Intentions, Implicatures and Processing of Complex Questions

Sanda M. Harabagiu and **Steven J. Maiorano** and **Alessandro Moschitti** and **Cosmin A. Bejan**
University of Texas at Dallas
Human Language Technology Research Institute
Richardson, TX 75083-0688, USA

Abstract

In this paper we introduce two methods for deriving the intentional structure of complex questions. Techniques that enable the derivation of implied information are also presented. We show that both the intentional structure and the implicatures enabled by it are essential components of Q/A systems capable of successfully processing complex questions. The results of our evaluation support the claim that there are multiple interactions between the process of answer finding and the coercion of intentions and implicatures.

1 Introduction

The Problem of Question Intentions.
When using a Question Answering system to find information, the user cannot separate the intentions and beliefs from the formulation of the question. A direct consequence of this phenomenon is that the user incorporates his or her intentions and beliefs into the interrogation. For example, when asking the question:
Q1: What kind of assistance has North Korea received from the USSR/Russia for its missile program?

the user associate with the question a number of intentions, that maybe expressed a set of *intended questions*. Each intended question, in turn generates implied information, that maybe expresses as *implied questions*. For question *Q1*, a list of intended questions and implied questions is detailed in Table1.

Most of the intended questions are similar with the questions evaluated in TREC[1]. For example questions Q^i_1, Q^i_2 and Q^i_3 are so-called *definition questions*, since they ask about defining properties of an object. However unlike the TREC definition questions, these questions express unstated intentions of the questioner and need to be processed in the context of the original complex question *Q1*. Questions Q^i_4 and Q^i_5 are *factoid questions*, requesting information about facts or events. Q^i_6 asks about the source of information that enables the answers of question *Q1*.

Questions Q^i_1, Q^i_2, Q^i_3, Q^i_4 and Q^i_5 result from the intentional structure generated when processing question *Q1* or questions similar to it. When intended questions are generated, their sequential processing (a) represents a decomposition of the complex question and (b) generates a scenario for finding information; thus questions like *Q1* are also known as *scenario questions*.

Intentions and Implicatures.
As Table 1 suggests, the implied information takes the form of alternatives that guide the answers to intended questions. For example, question Q^{m1}_1 lists alternatives for the answer to Q^i_1 whereas Q^{m2}_1 lists components of the answer of Q^i_1. Implicatures may also involve temporal inference, e.g. the implied questions pertaining to Q^i_3 and Q^i_4. Additionally, the reliability of information is commonly an implicature in the case of scenario questions, since the causal and temporal inference is based on the quality and correctness of the available data sources. Neither intentions or implicatures are recognizable at syntactic or semantic level, but they both play an important role in the question interpretation. Interpretations disregard the implied information or the user intentions determine the extraction of incorrect answers, thus influence the performance of Q/A systems.

Our solution.
In this paper we present two different mechanisms of deriving the question implicatures. Both methods start from the syntactic and semantic content of the interrogation. The first method considers only the semantic

[1] The Text REtrieval Conferences (TREC) are evaluation workshops in which Information Retrieval tasks are annually tested. Since 1999 the performance of question answering systems are measured in the TREC QA track.

Intended Questions	**Implied Questions**
Q_1^i :*What is the USSR/Russia?*	Q_1^{m1} :*Is this the Soviet/Russian government?* Q_2^{m1} :*Does it include private firms, state-owned firms, educational institutions, and individuals?*
Q_2^i :*What is North Korea?*	Q_1^{m2} :*Is this the North Korean government only?* Q_2^{m2} :*Does it include private firms, state-owned firms, educational institutions, and individuals?*
Q_3^i :*What is assistance?*	Q_1^{m3} :*Is it the transfer of complete missile systems, licensing agreements, components, materials, or plans?* Q_2^{m3} :*Is it the training of personnel?* Q_3^{m3} :*What kind of training?* Q_4^{m3} :*Does transfer include data, and, if so, what kind of data?* Q_5^{m3} :*Does transfer include financial assistance, and, if so, what kind of financial assistance?*
Q_4^i :*What are the missiles in the North Korean inventory?*	Q_1^{m4} :*Are any based upon Soviet/Russian designs?* Q_2^{m4} :*If so, which ones?* Q_3^{m4} :*What was the development timeline of the missiles?* Q_4^{m4} :*Did any timeline differ significantly from others?* Q_5^{m4} :*Did North Korea receive assistance from other sources besides USSR/Russia to develop these missiles?*
Q_5^i :*When did North Korea receive assistance from the USSR/Russia?*	Q_1^{m5} :*Was any intended assistance halted, stopped or intercepted?*
Q_6^i :*What are the sources of information?*	Q_1^{m6} :*Are the sources reliable?* Q_2^{m6} :*Is some information contradictory?*

