Cognitive Systems and Question Answering

Ulrich Furbach, Claudia Schon, Universität Koblenz-Landau
und Frieder Stolzenburg, Hochschule Harz

This paper briefly characterizes the field of cognitive computing. As an exemplification, the field of natural language question answering is introduced together with its specific challenges. A possibility to master these challenges is illustrated by a detailed presentation of the LogAnswer system, which is a successful representative of the field of natural language question answering.

Kognitive Systeme und Fragenbeantwortungssystem

Dieser Beitrag beschreibt das Forschungsgebiet Cognitive Computing. Als Beispiel wird das Gebiet der Beantwortung natürlichsprachlicher Fragen eingeführt und die besonderen Herausforderungen dieses Gebiets aufgezeigt. Eine Möglichkeit, diese Herausforderungen zu meistern, wird durch die detaillierte Präsentation des LogAnswer-Systems vorgestellt, das ein erfolgreiches System im Bereich der Beantwortung von natürlichsprachlichen Fragen ist.

Schlüsselwörter:
Cognitive Computing, Verarbeitung natürlicher Sprache, Fragenbeantwortungssystem, Theorembeweise

Human computer interaction is a discipline with increasing importance. Many people spend a lot of time with computers playing games, watching movies but, of course, also solve problems during their professional activities. This becomes even more important, the more data and information has to be taken into account. Indeed, this amount is increasing every day. Big data and open data are keywords that relate to fields of computer science, where exactly these aspects are tackled.

This paper briefly describes the term cognitive computing and demonstrates that natural language question answering is an example for this new computing paradigm. In the next section, cognitive computing is discussed. After this, a brief overview on natural language question answering is given. Then the LogAnswer system is described and finally we conclude with current extensions and future work.

Cognitive Computing

IBM is certainly one of the major companies that pushed the development of modern computers from the very beginning. With respect to the development of intelligent machines, IBM succeeded twice to set a milestone: In 1997 the chess playing computer Deep Blue managed to beat the world-champion Garry Kasparov. There was a discussion after this match whether the IBM team was cheating during the tournament. Kasparov demanded a rematch, which was refuted and, even more, Deep Blue was dismantled. In 2011 the IBM computer system Watson beat two former winners in the quiz program Jeopardy. In Jeopardy, the players have to understand natural language questions from various domains and give quick answers. This kind of question answering and reasoning is called deep question answering. The Watson system used many different sources of knowledge. Being not connected to the internet, Watson had access to databases, dictionaries, encyclopedias, formal ontologies but also literary works and newspaper articles. Very different to Deep Blue, after this effective public event, the Watson system was developed further and also tailored to various application domains [1]. It is now applied in eHealth, cancer research, finance, and the list is steadily increasing. There is even a version of Watson which is acting as chef, creating really extraordinary dishes, e.g. a Vietnamese Apple Kebab [2]. The keyword which turns the Jeopardy winning system into the basis of a business plan is cognitive computing system. Such a system is designed to learn and to interact with people in a way that the result could not be achieved either by humans or machine on their own. Of course, mastering Big Data also plays an important role - IBM's marketing slogan is "Artificial Intelligence meets Business Intelligence". Such a cognitive computing system has the following properties:

Prof. Dr. Ulrich Furbach ist Professor für Künstliche Intelligenz an der Universität Koblenz-Landau.

Dipl.-Inform. Claudia Schon ist wissenschaftliche Mitarbeiterin in der Arbeitsgruppe Künstliche Intelligenz an der Universität Koblenz-Landau.

Prof. Dr. Frieder Stolzenburg ist Professor für Wissensbasierte Systeme an der Hochschule Harz in Wernigerode.

www.uni-koblenz-landau.de/
koblenz/fb4/ifi/AGKI

a) Multiple knowledge formats have to be processed: Formal knowledge, like ontologies, but also a broad variety of natural language sources, like textbooks, encyclopedias, newspapers and literary works.

b) The different formats of knowledge also entail the necessity to work with different reasoning mechanisms, including information retrieval, automated deduction in formal logic and probabilistic reasoning.

c) The different parts and modules have to interact and cooperate very closely.

d) The entire processing is time critical, because of the interaction with humans.

e) The system must be aware of its own state and accuracy in order to rank its outcome.

Natural language question answering is obviously one example of cognitive computing as depicted above. There are one or several huge text corpora together with other background knowledge, which can be given in various formats. The user interaction is rather simple: The user asks a natural language question and the system answers in natural language. In the following, natural language question answering is briefly introduced.

