A Logic Prover for Text Processing

Dan Moldovan and Christine Clark
Language Computer Corporation
Richardson, Texas
{moldovan,christine}@languagecomputer.com

Abstract

This paper demonstrates the applicability of automated reasoning to text processing, specifically to Question Answering. It is shown that the approach is feasible, effective, and scalable. A Logic Prover has been implemented and integrated into a state-of-the-art Question Answering System.

1 Introduction

Automated reasoning has been applied to many application areas [Wos 1988], but little work was done towards applying it to text processing. The challenges faced when using automated reasoning for text processing are: logic representation of open text, need of world knowledge axioms, logic representation of semantically equivalent linguistic patterns, and others.

In this paper we argue that considerable progress was made in key areas of natural language processing and it is possible to implement logic provers for some specific NLP applications. One such application is Question Answering where the problem is to find exact answers to questions expressed in natural language by searching a large collection of documents [Voorhees 2002]. We have implemented a Logic Prover for QA, called COGEX. Integrated into a QA system, the Prover captures the syntax-based relationships such as the syntactic objects, syntactic subjects, prepositional attachments, complex nominals, and adverbial/adjectival adjuncts provided by the logic representation of text. In addition to the logic representations of questions and candidate answers, the QA Logic Prover uses world knowledge axioms to link questions to answers. These axioms are provided by the WordNet glosses represented in logic forms. Additionally, the prover implements rewriting procedures for semantically equivalent lexical patterns. With this deep and intelligent representation, COGEX effectively and efficiently re-ranks candidate answers by their correctness and ultimately eliminates incorrect answers. In this way, the Logic Prover is a powerful tool in boosting the accuracy of the QA system. Moreover, the trace of a proof constitutes a justification for that answer.

2 Logic Prover Inputs and Outputs

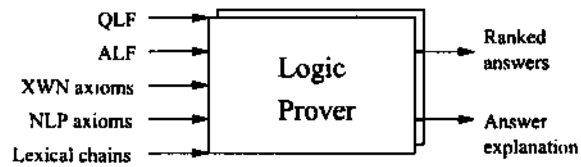

Figure 1: COGEX Architecture

A logic form provides a one to one mapping of the words of the text into the first order logic predicates. The predicate names consist of the base form of the word concatenated with the part of speech of the word[Moldovan and Rus 2001]. An important feature of the Logic Form representation is the fixed-slot allocation mechanism of the verb predicates [Hobbs 1993].

The term Question Logic Form (QLF) refers to the questions posed to the Question Answering system in logic form.

Question: *Which company created the Internet Browser Mosaic ?*

QLF: .organization_AT(x2)) & company_NN(x2) & create_VB(cl,x2,x6) & Internet_NN(x3) & browser_NN(x4) & Mosaic_NN(x5) & nn_NNC(x6,x3,x4,x5)

The term Answer Logic Form (ALF) refers to the candidate answers in logic form. Candidate answers are provided by the Question Answering system [Moldovan 2002].

Answer: *In particular, a program called Mosaic, developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana - Champaign , is gaining popularity*....

ALF: **In_IN(x1,x28) & particular_JJ(x29) & program_NN(x1) & call_VB(e1,x27,x30) & Mosaic_NN(x2) & develop_VB(e2,x2,x31) & by_IN(e2,x8) & National_NN(x3) & Center_NN(x4) & for_NN(x5) & Supercomputing_NN(x6) & application_NN(x7) & nn_NNC(x8,x3,x4,x5,x6,x7) & NCSA_NN(x9) & at_IN(e2,x15) & University_NN(x10) & of_NN(x11) & Illinois_NN(x12) & at_NN(x13) & Urbana_NN(x14) & nn_NNC(x15,x10,x11,x12,x13,x14) & Champaign_NN(x16) & gain_VB(e3,x1,x17) & popularity_NN(x17) ...**

Logic representation of WordNet glosses

A major problem in QA is that often an answer is expressed in words different from those in the question. World knowledge is necessary to conceptually link questions to answers. WordNet glosses contain a source of world knowledge. To be useful in automated reasoning, the glosses need to be transformed into logic forms. Taking the same approach as for open text, we have parsed and represented in logic forms more than 50,000 WordNet glosses.

The question contains the verb *create* while the answer contains the verb *develop.* In order to prove that this answer is in fact correct, we need to detect and use a lexical chain between *develop* and *create.* WordNet supplies us with that chain such that: *develop* ↔ *make* and *make* ↔ *create.*

NLP Axioms

In addition to world knowledge axioms, a QA Logic Prover requires linguistic knowledge. This is what distinguishes an NLP prover from a traditional mathematical prover. General axioms that reflect equivalence classes of linguistic patterns need to be created and instantiated when invoked. We call these NLP axioms and show some examples below. More details are presented in [Moldovan 2003].

