). “Autonomously reviewing and validating the knowledge base of a never-ending learning system.” In: *Proceedings of the 22nd ...* pp. 1195–1203. ISBN: 9781450320382. URL: http://dl.acm.org/citation.cfm?id=2488149.

Pedro, Saulo D. S. and Estevam R. Hruschka (2012). “Collective intelligence as a source for machine learning self-supervision.” In: *Proceedings of the 4th International Workshop* *on Web Intelligence & Communities - WI&C '12.* 3, p. 1. ISBN: 9781450311892. DOI: 10.1145/2189736.2189744. URL: http://dl.acm.org/citation.cfm?id=2189736.2189744.

Sharma, Abhishek and Kenneth D Forbus (2010). "Graph-Based Reasoning and Reinforcement Learning for Improving Q/A Performance in Large Knowledge-Based Systems." In: *2010 AAAI Fall Symposium Series*, pp. 96–101. ISBN: 9781577354840. URL: http://www.aaai.org/ocs/index.php/FSS/FSS10/paper/download/2246/2596.

Singh, Push (2002). "The Public Acquisition of Commonsense Knowledge Push Singh The Diversity of Commonsense Knowledge." In: *AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, pp. 47–53. URL: http://www.aaai.org/Papers/Symposia/Spring/2002/SS-02-09/SS02-09-011.pdf.

Singh, Push et al. (2002). "Open Mind Common Sense: Knowledge acquisition from the general public." In: *Cooperative Information Systems Oct. 30-Nov. 1 2002*, pp. 1223–1237. ISSN: 03029743. DOI: 10.1007/3-540-36124-3_77. URL: http://portal.acm.org/citation.cfm?id=646748.701499.

Soderland, Stephen et al. (2007). "Open information extraction from the web." In: *International Joint Conference On Artificial Intelligence*, pp. 2670–2676. ISSN: 00010782. DOI: 10.1145/1409360.1409378. URL: http://portal.acm.org/citation.cfm?id=1625705.

Speer, Robert (2007). "Open mind commons: An inquisitive approach to learning common sense." In: *Proceedings of the Workshop on Common Sense and Interactive Applications.* URL: http://www.fatih.edu.tr/%7B~%7Dhugur/inquisitive/Open%20Mind%20Commons%20An%20Inquisitive%20Approach%20to.PDF.

Speer, Robert, Joshua Chin, and Catherine Havasi (2016). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge." In: Singh 2002. arXiv: 1612.03975. URL: http://arxiv.org/abs/1612.03975.

Speer, Robert, Jayant Krishnamurthy, et al. (2009). "An interface for targeted collection of common sense knowledge using a mixture model." In: *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pp. 137–146. DOI: 10.1145/1502650.1502672.

Speer, Robert, Henry Lieberman, and Catherine Havasi (2008). "AnalogySpace : Reducing the Dimensionality of Common Sense Knowledge." In: *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence*, pp. 548–553. ISBN: 9781577353683.

Wallace, Richard (2013). "AIML 2.0 Draft Specification." URL: http://www.alicebot.org/style.pdf.

Wallace, Richard S. (2003). *The Elements of AIML Style.* Tech. rep. Alice AI Foundation. DOI: 10.1.1.693.3664. URL: http://www.alicebot.org/style.pdf.

Weizenbaum, Joseph (1966). "ELIZA–A Computer Program For the Study of Natural Language Communication Between Man and Machine." In: *Communication of the ACM* 9.1, pp. 36–45. ISSN: 00010782. DOI: 10.1145/365153.365168.

Wilcox, Bruce (2011). "Beyond Façade: Pattern Matching for Natural Language Applications." In: *Gamasutra*, pp. 1–5. URL: http://www.gamasutra.com/view/feature/6305/beyond%7B%5C_%7Dfa%EF%BF%BDade%7B%5C_%7Dpattern%7B%5C_%7Dmatching%7B%5C_%7D.php?page=1.

Witbrock, Michael (2010). "Acquiring and Using Large Scale Knowledge Knowledge Capture : Mixed Initiative." In: *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces.* Cavtat/Dubrovnik, pp. 37–42. URL: http://ieeexplore.ieee.org/document/5546360/?reload=true%7B%5C&%7Dtp=%7B%5C&%7Darnumber=5546360.

Witbrock, Michael, David Baxter, et al. (2003). “An Interactive Dialogue System for Knowledge Acquisition in Cyc.” In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.* Acapulco, Mexico.

Witbrock, Michael, Cynthia Matuszek, et al. (2005). “Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc.” In: *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pp. 99–105. URL: `http://www.aaai.org/Papers/Symposia/Spring/2005/SS-05-03/SS05-03-015.pdf`.

Wu, Wentao et al. (2012). “Probase: A probabilistic taxonomy for text understanding.” In: *Proceedings of the 2012 ACM SIGMOD ...* pp. 481–492. ISSN: 00043702. DOI: `10.1016/j.artint.2011.01.003`. URL: `http://dl.acm.org/citation.cfm?id=2213891`.

Zang, Liang-Jun et al. (2013). “A Survey of Commonsense Knowledge Acquisition.” In: *Journal of Computer Science and Technology* 28.4, pp. 689–719. ISSN: 1000-9000. DOI: `10.1007/s11390-013-1369-6`. URL: `http://link.springer.com/10.1007/s11390-013-1369-6`.

Bibliography

Publications Related to the Thesis

All publications related to the thesis should be referenced in the text.

Other Publications (optional)

...

Biography

The author of this thesis . . .