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

List of Tables

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
NL	...	Natural Language
NTU	...	National Taiwan University
PTT	...	Taiwanese Bulletin Board System

Symbols

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

Glossary

Glossary of terms, dada, bada

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(**Zang2013**). 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)(**Lenat1995**), 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(**Feigenbaum1977**). In more recent survey of KA approaches (**Zang2013**), 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) (**Singh2002a**; **Witbrock2003**; **Forbus2007**; **Kvo2010**; **Sharma2010**; **Mitchel2015**), reasoning (**Witbrock2003**; **Speer2007**; **Speer2008**; **Kuo2010**), crowd-sourcing (**Singh2002**; **Speer2009**; **Kuo2010**; **Pedro2012a**; **Pedro2013**), acquisition through interaction (**Speer2009**; **Pedro2012**; **Pedro2013**), acquisition through labour(add, probably rather refer to subsections) () and natural language conversation(**Pedro2012**; **Speer2007**; **Speer2009**; **Witbrock2003**; **Kuo2010**).

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)	/	(**Lenat1995**)	Labour	K. Exp.	CycL	/	/	/
ThoughtTrasure(Signiform)	/	(**Mueller2003**)	Labour	K. Exp.	LAGS	/	/	/
HowNet (Keen.)	/	(**Dong2010**)	Labour	K. Exp.	KDML	/	/	/
OMCS/ConceptNet (MIT)	/	(**Singh2002a**)	Labour	Public	ConceptNet	/	✓	/
KRAKEN (Cycorp)	Cyc	(**Panton2002a**)	Interaction	D. Exp	CycL	✓	/	/
UIA (Cycorp)	Cyc	(**Witbrock2003UIA**)	Interaction	D. Exp	CycL	✓	/	/
Factivore (Cycorp)	Cyc	(**Witbrock2005**)	Interaction	D. Exp	CycL	✓	/	/
Predicate Populator (Cycorp)	Cyc	(**Witbrock2005**)	Interaction	D. Exp	CycL	✓	/	/
CURE (Cycorp)	Cyc	(**Witbrock2010**)	Interaction	D. Exp	CycL	✓	/	/
OMCommons (MIT)	OMCS	(**Speer2007**)	Interaction	Public	ConceptNet	✓	✓	/
Freebase (Metaweb/Google)	/	(**Bollacker2008**)	Interaction	Public	RDF	/	/	/
20 Questions (MIT)	OMCS	(**Speer2009**)	Game	Public	ConceptNet	/	/	/
Verbosity (CMU)		(**VonAhn2006a**)	Game	Public	/	/	✓	/
Rapport (NTU)	ConceptNet	(**Kuo2009**)	Game	public	ConceptNet	/	✓	/
Virtual Pet (NTU)	ConceptNet	(**Kuo2009**)	Game	public	ConceptNet	/	✓	/
GOKC (NTU)	ConceptNet	(**Kuo2010**)	Game	Public	ConceptNet	✓	✓	/
Collabio (MS)	/	(**Bernstein2010**)	Game	Public	/	/	✓	/

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.

Cyc. The most famous and also most comprehensive and expensive knowledge acquired this way, is Cyc KB, which is part of Cyc AI system (**Lenat1995**). 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 (**Matuszek2006a**). As of 2006 (**Matuszek2006**), 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 (**Zang2013**), 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 17. Cyc Project is still work in progress and continues to live and expand through various research and commercial projects.

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 (**Mueller1999**; **Mueller2003**). 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.

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 (**Dong2010**). 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.

Open Mind Common Sense (OMCS) is a crowdsourcing knowledge acquisition project that started in 1999 at the MIT Media Lab(**Singh2002a**). 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 (**Zang2013**), 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 crowd-sourcing 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] (**Speer2016**), which is now (in 2017) part of Linked Open Data (LOD) and maintained as open-source project.

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, (**Singh2002b**)), 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 2.4) systems.

