Syntax-based Concept Extraction for Question Answering Using SEMEX

Demetrios G. Glinos and Fernando Gomez

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362
glinosd@saic.com

Abstract

The SEMEX tool for question answering is presented. Its architecture and features for extracting from input text a network of concept nodes that index syntax-based logical forms, are described. Methods are shown for decomposing questions into boolean combinations of question patterns and for using the concept network and logical forms together with WordNet for question answering. SEMEX's encouraging performance against the TREC 2005 question answering test set is discussed.

Introduction

In question answering (QA), one goal is to understand the information content of sentences such as this excerpt from a newswire article taken from the AQUAINT corpus:

> *Russia is testing a new nuclear submarine "Gepard" (Cheetah) with most advanced technologies on board for the Northern Fleet, which lost a nuclear- powered sub Kursk sinking in the Barents Sea during a naval exercise in August, Russian media reported Friday.*

sufficiently to be able to extract "in August" as the answer to the question, "When did the submarine sink?" Typically, this 41-word sentence appears in the context of a great number of other sentences which, taken together, constitute the textual information repository against which the question is posed.

The following sections present the SEMEX ("SEMantic Extractor") tool for automatically extracting a network of concept nodes from input text, and for answering questions posed against this concept base.

SEMEX Architecture

The tool creates a concept network by processing the input text through a cascade of modules for: (1) part of speech tagging; (2) partial parsing; (3) chunking; (4) sentence decomposition; (5) resolution; and (6) concept extraction.

Prior to tagging, SEMEX removes spurious HTML escape codes, newswire datelines, paragraph tags, and similar cleanup items. Punctuation is also separated; dates, numbers and proper nouns are aggregated; and initials are flagged so the tagger does not interpret them as sentence boundaries. Tagging is performed by the Brill tagger (Brill 1994), whose output is corrected as needed.

Parsing is performed using the Cass partial parser (Abney 1996). SEMEX then applies its own comprehensive set of empirically derived heuristics to build up phrases at the chunking stage. The resultant parse trees are then simplified and reorganized in the sentence decomposition stage to create separate propositions for the clauses, appositions and coordinations of complex sentences. Syntactic roles are assigned to the proposition components and pronoun references are resolved. And finally, a network of concepts is built from the discourse entities contained in the resolved propositions.

To answer a question, SEMEX's question analysis module first resolves pronoun references to the target or subject of the question, and then tags and parses the question to produce a question pattern, or a boolean combination of patterns, of the same form as for propositions, except that the expected answer is replaced with a free variable. To extract the answer, SEMEX performs a unification of the question patterns with the relevant logical forms retrieved from the concept network. WordNet (Fellbaum 1998) is used in the unification process to improve recall, as more fully described below.

Sentence Decomposition

SEMEX decomposes sentences into logical propositions, each of which represents a single action, from which it extracts logical forms consisting of actions as predicates and their arguments. Using a comprehensive set of heuristics that operate on the syntactic chunks, part of speech tags for key tokens, punctuation, WordNet's ability to characterize alternative parts of speech for key words, and in some cases the individual key words themselves, SEMEX creates atomic propositions for sentences containing subordinate clauses, non-defining relative clauses, verb and noun coordinations, and appositions.

Concept Extraction

SEMEX extracts from each resolved proposition a logical form, referred to as a "proposition tuple", which is given by *<subject,verb,gerinf,modifiers, indirect,direct>,* where "subject" refers to the noun phrase representing the subject of the sentence, "verb" refers to the main verb phrase, "gerinf" represents any gerund or infinitive form, "modifiers" refers to adverbials and adverbial complements, typically prepositional phrases, "indirect" refers to the indirect object, if any, and "direct" refers to the direct object, if any.

To support complex question answering, SEMEX implements a concept node as the 4-tuple: *<name, {parents},{children},{tuples}>,* where "name" refers to the noun phrase for the discourse entity for which the node is constructed, "{parents}" and "{children}" refer to the (possibly empty) sets of parent and children nodes for the concept, and "{tuples}" contains links to the proposition tuples in which the concept (discourse entity) appears.

SEMEX establishes parent-child "is-a" links based on tuples that encode explicit copular relationships and derivations based on existing concept node names. SEMEX currently implements over twenty parent-child derivations, including: (a) common noun-proper noun, for example, "space shuttle Atlantis" is-a "space shuttle"; (b) common noun-common noun, for example, "oil tanker" is-a "tanker"; and (c) proper noun-preposition, for example, "King of England" is-a "king".

By construction, the set of concept nodes is organized as a network of possibly disjoint subnetworks.

Question Analysis

SEMEX decomposes questions into proposition tuples in the same manner as for the document set, with the addition of free variables for the desired answer. These variables may take the form of a general directive, such as "*who", "*what", *when", "*where", and "*why", or a target preposition type, such as "*in", when the answer is expected to be modified by a preposition. The answer variable "*ans" is also used to serve as a referent to a candidate answer obtained in response to a previous tuple, so that question patterns may be constructed as boolean combinations of separate tuple patterns. Additionally, SEMEX augments the pattern set with passive patterns for active constructions, and vice versa, and also with possessive patterns for pattern subjects or direct objects that include the preposition "of."

Question Answering

Once the question is analyzed, SEMEX performs question answering by first identifying the tuples of interest in the concept network, and then by matching them against the question patterns to extract the answer. The tuples of interest are obtained from the concept nodes for the noun phrases extracted from the various components of the question tuple or tuples. SEMEX then uses a straightforward unification algorithm in which the entire boolean combination of question tuples is applied to each tuple until an answer is found, for a factoid question, or for a list question, until all tuples are examined.

Each proposition tuple is examined by checking its tuple components against the non-null components of the question pattern. For each such component that does not involve a free (answer) variable, a matching algorithm is executed. WordNet is used for robustness in matching. If any such component fails to match, the proposition tuple is rejected and examination proceeds to the next tuple in line. If the question pattern component contains a target answer variable, an answer retrieval algorithm is executed according to the component type.

A Test Implementation

The annual Text Retrieval Conferences (TRECs) have been a focal point for leading edge question answering systems (Voorhees 2005). SEMEX was configured to exercise the TREC 2005 QA test set, against the top fifty documents for each target returned by the NIST's generic IR engine from the AQUAINT newswire document collection. A detailed review of SEMEX performance against the first 200 factoid questions shows that, after taking into account the limitations of the heuristic parser and the answer documents that were not read due to spurious characters, SEMEX answered correctly 60 out of 129 questions (46%), which is good performance for a QA system. Of the 71 discards, 29 involved questions that had no answers in the document sets, 14 involved the headline or dateline, 6 involved question types that were not implemented, and 8 had no answer or an incorrect answer.

References

Abney, S. 1996. Partial Parsing via Finite-State Cascades. In Proceedings of Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information, 8-15. Prague, Czech Republic.

Brill, E. 1994. Some Advances in Part of Speech Tagging. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94). Seattle, Washington: American Association for Artificial Intelligence.

Fellbaum, C. ed. 1998. WordNet: An Electronic Lexical Database. Cambridge, Mass.: The MIT Press.

Voorhees, E.M. 2005: Overview of the TREC 2004 Question Answering Track. In Voorhees, E.M., and Buckland, L.P. eds.: Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.