Table 1: Question decomposition associated with question *Q1*

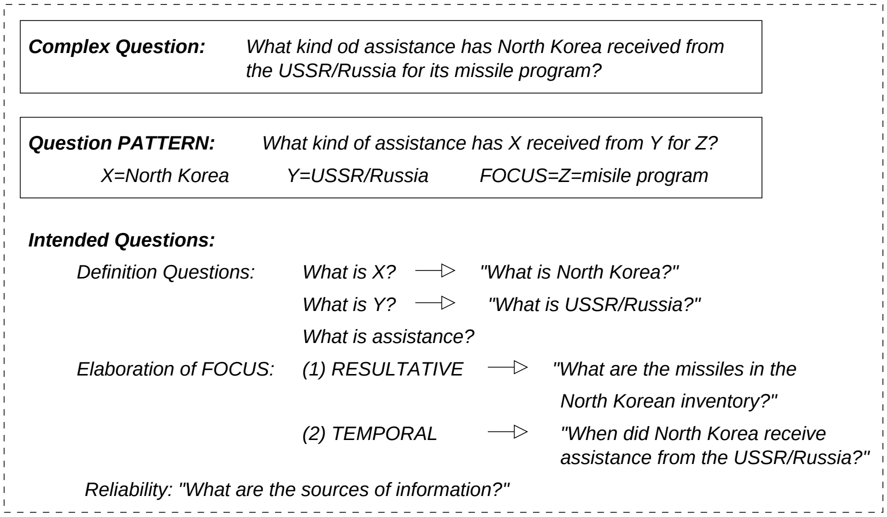
meaning of the words used in the question whereas the second method considers the predicate-argument structure of the question and candidate answers as a form of shallow semantics that enables the inference of the intentional structure. Question implicatures are derived from lexico-semantic paths retrieved from the WordNet lexico-semantic database. These paths bring forward new concepts, that may be associated with the question implicatures when testing the paths against the conversational maxims introduced by Grice in (Grice, 1975a). For example, if the user asks *"Will Prime Minister Mori survive the crisis?"*, the first method detects the user's belief that the position of the Prime Minister is in jeopardy, since the concept DANGER is coerced although none of the question words directly imply it.

The second method generates the intentional structure of the question, enabling a more structured representation of the pragmatics of question interpretation. The intentional structure is based on a study that we have conducted for capturing the motivations of a group of users when asking series of questions in several scenarios. We show how the intentional structures that we have gathered guide the coercion of knowledge that helps to support the acceptance of rejection of computational implicatures.

The derivation of intentional structures is made possible by predicate-argument structures that are recognized both at the question level and at the candidate answer level. In this paper we show how richer semantic objects can be derived around predicate-argument structures and how inferential mechanisms can be associated with such semantic objects for obtaining correct answers. The rest of the paper is organized as follows. In Section 2 we describe several forms of complex questions that require the derivation of computational implicatures. Section 3 details the models of Question Answering that we considered and Section 4 shows our methods of deriving predicate-argument structures and their usage in identifying answers for questions. Section 5 details the intentional structures whereas Section 6 summarizes the conclusions.

2 Question Complexity

Since 1999, the TREC QA evaluations focused on factoid questions, such as *"In what year did Joe Di Maggio compile his 56-game hitting streak?"* or *"Name a film in which Jude Law acted."*. The answers to most of these questions belong to semantic categories associated with each question class. For example, questions asking about a date or a year can be answered because Named Entity Recognizers identify a temporal expression in a candidate text span. Similarly, names of people or organizations are provided as answers to questions such as *"Who is the first Russian astronaut?"* or *"What is the largest software company in the world?"*. Most Named Entity Recognizers detect names of PEOPLE, ORGANIZATIONS, LOCATIONS, DATES, PRICES and NUMBERS. For factoid Q/A, the list of name categories needs to be extended, as reported in (Harabagiu et al., 2003) for recognizing many



Complex Question: *What kind od assistance has North Korea received from the USSR/Russia for its missile program?*

Question PATTERN: *What kind of assistance has X received from Y for Z?*

X=North Korea Y=USSR/Russia FOCUS=Z=misile program

Intended Questions:

Definition Questions: What is X? → "What is North Korea?"

What is Y? → "What is USSR/Russia?"

What is assistance?

Elaboration of FOCUS: (1) RESULTATIVE → "What are the missiles in the North Korean inventory?"

(2) TEMPORAL → "When did North Korea receive assistance from the USSR/Russia?"

Reliability: "What are the sources of information?"

Figure 1: Decomposition of scenario question into intended questions

more types of names, e.g. names of movies, names of diseases, names of battles. Moreover, the semantic categories of the extended set of names need to be incorporated into an answer type taxonomy that enables the recognition of (a) the expected answer type and (b) the question class. The taxonomy of expected answer types is useful because the answer is not always a name; it can be a lexicalized concept or a concept that is expressed by a paraphrase.