1. Complex nominals and coordinated conjunctions. An axiom is built such that the head noun of the complex nominal in the question implies the entire noun phrase:

all xl (mosaic_nn(xl) → internet_nn(xl) & browser_nn(xl) & mosaic _nn(x1))

2. Appositions. An apposition implies that the two noun phrases in the apposition "stand for" each other. *"..Italian Andrea Pfister , designer of the 1979 " bird cage " shoe...* uses an apposition to describe the designer. An axiom is built to link the head of the noun phrases in the apposition such that they share the same argument.

all xl2 x13 xl4 xl5 xl6 xl7 xl8 x19 (italian_nn(xl2) & andrca.nn(xl3) & pfister_NN(xl4) & nn.nnc(xl5,xl2,xl3,xl4) & dcsigner_nn(xl6)&ofJn(x!6,xl7)& 1979_nn(xl7) & birdnn(x!8) & cage_nn(xl9)) —► dcsigncr_nn(xl5) & of_in(xl5,xl7))

3. Part-of relations in location questions. A location seeking question may have a candidate answer that identifies a location by referring to a part of the location. For example, an axiom is built to connect Wyoming to its part:

all xl x2 x3 (corner _nn(xl) & of_in(xl,x2) & wyoming_nn(x2) → Wyoming_nn(xl))

4. Attribute of relations in quantity seeking questions. An axiom is built to connect the quantity to its subject, redwood:

all xl x2 (_quantity_(xl) & redwood JMN(x2) → ofin(xl,x2))

5. *Translating words from noun form to verb form.* Axioms are built to link nouns with verbs. For example the noun *seizure* is linked to the verb *seize,* by filling the object with the tail of the predicate attached to seizure, and assigning the event argmuent to be the argument of the noun predicate for *seizure.*

all x7 x8 x9 (its_prp(x8,x7) & seizure_nn(x8) & ofin(x8,x9) → seize_vb(x8,x7,x9))

3 Control Strategy

The search strategy used is the Set of Support Strategy [Wos 1988]. Axioms placed in the Usable list are: (1) Extended WordNet axioms, (2) NLP axioms, and (3) axioms based on outside world knowledge, such as people and organizations. The inference rule sets are based on hyperresolution and paramodulation.

When the proof fails, we devised a way to incrementally relax some of the conditions that hinder the completion of the proof. This relaxation process puts weights on the proof such that proofs weaker than a predefined threshold are not accepted.

4 Results

COG EX was implemented and integrated into a state-of-the-art Question Answering system that participated in TREC 2002 [Moldovan 2003]. All questions are attempted by the prover, but if the proof fails the QA system resorts to other answer extraction methods that were part of the system before the prover. Thus, some questions are answered by the QA system without the prover, some only by the prover and some by both the non-prover system and the prover. The complete system answered 415 questions out of 500 TREC 2002 questions. Of these, 206 were answered by COGEX. A careful analysis indicates that the QA system without logic prover answered 317 questions and the prover can answer only 98 additional questions for which the system without prover failed. Table 1 summarizes these results. The added value of automated reasoning to the QA system is 30.9% (or 98/317). The failures of the prover are due primarily to the lack of linguistic axioms.

Total number of TREC questions	500
Questions answered by the complete system	415
Questions answered by COGEX	206
Questions answered only by COGEX	98
Questions answered without COGEX	317

Table 1: Performance over 500 TREC 2002 questions

Acknowledgment

This work was supported in part by the ARDA AQUAINT program.

References

[Hobbs 1993] J. Ilobbs, M. Stickel, and P. Martin. Interpretation as abduction. *Artificial Intelligence,* 63 (1993), pp 69-142.

[Moldovan and Rus 2001] D. Moldovan and V. Rus. Logic Form Transformation and its Applicability to Question Answering. In *Proceedings of ACL 2001.*

[Moldovan et. al. 2002] Dan Moldovan et al. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of ACL 2002.*

[Moldovan et. al. 2003] D. Moldovan, C. Clark, S. Harabagiu, S. Maiorano COGEX: A Logic Prover for Question Answering. In *Proceedings of HLT 2003.*

[Voorhees 2002] Ellen Voorhees, Overview of the TREC 2002 Question Answering Track. In *TREC 2002 http://trec.nist.gov*

[Wos 1988] Larry Wos. Automated Reasoning, 33 Basic Research Problems. *Prentice Hall, 1988.*