Natural Language Question-Answering

Up to the 1980s, most systems processing natural language were based on explicit, hand-coded rules. With these rules, the syntactic structure of sentences or the dependency of semantic constituents was analyzed. Here, utterances were understood as to have a syntax characterized by a formal grammar, in particular, a context-free grammar. Later on, syntactic rules were enhanced by feature descriptions of syntactic constituents, e.g. in case grammars or semantic nets [3]. These representations can also be expressed by logical formulas, which lead to so-called phrase-structure grammars. One of the latest and most prominent theories in this respect is HPSG – Head-Driven Phrase Structure Grammar [4].

Starting in the 1980s, machine learning (ML) techniques were introduced in the field of natural language processing (NLP). This changed the field completely. Until that time, the idea was that, in order to process natural language, the sentential structure of an utterance has to be analyzed and hence somehow the meaning understood first. But since computers became much faster, more or less brute-force methods based on statistical analyses and machine learning were employed. Hidden Markov Models were used to predict which word or part of speech is used next [5].
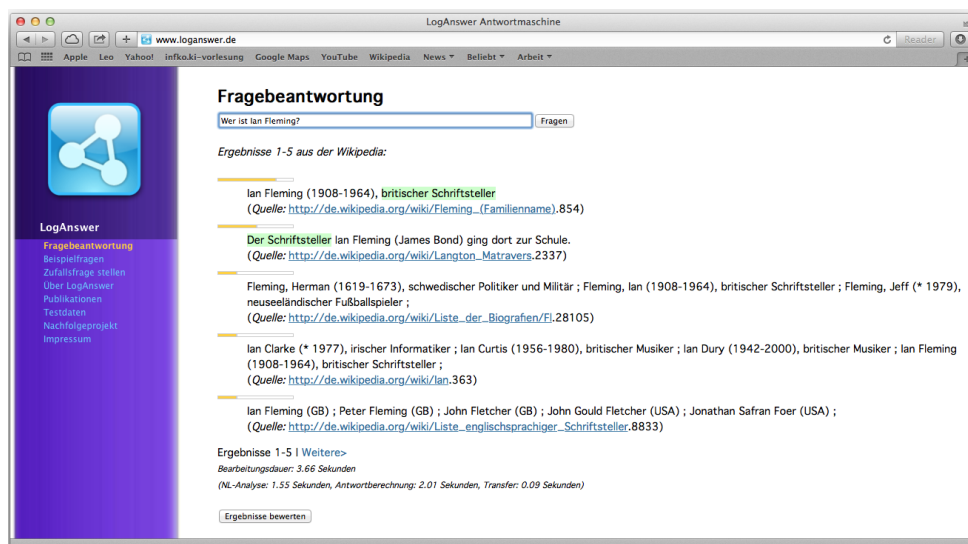
Nowadays, computers and the World Wide Web provide an ever-growing amount of digitally stored knowledge, which is accessible to anyone from home, the workplace or even with mobile devices. While the abundance of available information offers manifold benefits, it can also make the search for some particular data quite tedious. The tool of choice is usually a search engine. However, this is inadequate if the user has a specific question in mind: Instead of simply entering a question, one has to guess suitable keywords.

The field of question answering (QA) intends to improve this search process. A QA system communicates with the user in natural language. It accepts properly formulated questions and returns concise answers. These automatically generated answers are usually not extracted from the web. Rather, the QA system operates on an extensive knowledge base which has been derived from textual sources, employing a natural language interface allowing untrained users an intuitive interaction with the system.

Currently, many QA systems rely on shallow linguistic methods for answer derivation, however, with little attempt to include semantics. This may prevent finding an answer. For example, a superficial word matching algorithm can fail when the textual sources use synonyms of the words in the question. Hence a system must model some form of background knowledge. In summary, cognitive aspects of linguistic analysis, e.g. semantic nets in a logical repre-