The TREC evaluations have also considered two more classes of questions: (1) list questions and (2) definition questions. The list questions have answers that are typically assembled from different documents. Such questions are harder to answer than factoid questions because the systems must detect duplications. Example of list questions are *"Name singers performing the role of Donna Elvira in performances of Mozart's "Don Giovani"."* or *"What companies manufacture golf clubs?"*. Definition questions require a different form of processing that factoid questions because no taxonomy of answer types needs to be used. The expected answer type is a definition, which cannot be represented by a single concept. Q/A systems assume that definitions are given by following a set of linguistic patterns that need to be matched for extracting the answer. Example of definition questions are *"What is a golden parachute?"* or *"What is ETA in Spain?"*.

In (Echihabi and Marcu, 2003) a noisy channel model for Q/A was introduced. This model is based on the idea that if a given sentence S_A contains an answer substring A to a question Q, then S_A can be re-written into Q through a sequence of stochastic operators. Not only a justification of the answer is produced, but the conditional probability P(Q—S_A) re-ranks all candidate answers.

A different viewpoint of Q/A was reported in (Ittycheriah et al., 2000). Finding the answers A to a question Q was considered a classification problem that maximizes the conditional probability P(A—Q). This model is not tractable currently, because (a) the search space is too large for a text collection like the TREC or the AQUAINT corpora; and (b) the training data is insufficient. Therefore, Q/A is modeled by the distribution P(C—A,Q) where C measures the *"correctness"* of A to question Q. By using a hidden variable E that represents the expected answer type, P(C—A,Q) = Σ_E p(C,E—Q,A) = Σ_E p(C—E,Q,A) * p(E—Q,A). Both distributions are modeled by using the maximum entropy.

All three forms of questions are also useful when processing complex questions, determined by a scenario resulting from a problem-solving situation. As illustrated in Figure 1, a scenario question may be associated with a pattern. One of the pattern variables represents the *focus* of the question. The notion of the question focus was first introduced by (Lehnert, 1978). The focus represents the most important concept of the question; a concept determining the domain of question. In the case of question *Q1*, the focus is *missile program*. The identification of the focus is based on the predicate-structure of the question pattern and on the order of the arguments. Figure 3 shows both the question pattern associated with *Q1* and its predicate-argument structure. The argument with the role of purpose is ranked highest, and thus it determines the question focus.

With the exception of the focus, all arguments from the predicate-argument structure may be used for generating definition questions. The focus is elaborated upon. Several forms of elaborations are possible. One is a temporal one, as illustrated in Figure 1. Other are resultative, causative or manner-based. For example, the knowledge that assistance in a missile program results in an inven-

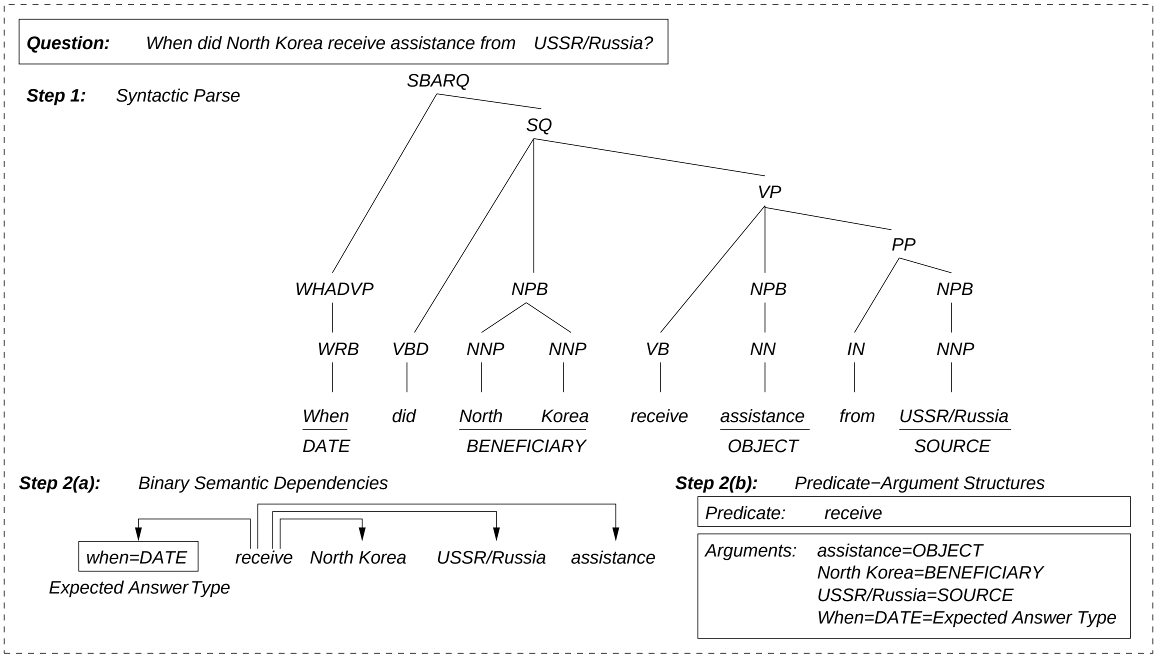


Figure 2: Deriving the Expected Answer Type

Question Pattern: *What kind of assistance has X received from Y for Z?*

Predicate−argument structure:

Predicate:	*receive*	
Arguments:	*Purpose: Z*	*Object: assistance*
	Beneficiary: X	*Source: Y*

Figure 3: Predicate-argument structure

tory of missiles allows for resultative elaboration. Further knowledge needs to be coerced for generating the implied questions as possible follow-ups to intended questions.

