12-2006

A Perception Based, Domain Specific Expert System for Question-Answering Support

Raheel Ahmad
Southern Illinois University Carbondale

Shahram Rahimi
Southern Illinois University Carbondale, rahimi@cs.siu.edu

A Perception Based, Domain Specific Expert System for Question-Answering Support

Raheel Ahmad[1], Shahram Rahimi[2]

Department of Computer Science
Southern Illinois University
Carbondale
[1]rahmad@cs.siu.edu, [2]rahimi@cs.siu.edu

Abstract

The current search engine technologies mostly use a keyword based searching mechanism, which does not have any deductive abilities. There is an urgent need for a more intellgent question-answering system that will provide a more intuitive, natural language interface, and more accurate and direct search results. The introduction of Computing with Words (CwW) provides a new theoretical base for developing frameworks with support for dealing with information in natural language. This paper proposes a domain specific question-answering system based on Fuzzy Expert Systems using CwW. In order to perform the translation of natural language based information into a standard format for use with CwW, Probabilistic Context-Free Grammar is used.

1. Introduction

Most search engines do not possess more than a very basic, keyword-based awareness of the user's needs regarding a query. For a search engine to possess some sort of 'question-answering' ability, it must incorporate a somewhat more sophisticated perception facility. The development of a perceptive system which interacts with a human must be centered around using natural language as this is the most effective means with which humans express and understand thoughts and ideas.

This paper proposes a framework for a domain-specific system based on Fuzzy Expert Systems that answers natural language queries. The core of this system is based on Zadeh's introduction of the concept of Computing with Words (CwW) and Perceptions [15], which is based on fuzzy set theory and fuzzy inference rules. The concept of performing computations directly on "perceptions", which are expressed using "words", is a marked shift from the current trend in computing. This paper is intended to be a step towards the advent of such technology which would hopefully lead to computing machines which are much easier to interact with.

One of the key ideas presented in this paper is the use of Probabilistic Context-Free Grammar (PCFG) for interfacing the fuzzy logic of CwW with natural language. This interface is one of the two areas that needs to be dealt with when developing a framework for CwW, the second area being constructing rules for inferencing the 'answer' to a user query. PCFG provides a quality solution for translating natural language perceptions into the canonical form that Zadeh proposed for CwW [19].

2. Background

2.1 Computing with Words

The proposal of Computing with Words attempts to redefine how we interact with computers, or rather how a computer understands us. The central idea in CwW is the use of the concept of perceptions. A perception in natural language can be shown to consist of a fuzzy constraint on a variable. When a perception's constraint and the constrained variable are then explicated, and the translated perception is produced in the form of a protoform (prototypical form). A protoform is a standard way of representing constraints in CwW.

The generalized constraint is represented as "*X* isr *R"* where the 'r' in 'isr' can be expanded to give various specific constraints such as the following:

X is R	(disjunctive)
X ise R	(equality)
X isp R	(probability)
X isu R	(usuality)
X isv R	(veristic)
X isrs R	(random set constraint)
X isfg R	(fuzzy graph constraint)

One of the key problems when using CwW is translating perceptions given in natural language into a standard format that can be manipulated with computation. Therefore, if a set of documents is available in natural language, it is necessary to translate these documents into a standard format before we can perform any deduction based on the

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00

implicit perceptions in these documents. As mentioned above, in the paradigm of computing with words, this task is represented as *precisiation into Generalized Constraint Language (GCL),* which is a fuzzy logic based *precisiation language.* We propose using Probabilistic Context-Free Grammar, introduced in the next section, to perform this translation.

2.2 Probabilistic Context Free Grammar

A context free grammar (CFG) describes a language by providing a set of production rules which govern how a non-terminal symbol of the language can be expanded into a set of terminal and non-terminal symbols. The CFG forms part of the phase structure (PS) grammars which were introduced by N. Chomsky [11] in 1957 when he applied Post production rules to natural languages. A production rule for a context-free grammar is of the form S -> w, where S is the language symbol and w represents a string of terminal symbols (words) or non-terminal symbols. *Probabilistic* CFG (PCFG) associates a probability with each production rule, with probabilities for rules with the same left hand side being unity. The probability of a sentence parsed by a set of rules is the product of the probabilities of all the involved rules. The computed probability is calculated for the two parse trees and the parse tree with more probability is considered the proper or preferred inference. This process is called syntactic disambiguation [4].