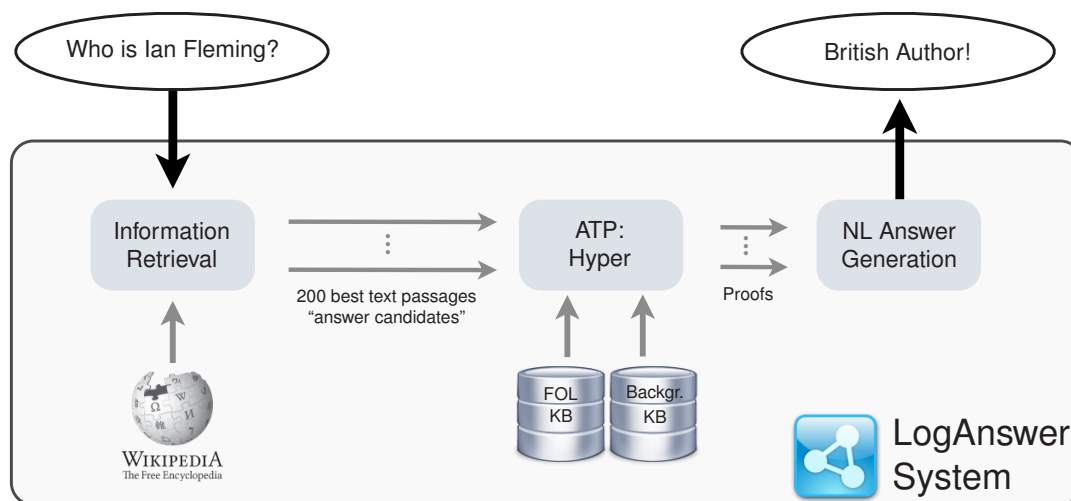**Figure 1: Screenshot of the LogAnswer System.**

sentation, should be combined with machine learning techniques, e.g. when determining the most appropriate answer candidate as done in the LogAnswer system.

The LogAnswer System

LogAnswer [6] is an open domain question answering system. It is accessible by a web interface (http://www.loganswer.de) similar to that of a search engine (Figure 1). The user enters a question into the text box and LogAnswer presents the three best answers, which are highlighted in the relevant textual sources to provide a context.

While many systems for natural language question answering focus on shallow linguistic methods, LogAnswer uses an automated theorem prover (ATP) to compute the replies.

The system was developed in the LogAnswer project which was a cooperation between the IICS (Intelligent Information and Communication Systems) at the FernUniversität in Hagen and the Artificial Intelligence Research Group (AGKI) at the University Koblenz-Landau. The project was funded by the German Research Foundation DFG (Deutsche Forschungsgemeinschaft) and aimed at the development of efficient and robust methods for logic-based question answering.

The IICS is experienced in computational linguistics and knowledge engineering. Within the LogAnswer project the IICS handled the natural language aspects and provided the knowledge base. As an expert in automated theorem proving, the AGKI was responsible for the deductive aspects of the LogAnswer project.

As indicated in (c) in the list of properties of cognitive computing systems, it is important to take care that the different modules interact and cooperate closely. When combining NLP and automated reasoning as in the LogAnswer system, paying attention to the conflicting aims of the two fields is important. Since NLP methods are often confronted with flawed textual data, they strive toward robustness and speed. Nevertheless, they lack the ability to perform complex inferences. In contrast to that, a theorem prover uses a sound calculus to derive precise complex proofs. However, even minor flaws or omissions in the data can lead to a failure of the derivation process. Furthermore, refutationally complete theorem provers can have problems when dealing with large amounts of data due to the fact that they can easily get stuck performing redundant inferences. In the LogAnswer system NLP is used to filter the input for the theorem prover to a fraction of the knowledge available to LogAnswer, and the prover is embedded into a relaxation mechanism which can lessen the proof requirements for imperfect input data [7].

As claimed in (a) in the list of properties, the LogAnswer system uses multiple knowledge formats. One part of the knowledge is provided by a snapshot of the German Wikipedia, which has been translated into a semantic network representation in the MultiNet (Multilayered Extended Semantic Networks) formalism [3]. To make the semantic networks accessible to modern theorem provers, LogAnswer is also equipped with a representation of the MultiNet knowledge base in first-order logic (FOL). See [7] for details on the translation of the MultiNet knowledge base into a first-order logic knowledge base. All in all, 29.1 million natural language sentences have been translated. In addition to that, a background knowledge consisting of 12,000 logical rules and facts is used. This background knowledge provides general knowledge which is advantageous for

Literatur

[1] Kelly III, J. E.; Hamm, S.: Smart Machines: IBM's Watson and the Era of Cognitive Computing. New York 2013.

[2] Why it works: Vietnamese apple kebab. URL: www.youtube.com/watch?v=MNg-5R6Hy_z8#t=18, 2013, Abrufdatum 16.10.2014.

[3] Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Cognitive Technologies. Berlin 2006.