The relationship between intended questions and implied questions is marked by the presence of multiple references, e.g. the pronouns *it* and *this* or *any* and *ones*. The generation of implied questions is made possible by knowledge that is coerced from the intended questions. For example, when asking Q_1^i :*"What is the USSR/Russia?"* the coercion process abstracts away from the concept that needs to be defined, i.e. a country. The implied question requests confirmation of the metonymy resolution involving USSR/Russia.This named entity may represent a country but most likely it refers to its government or, as Q_2^{m1} suggests, organizations or individuals acting on behalf of the country. Both Q_1^{m1} and Q_2^{m1}, implied questions derived from the intended question Q_1^i, refer to the metonymy by using the pronouns *this* and *it* respectively. Different forms of coercion are used for Q_3^i because in this case the knowledge is associated with the predicate. The implied questions associated with the focus, i.e. the intended question Q_4^i, coerce the design and development predicates which are associated with the missiles as well as the timelines of possible additional assistance.

3 Models of Question Answering

The processing of questions is typically performed as a sequence of three processes: (1) Question Processing; (2) Document Processing and (3) Answer Extraction. In the case of factoid questions , question processing involves the classification of questions with the purpose of predicting what semantic class the answer should belong to. Thus we may have questions asking about PEOPLE, ORGANIZATIONS, TIME or LOCATIONS. Since open-domain Q/A systems process questions regardless of the domain of interest, question processing must be based on an extended ontology of answer types. The identification of the expected answer type is based either on binary semantic dependencies extracted from the syntactic parse of the question (Harabagiu et al., 2001) or on the predicate-argument structure of the question. In both cases, the relation to the question stem (i.e. what, who, when) enables the classification. Figure 2 illustrates a factoid question generated as an intended question and the derivation of its expected answer type.

However, many times the expected answer type needs to be identified from an ontology that has high lexico-semantic coverage. Many Q/A systems use the WordNet database for this purpose. In contrast, definition questions do not require the identification of the expected an-

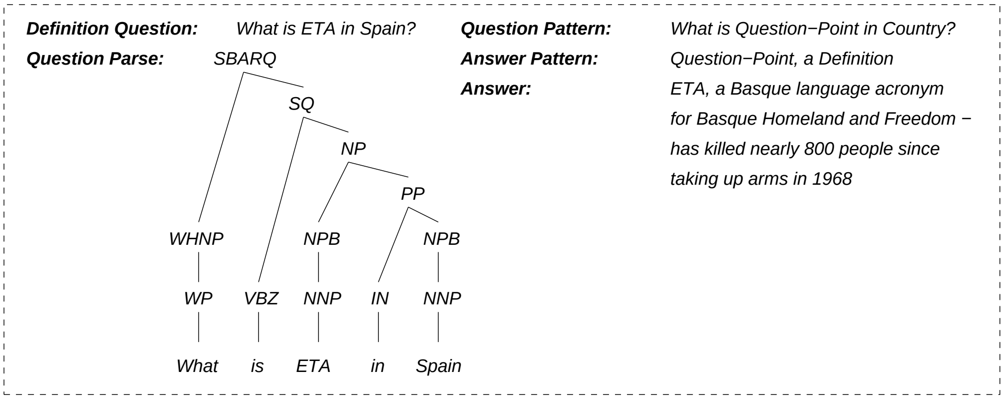

Definition Question: What is ETA in Spain?
Question Parse:
SBARQ
SQ
NP
PP
WHNP
NPB
NPB
WP
VBZ
NNP
IN
NNP
What
is
ETA
in
Spain
Question Pattern: What is Question−Point in Country?
Answer Pattern: Question−Point, a Definition
Answer: ETA, a Basque language acronym for Basque Homeland and Freedom − has killed nearly 800 people since taking up arms in 1968

Figure 4: Patterns for Processing Definition Questions

swer type, since they always request a definition. However, definition questions are matched against a set of patterns, which enables the extraction of the definition from the candidate answers. Figure 4 illustrates a definition question, the pattern it matched as well as the extracted answer.

Both factoid and definition questions can be answered only if candidate passages are available. The retrieval of these passages is made possible by keywords that are selected from the question words. The Documents Processing module implements a search engine that returns passages that are likely to contain the expected answer type in the case of factoid questions or the definition pattern in the case of definition questions. The answer extraction module optimizes the extraction of the correct answer by unifying the question information with the answer information. The unification may be based on pattern matching; on machine learning algorithms based on the question and answer features or on abductive reasoning that justifies the answer correctness.