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(**Panton2002a**). 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

[1] http://conceptnet.io/

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.

User Interaction Agenda (UIA) was a web based user interface for KRAKEN KA tool(**Panton2002a**; **Witbrock2003UIA**). 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.

Factivore was a Java Applet user interface for an extended KRAKEN system, meant for quick facts entering (**Witbrock2005**). 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).

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 (**Witbrock2005**). 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.

Freebase started in 2007(**Bollacker2008**) 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.

OMCommons (Open Mind Commons) is an interactive interface to OMCS which can respond with a feedback to user answers and maintain dialogue (**Speer2007**). 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.

2.2.2 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 (**VonAhn2006**; **VonAhn2008**) under the name 'Games with Purpose' paradigm.

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 (**Speer2009**), 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.

Verbosity. Similarly as Q20 above, Verbosity is a spoken game for two persons randomly selected online. It was inspired by Taboo board game(**TabooGame**) which required players to state common sense facts without mentioning the secret concept. While having similar game-play as aforementioned board game, Verbositywas developed with the intent to collect common sense knowledge (**VonAhn2006a**). 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.

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(**Kuo2009**). 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.

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 (**Kuo2009**). 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.

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.

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(**Bernstein2009**; **Bernstein2010**).

2.2.3 Interactive Natural Language Conversation

Natural Language Knowledge Acquisition 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 pre-defined textual patterns with missing words (concepts) that needs to be filled in. Common denominator of these systems is that they intentionally try to acquire knowledge and then use natural language to help. As a side effect and as motivation for users, sometimes consequent questions and answers give a feeling of conversation. On the other side

Describe also which Games goes under this category. But this should mostly be about chatbots

AIML DAda

CHatSccript

CYCN

2.3 Reasoning Acquisition

adad dada

2.4 Mining Acquisition

adad

2.5 Acquisition with the help of existing knowledge

adad

2.6 Crowdourcing Acquisition

adad

2.7 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.