[4] Pollard, C. J.; Sag, I. A.: Head-Driven Phrase Structure Grammar. Chicago London 1994.

[5] Manning, C. D.; Schütze, H.: Foundations of statistical natural language processing. Cambridge, Mass. 2001.

[6] Furbach, U.; Glöckner, I.; Helbig, H.; Pelzer, B.: Logic-based question answering. In: Künstliche Intelligenz 24 (2010) 1, S. 51-55. Special Issue on Automated Deduction.

[7] Furbach, U.; Glöckner, I.; Pelzer, B.: An application of automated reasoning in natural language question answering. In: AI Communications 23 (2010) 2-3, S. 241-265.

[8] Wernhard, C.; Pelzer, B.: System description: E-KRHyper. In: Pfenning, F.: Automated Deduction – CADE-21, Proceedings 2007, S. 508-513. Band 4603 in LNCS.

[9] Baumgartner, P.; Furbach, U.; Niemelä, I.: Hyper tableaux. In: Alferes, J. J.; Pereira, L. M.; Orlowska, E. (Hrsg): Proceedings of the Logics in Artificial Intelligence European Workshop (JELIA) 1996, S. 1-17.

[10] Furbach, U.; Schon, C.; Stolzenburg, F.: Automated reasoning in deontic logic. In: Ragni, M.; Stolzenburg, F. (Hrsg): Proceedings of the KI 2014 Workshop on Higher-Level Cognition and Computation 2014, S. 7-18.

[11] Wirth, C.-P.; Stolzenburg, F.: David Poole's specificity revised. In: Baral, C.; Giacomo, G. D.; Eiter, T. (Hrsg): Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning. Palo Alto, Kalifornien 2014, S. 168-177.

the setting of question answering. Automated reasoning enables the integration of this background knowledge.

In Figure 2 it is depicted how LogAnswer processes a question. Since it is a web-based question answering system, users expect the system to respond quickly. This aspect of time criticality corresponds to (d) in the list of properties and is a serious restriction of the time available for the LogAnswer system to process a question. In such a restricted time, a question cannot be answered directly using the whole knowledge base. Therefore, several different techniques from natural language processing, information retrieval, machine learning and automated deduction come to use. This corresponds to claim (b) in the list of properties. After translating the question into the MultiNet and FOL representation, the Wikipedia content is matched against the given query using retrieval and shallow linguistic criteria. By this, lists of features like the number of matching lexemes between passages and the question or the occurrences of proper names in the passage are computed. Afterwards an ML algorithm ranks text passages using these features. Then up to 200 text passages are extracted from the knowledge base according to this ranking. These so-called answer candidates have a high probability to contain the answer and can be computed rapidly. The computation of feature lists is implemented robustly, which allows to handle documents containing syntactic errors and thus to extract answers from text passages which cannot be parsed completely. In the next step, the theorem prover Hyper [8] is used. The Hyper theorem prover is an implementation of the hypertableaux calculus [9] extended with equality. It has been shown to be very suitable for the type of reasoning problems occurring in the question answering setting, which are characterized by their large number of irrelevant axioms.

With the help of Hyper the answer candidates are tested consecutively. For each of these tests, the logical representation of both the query and an answer candidate together with the background knowledge are fed into Hyper. A successful proof provides an answer by giving an instantiation of the variables of the logical representation of the query. If no proof can be found in time, query relaxation techniques come to pass. These techniques allow certain subgoals of the query to be weakened or dropped in order to enable the prover to find a proof in short time. Query relaxation increases the likelihood of finding an answer even if the knowledge at hand is incomplete. However, the drawback of this technique is that it decreases the probability that the answer found is relevant to the query. As claimed in (e) in the list of properties, the LogAnswer system is aware of its own accuracy, because all proofs are ranked by machine learning algorithms. The three proofs with the highest rank are translated back into natural language answers and are presented to the user.

Conclusions

In this paper, the state of the art in cognitive computing systems and in natural language question answering is discussed. As a prototypical example, the LogAnswer system is described in detail and its properties are checked against criteria for cognitive computing systems.

Currently the LogAnswer system is extended in the follow-up project RatioLog, aiming at the inclusion of rational and human-like reasoning components. In [10] the use of deontic logic for modeling human reasoning and its automating with the logical machinery from LogAnswer is demonstrated. Another extension is with respect to defeasible reasoning [11], which is helpful to determine the best answer from the possibly contradicting answer candidates.