Current state-of-the-art QA systems search for the candidate answer by assuming that the answers are single concepts, that can be recognized from a hierarchy or by a Named Entity Recognizer. This is a serious limitation, but it works well for the factoid, list or definition questions evaluated in TREC.

The three modules of current Q/A systems reflect the three functions that need to be considered by any Q/A model: (1) understanding what the question asks; (2) identify candidate text passage that might contain the answer; and (3) the extraction of the correct answer. Currently, the expected answer type represents what question asks about: a semantic concept, e.g. the name of a person, location or organization, kinds of diseases, types of animals or plants. Generally these semantic concepts are lexicalized in a single word or in 2-word collocations. Clearly, this represents a limitation, since often the questions ask for more than a single concept. As we have seen in Table1, there is additional intended and implied information that is requested. Therefore new models of Question/Answering need to incorporate these additional forms of knowledge.

When definition questions are processed in current Q/A systems, they are matched against a pattern, which is different from the question patterns associated with complex questions similar to those illustrated in Figure 1. In the case of a definition question like *"What is ETA in Spain?"*, the pattern identifies the question-point (QP) as *ETA*- the concept that needs to be defined and *Spain* as its *context*. The definition question pattern also contains several surface-form patterns that are matched in the candidate paragraphs. One such pattern is recognized in an apposition, by [QP, a AP] where AP represents the *answer phrase*. In the following passage:
"ETA, a Basque language acronym for Basque Homeland and Freedom - has killed nearly 800 people since taking up arms in 1968."
the exact answer representing the definition is identified in AP: *Basque language acronym for Basque Homeland and Freedom*. The fact that Basque country is a region in Spain allows a justification of the question *context*.

In this paper, by considering the intentional information and the implied information that can be derived when processing questions, we introduce a novel model of Q/A, which has access to rich semantic structures and enables the retrieval of more accurate answers as well as inference processes that explain the validity and contextual coverage of answers.

Figure 5 shows the structure of the novel model of Q/A we propose. Both Question Processing and Document Processing have the recognition of predicate-argument structures as a crux of their models. As reported in (Surdeanu et al., 2003), the recognition of predicate-argument structures depends on features made available by full syntactic parses and by Named Entity Recognizers. As we shall show in this paper, the predicate-argument structures enable the recognition of question pattern, the question focus and the intentional structure associated with

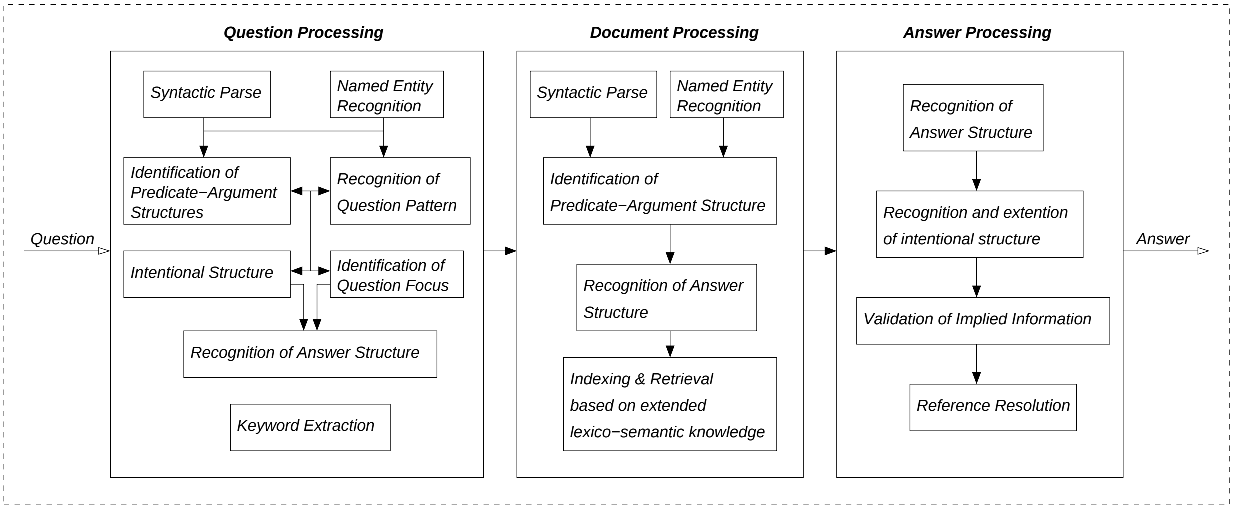

Question Processing
Document Processing
Answer Processing
Syntactic Parse
Named Entity Recognition
Identification of Predicate−Argument Structures
Recognition of Question Pattern
Intentional Structure
Identification of Question Focus
Recognition of Answer Structure
Keyword Extraction
Syntactic Parse
Named Entity Recognition
Identification of Predicate−Argument Structure
Recognition of Answer Structure
Indexing & Retrieval based on extended lexico−semantic knowledge
Recognition of Answer Structure
Recognition and extention of intentional structure
Validation of Implied Information
Reference Resolution
Question
Answer

Figure 5: Novel Question/Answering Architecture.

a question. When the intentions are known, the answer structure can be identified and the keywords extracted. For better retrieval of candidate answers, documents are indexed and retrieved based on the predicate-argument structures as well as on complex semantic structure associated with different question patterns. Similarly, the intentional structures are used for indexing/retrieving candidate passages. The Answer Processing function involves the recognition of the answer structure and intentional structure. Often this requires reference resolution. The implied information coerced from both the question and the candidate answer is also validated before deciding on the answer correctness.