The next section will illustrate the use of PCFG in performing translation of perceptions from natural language to their canonical form.

3. A Fuzzy Expert System for Question-Answering Support

At this point, the difference between a conventional search engine and a question-answering system must be made clear. A search engine has limited usefulness in terms of functionality and usefulness. It traditionally finds certain keywords within a given query and searches for these keywords in the documents within the domain. A question-answering (QA) system allows the user to query the system for knowledge using a restricted natural language syntax where the user expects an answer to the query rather than a list of documents that contain the query keywords.

A question-answering system offers several advantages over traditional keyword based searching:

• *Perceptiveness through use of natural language*: Using natural language as the medium for query ing lets the user be more expressive as well as precise when formulating the query. Traditional keyword-based search invariably results in omitting from the results those documents that may be very relevant to the user but may not contain the exact vocabulary that the user was hoping for.

• *Deductive capability*: The QA system has the capability to understand the particular information that the user is looking for. Therefore, it can be said to possess a level of *intelligence* that is absent from traditional search engines.

• *Query result*: The result that the user gets is not a stream of documents that *may* contain the relevant information but rather than an *answer* which is conclusive and therefor does not require any extra effort from user.

This section describes the framework for a fuzzy Expert System based Query Support mechanism, which can provide question-answering support for a specific domain (Fig. 3). The system is conceptually divided into two modules - a pre-processing module, which builds the fuzzy expert system for the domain, and the actual query system, which allows a user to ask questions about the domain in natural language and provides answers using the expert system. This system is implemented using Fuzzy Clips [7], which is a fuzzy logic extension to the popular Clips expert system.

Below we describe the working of the above components one by one.

3.1 Preprocessing module

This module creates the required Expert System's *rule base*. This is an offline process, similar to the keyword indexing performed in a conventional search engine. The PCFG based system introduced in the last section plays a major role in this module, and this is explained below in more detail.

3.1.1. Using PCFG for translating Natural Language to GCL. Performing language translation is a difficult process specially for an unsupervised natural language translation, either to a formal language or another natural language. There are several issues including syntax, semantics, morphology, and lexicon, which must be taken into consideration when considering a translation. As mentioned earlier, one of the problems to be dealt with in CwW is performing the natural language translation. PCFG provides a well tested tool to describe the grammar of a language, which can be employed in doing the translation from natural language to GCL.

In order to perform the translation, we propose the construction of a set of production rules for each of the above recognized constraints. Therefore a set of production rules for a canonical constraint can be defined for a particular domain, according to the various forms that constraint takes in natural language.

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE

This can be then seen as a mapping of perceptions from a natural language to a standard canonical form. Such a set of rules may differ for the same canonical form from domain to domain.

A *push-down automaton is a* finite state machine equipped with a memory device and can be used to recognize a context free language: for every context free grammar there exists a push down automaton [6], [2]. When used in the capacity of an *acceptor,* it can recognize whether a given sentence was generated by a given PCFG [5]. In the current proposal for the system we have restricted the effort to a single domain.

3.1.2 Generation of Terminal Rules. For a particular domain, we collect a set of all *domain keywords*. These keywords define information that is relevant to the domain and provide the terminal rules for the PCFG. It must be noted that the terminal rules remain the same for all the PCFGs that are written for the various canonical forms. The terminal rules are a part of the general domain knowledge and are shared by all the canonical representations in the grammar [13].

3.1.3 Canonical Form Generation using PCFG. In this step the sentences of the available documents are transformed into perceptions in canonical form. This is done by matching each sentence to a grammar for a particular canonical form. This matching is best performed by employing a statistical parser which derives its grammar and probabilities from the above constructed rules. The main advantage of using a statistical parser such as the one proposed in [8] is that the process is essentially unsupervised.