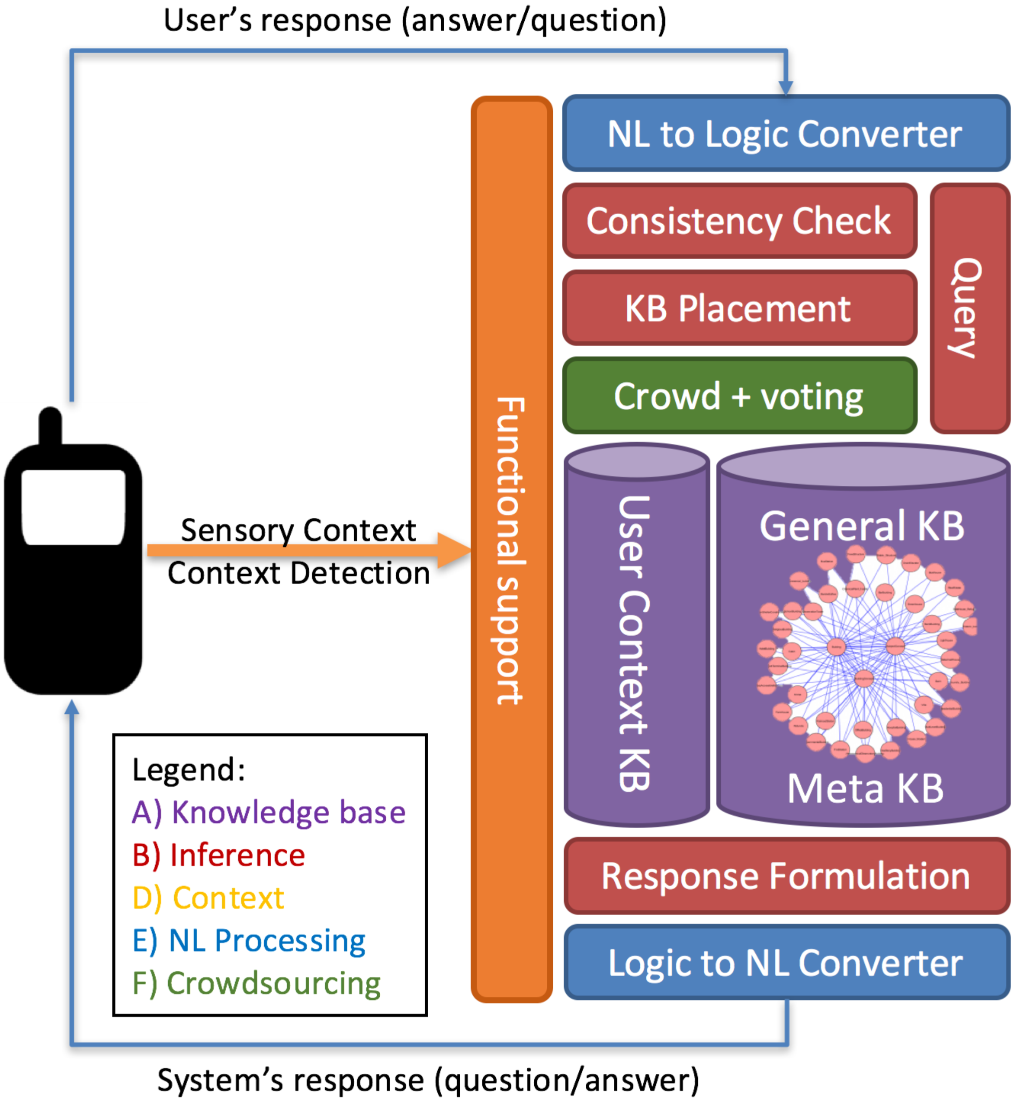

Figure 3.1: General Architecture of the KA system, with a simple interaction loop.

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 ...

Appendix A

Proofs of Theorems

A.1 Proof of the Pythagorean Theorem

Let us prove the Pythagorean Theorem from page **??**.

Proof. This proof is based on the proportionality of the sides of two similar triangles, that is, upon the fact that the ratio of any two corresponding sides of similar triangles is the same regardless of the size of the triangles.

Let ABC represent a right triangle, with the right angle located at C, as shown in Figure A.1. We draw the altitude from point C, and call H its intersection with the hypotenuse AB. Point H divides the length of the hypotenuse into two parts.

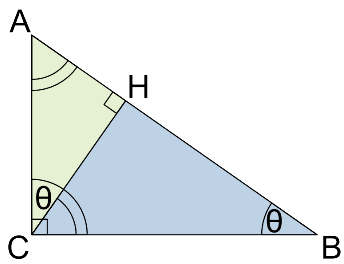

Figure A.1: Similar triangles used in the proof of the Pythagorean theorem.

The new triangle ACH is similar to triangle ABC, because they both have a right angle (by definition of the altitude), and they share the angle at A, meaning that the third angle will be the same in both triangles as well, marked as θ in Figure A.1. By a similar reasoning, the triangle CBH is also similar to ABC.

Similarity of the triangles leads to the equality of ratios of corresponding sides:

$$\frac{BC}{AB} = \frac{BH}{BC} \text{ and } \frac{AC}{AB} = \frac{AH}{AC}. \tag{A.1}$$

The first result equates $\cos\theta$ and the second result equates $\sin\theta$.

These ratios can be written as:

$$BC^2 = AB \times BH \text{ and } AC^2 = AB \times AH. \tag{A.2}$$

Summing these two equalities, we obtain:

$$BC^2 + AC^2 = AB \times BH + AB \times AH = AB \times (AH + BH) = AB^2, \tag{A.3}$$

which, tidying up, is the Pythagorean theorem:

$$BC^2 + AC^2 = AB^2. \tag{A.4}$$

$\square$

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 . . .