4 Predicate-Argument Structures

To identify predicate-argument structures in questions and passages, we have: (1) used the Proposition Bank or PropBank as training data; and (2) a mode for predicting argument roles similar to the one employed by (Gildea and Jurafsky, 2002).

PropBank is a one million word corpus annotated with predicate-argument structures on top of the Penn Treebank 2 Wall Street Journal texts. For any given predicate, the expected arguments are labeled sequentially from Arg 0 to Arg 4. Generally, Arg 0 stands for *agent*, Arg 1 for *direct object* or *theme* or *patient*, Arg 2 for *indirect object* or *benefactive* or *instrument* or *attribute* or *end state*, Arg 3 for *start point* or *benefactive* or *attribute* and Arg4 for *end point*. In addition to these core arguments, adjunctative arguments are marked up. They include functional tags from Treebank, e.g. ArgM-DIR indicates a directional, ArgM-LOC indicates a locative, and ArgM-TMP stands for a temporal.
An example of PropBank markup is:

$[_{Arg_0^1}$ Analysts] have been $[_{predicate_1}$ expecting] $[_{Arg_1^1}$ a GM-Jaguar pact] that would $[_{predicate_2}$ give] $[_{Arg_2^2}$ the U.S. car maker] $[_{Arg_1^2}$ an eventual 30% state in the British Company].

The model of identifying the arguments of each predicate consists of two tasks: (1) the recognition of the boundaries of each argument in the syntactic parse tree; (2) the identification of the argument role. Each task can be cast as a separate classifier. Next section describes our approach based on Support Vector Machines (SVM) (Vapnik, 1995).

4.1 Automatic Predicate-Argument extraction

Given a sentence in natural language, all the predicates associated with its verbs have to be identified along with their arguments. This problem can be divided in two subtasks: (a) detection of the target argument boundaries, i.e. all its compounding words, and (b) classification of the argument type, e.g. *Arg0* or *ArgM*.

A direct approach to learn both detection and classification of predicate arguments is summarized by the following steps:

1. Given a sentence from the *training-set*, generate a full syntactic parse-tree;

2. let $\mathcal{P}$ and $\mathcal{A}$ be the set of predicates and the set of parse-tree nodes (i.e. the potential arguments), respectively;

3. for each pair $<p, a> \in \mathcal{P} \times \mathcal{A}$:

   - extract the feature representation set, $F_{p,a}$;
   - if the subtree rooted in a covers exactly the words of one argument of p, put $F_{p,a}$ in T^+ (positive examples), otherwise put it in T^- (negative examples).

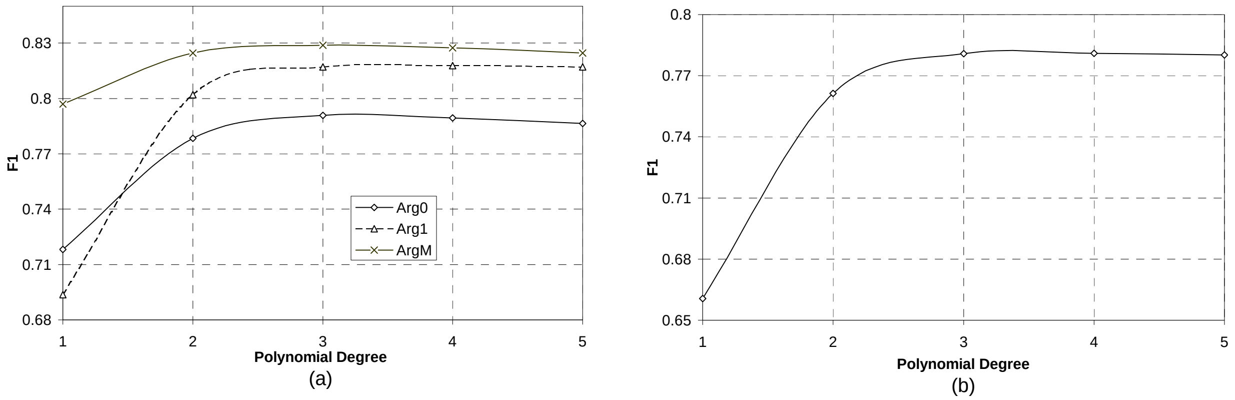


Figure 6: Single classifiers and Multi-classifier performance for argument extraction.