A particular sentence can be generated in more than one way (more than one parse tree) by a particular PCFG. In this case, the sum of the probabilities of all such parse trees gives the cumulative probability of that PCFG having generated that sentence. Moreover, a sentence can be found to be generated by more than one grammar. The PCFG with the highest such probability is considered to be the canonical form that represents the implicit constraint of this natural language perception.

3.1.4. Facts Generation. The natural sentences are mapped to their related canonical form in the last step. In this step, canonical forms are stored in the expert system as facts. This is a fairly straightforward process since the canonical forms are as close in representation to a fact as required by the expert system. After generating the perceptions in canonical form using a certain PCFG given in the last step, the respective deftemplate for that PCFG is instantiated for each such perception. The perceptions are the facts asserted into the expert system. These facts constitute the 'working memory' of the expert system.

3.1.5. Rules Generation. This step is the basis for the formation of the expert system. The rules for the perception-based expert system would be the main processing unit of the query system. These rules determine the 'answer' of the system given a certain query in the form of a question. Following the core ideas of computing with words, these rules include the major fuzzy propagation rules specified by Zadeh [2], which are comprised mainly of fuzzy inference rules.

These rules are easily implemented in Fuzzy Clips as it supports fuzzy logic and the expression of fuzzy rules such as these. However, there are several issues that need to be addressed, including morphology and syntactic disambiguation, before we can move on to the implementation of the constraint propagation rules. Nonetheless, the expert system model is an ideal fit for stating these rules as long as the issue is restricted to a specific domain.

3.2 Query Module

This module performs the query processing and is responsible for providing the user with an answer for a given question.

1) *Translating the query to a Canonical Form*: This step is similar to step 2 of the preprocessing module and involves translating the query, which should be entered in a restricted natural language format, into a canonical form. It must be noted that we are only concerned with a restricted natural language syntax rather than generalizing the queries to be of a general format. Therefore, the query does not necessarily have to be written like *What is the nearest town to Carbondale*? for our system to be qualified a question-answering system. The main characteristic of the system that we intend to stress in our work is its ability to *perceive* the query and reply accordingly. Nevertheless, the idea of using computing with words is to interface natural language with computing, so the eventual goal is to use a natural language that is as general as possible. However, the current technology in natural language translation has not matured enough to allow such an general interface. After the translation to canonical form, the query is transformed to a fact, in a process similar to that in step 3 of the pre-processing module.

2) *Inferencing using the Rules Base*: The rule base constructed by the preprocessing module is used in this step to provide the "answer" to the query in CF from the last step. The question acts as a trigger for the expert system. The fuzzy constraint propagation rules, represented in the expert system as its rule base, process the required facts from the expert system and the query to provide the resulting fact as an answer.

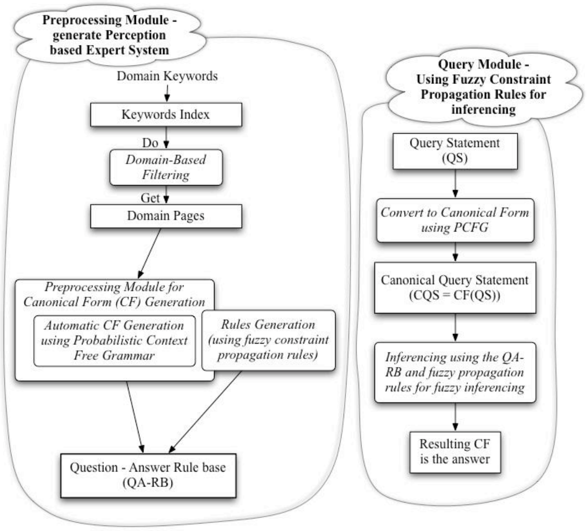

Fig. 3 Architecture of the Question-Answering system

5. Conclusions and Future work

We have proposed a question-answering system that exploits the unique ability of CwW to manipulate perceptions and work with natural language information. The use of Probabilistic Context-Free Grammar as an interface to CwW provides an ability to analyze the structure of a natural language sentence for generating a canonical form. The use of a fuzzy expert system provides the mechanism for implementing the fuzzy constraint propagation rules which perform the inference necessary for generating the final answer for the user.