The above T^+ and T^- sets can be re-organized as positive $T^+_{arg_i}$ and negative $T^-_{arg_i}$ examples for each argument i. In this way, an individual ONE-vs-ALL SVM classifier for each argument i can be trained. We adopted this solution as it is simple and effective (Pradhan et al., 2003). In the classification phase, given a sentence of the *test-set*, all its $F_{p,a}$ are generated and classified by each individual SVM classifier. As a final decision, we select the argument associated with the maximum value among the scores provided by the SVMs[2], i.e. $argmax_{i \in S} \quad C_i$, where S is the target set of arguments.

The discovering of relevant features is a complex task. Nevertheless there is a common consensus on the basic features that should be adopted. These standard features, first proposed in (Gildea and Jurafsky, 2002), are derived from parse trees as illustrated by Table 2.

4.2 Parsing Sentence into Predicate Argument Structures

For the experiments, we used PropBank (`www.cis.upenn.edu/∼ace`) along with Penn-TreeBank[3] 2 (`www.cis.upenn.edu/∼treebank`) (Echihabi and Marcu, 2003). This corpus contains about 53,700 sentences and a fixed split between training and testing which has been used in other researches (Gildea and Jurafsky, 2002; Surdeanu et al., 2003; Hacioglu et al., 2003; Chen and Rambow, 2003; Gildea and Hockenmaier, 2003; Gildea and Palmer, 2002; Pradhan et al., 2003). In this split, Sections from 02 to 21 are used for training, section 23 for testing and sections 1 and 22 as developing set. We considered all PropBank arguments from *Arg0* to *Arg9*, *ArgA* and *ArgM* even if only *Arg0* from *Arg4* and *ArgM* contain enough training/testing data to affect the global performance.

The classifier evaluations were carried out using the SVM-light software (Joachims, 1999) available at `http://svmlight.joachims.org/` with the default polynomial kernel according to a degree $d \in \{1, 2, 3, 4, 5\}$. The performances were evaluated using the F_1 measure for both single argument classifiers and the multi-class classifier.

- PHRASE TYPE (pt): This feature indicates the syntactic type of the phrase labeled as a predicate argument.
- PARSE TREE PATH (path): This feature contains the path in the parse tree between the predicate phrase and the argument phrase, expressed as a sequence of nonterminal labels linked by direction (up or down).
- POSITION (pos) Indicates if the constituent appears before or after the predicate in the sentence.
- VOICE (voice) This feature distinguishes between active or passive voice for the predicate phrase.
- HEAD WORD (hw) This feature contains the head word of the evaluated phrase. Case and morphological information are preserved.
- GOVERNING CATEGORY (gov) This feature applies to noun phrases only, and it indicates if the NP is dominated by a sentence phrase (typical for subject arguments with active voice predicates), or by a verb phrase (typical for object arguments).
- PREDICATE WORD In our implementation this feature consists of two components: (1) VERB: the word itself with the case and morphological information preserved; and (2) LEMMA which represents the verb normalized to lower case and infinitive form.

Table 2: Standard Features used in Predicate Argument Extraction.

Figure 6 illustrates the F_1 measures for the overall argument extraction task (i.e. identification and classification) according to different polynomial degrees. Figure 6(a) illustrates the F_1-performance of single classifiers

[2] This is a basic method to pass from binary categorization into a multi-class categorization problem; several optimization have been proposed, e.g. (Goh et al., 2001).

[3] We point out that we removed from the Penn TreeBank the special tags of noun phrases like *Subj* and *TMP* as parsers usually are not able to provide this information.

for the arguments Arg0, Arg1 and ArgM. Figure 6(b) illustrates the performance for all the arguments (i.e. the multi-classifier). In general, we were able to recognize predicate argument structures with an F_1-score of 80%.

4.3 Using Predicate-Argument Structures in Question Answering.

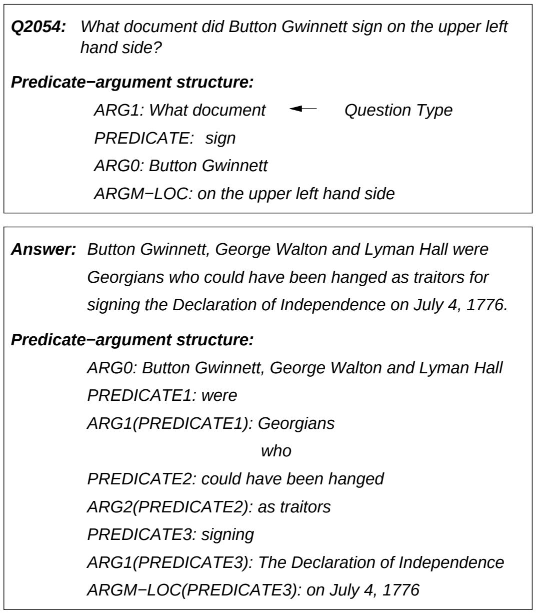
Predicate-argument structures are useful for identifying candidate answers. Since they recognize long-distance dependencies between a predicate and one its arguments, they enable (1) the identification of the exact boundaries of an answer; and (2) they unify the predicate-argument relation sought by question with those recognized in candidate passages.