The focus on our research has been on implementing the proposed system for a spcific domain. The next natural step would be an extension to multiple domains and dealing with the problems listed in the last section. Also, we do not consider the use of a very general or colloquial natural language syntax for our query interface, although the developments in computational linguistics specially those related to context freee grammars can be incorporated in the future to allow a more flexible interface. The future implementations of the expert system would require a deeper understanding of linguistics in general.

6. References

[1] E. Black, F. Jelinek, J. Lafferty, D. Magerman, R. Merger, and S. Roukos, "Towards History-based grammars: using richer models for probabilistic parsing". In *Proceedings of the 31st Annual Meeting of the Association of Computational Linguistics.* 1993, 31-37.

[2] E. Charniak, "Statistical Parsing with Context-Free Grammar and Word Statistics", in *American Association for Artificial Intelligence*, 1997.

[3] E. Charniak and G. Carroll, "Context-sensitive statistics for improved grammatical language models," *Technical Report CS-94-07*, Department of Computer Science, Brown University, 1994.

[4] Z. Chi, "Statistical Properties of Probabilistic Context-Free Grammar", in *Association for Computational Linguistics*, Vol. 25, No. 1, 1999.

[5] N. Chomsky, "Three models for the description of language", *IRE Transactions on Information Theory, 2*, pp. 113-124, 1956.

[6] N. Chomsky, "Context-free grammar and pushdown storage," *Quarterly Progress Report 65,* M.I.T. Research Laboratory in Electronics, pp. 187-194, 1962.

[7] J. Hetherington, "Fuzzy Clips: A Rule Based Teaching Tool", *Processing of the 17th NACCQ*, 2004.

[8] J. Kacprzyk, and S. Zadrozny, "Towards More Powerful Information Technology via Computing with Words and Perceptions: Precisiated Natural Language, Protoforms and Linguistic Data Summaries," *Soft Computing for Information Processing and Analysis*, M. Nikravesh, L. Zadeh, J. Kacprzyk (editors), pp. 19 - 32, Springer, 2005.

[9] D. M. Magerman, and C. Weir, "Efficiency, robustness and accuracy in Picky chart parsing". *In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 40-47, 1992.

[10] J. M. Mendel, "Computing with Words: When words can mean different things to different people," in *Proc. of Third International ICSC Symposium on Fuzzy Logic and Applications*, Rochester Univ., NY, June 1999.

[11] H. Lewis, and C. Papadimitriou. *Elements of the Theory of Computation*, Upper Saddle River, NJ, Prentice Hall, 1998.

[12] M. Nikravesh, and D. Choi, "Soft Computing for Perception Based Information Processing". *Soft Computing for Information Processing and Analysis*, M. Nikravesh, L. Zadeh, J. Kacprzyk (editors), pp. 203 - 226, Springer, 2005.

[13] D. V. Pynadath and M. P. Wellman, "Generalized Queries on Probabilistic Context-Free Grammars", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, 1998.

[14] P. P. Wang (Editor), *Computing with Words*, John Wiley & Sons, Inc., NY, 2001.

[15] L. A. Zadeh, "Fuzzy Logic = Computing with Words", *IEEE Trans. On Fuzzy Systems*, pp. 103-111, 1996.

[16] L. A. Zadeh, "From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Measurement of Perceptions", *IEEE Transactions on Circuits and Systems*, 45(1), pp. 105-119, 1999.

[17] L. Zadeh, "From computing with numbers to computing with words — from manipulation of measurements to manipulation of perceptions," In *International Journal of Applied Math and Computer Science*, pp. 307-324, vol. 12, no. 3, 2002.

[18]L. A. Zadeh and J. Kacprzyk, "Computing With Words in Information/Intelligent Systems*". Heidelberg, Germany: Physica-Verla*, vol. 1, 1999.

[19] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic" *Fuzzy Sets and Systems,* 90, 111–127, 1997.