Moreover, they are very useful in situations when the expected answer type of the question could not be recognized. There are two causes when the expected answer type cannot be identified:
Case1: the answer class is a name that cannot be correctly classified by an available Named Entity Recognizer, because its class name is not encoded.
Case2: the answer class cannot be found in the Answer Type hierarchy. The example from Figure 7 shows an instance of case 1. In this figure, the TREC question Q2054 has a predicate that can be unified with PREDICATES from the answer passage. The Arg1 of the predicate is the expected answer, which is identified as ”the Declaration of Independence”. The Arg0 in the question is Button Gwinnett, whereas in the answer, it is underspecified, and should be resolved to who. This relative pronoun has Button Gwinnett as one of its antecedents.

In Figure 8 the second case is illustrated. The question asked about the first argument of the predicate ”measure”, when its Arg2 = ”a theodolite”. In the answer, Predicate 2, with its infinite form, has as Arg 2 the same ”theodolite”. However, the predicates are lexicalized by different verbs. In WordNet, the first sense of the verb ”measure” as the verb ”determine” as a hypernym, therefore Arg1 = ”wind speeds” is the correct answer.

5 Intentional Structures

The correct interpretation of many questions requires the inference of implicit information, that is not directly stated in the question, but merely implied. The mechanisms of recognizing the intentions of the questioner are helpful means of identifying the implied information. For example, in the question Q^I:*“Will Prime Minister Mori survive the crisis?”*, the user does not literally mean *“Will Prime Minister Mori be still alive when the political crisis is over ?”*, but rather (s)he implies her/his belief that the current political crisis might cost the Japanese Prime Minister his job. It is very unlikely that any expert knowledge base covering Japanese politics will encode knowledge covering all situations of political crisis and the possible outcomes of the prime minister. However, this pragmatic knowledge is essential for the correct interpretation of the question.



Q2054: *What document did Button Gwinnett sign on the upper left hand side?*

Predicate−argument structure:

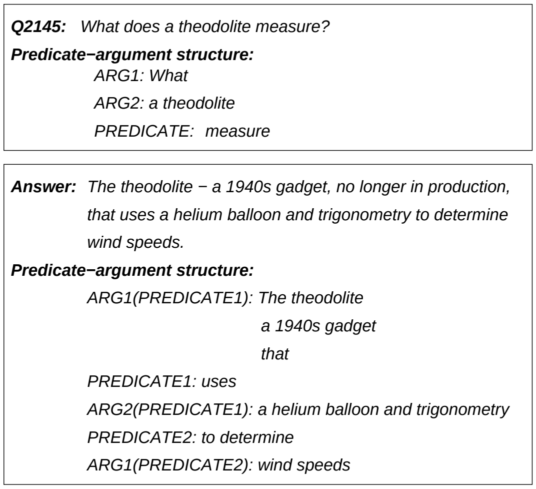
ARG1: What document ← *Question Type*
PREDICATE: sign
ARG0: Button Gwinnett
ARGM−LOC: on the upper left hand side

Answer: *Button Gwinnett, George Walton and Lyman Hall were Georgians who could have been hanged as traitors for signing the Declaration of Independence on July 4, 1776.*

Predicate−argument structure:

ARG0: Button Gwinnett, George Walton and Lyman Hall
PREDICATE1: were
ARG1(PREDICATE1): Georgians
who
PREDICATE2: could have been hanged
ARG2(PREDICATE2): as traitors
PREDICATE3: signing
ARG1(PREDICATE3): The Declaration of Independence
ARGM−LOC(PREDICATE3): on July 4, 1776

Figure 7: Answer extraction from predicate-argument structures: Case1



Q2145: *What does a theodolite measure?*

Predicate−argument structure:

ARG1: What
ARG2: a theodolite
PREDICATE: measure

Answer: *The theodolite − a 1940s gadget, no longer in production, that uses a helium balloon and trigonometry to determine wind speeds.*

Predicate−argument structure:

ARG1(PREDICATE1): The theodolite
a 1940s gadget
that
PREDICATE1: uses
ARG2(PREDICATE1): a helium balloon and trigonometry
PREDICATE2: to determine
ARG1(PREDICATE2): wind speeds

Figure 8: Answer extraction from predicate-argument structures: Case 2

The design of advanced Question&Answering systems capable of grasping the intention of a professional analyst when (s)he poses a question depends both on the knowledge of the domain referred by the question as well as on a variety of rules and conventions that allow the communication of intentions and beliefs in addition to the literary meaning of the question. Access to domain knowledge is granted by a combination of retrieval mechanisms that bring forward relevant document passages from unstructured collections of documents, specialized knowledge

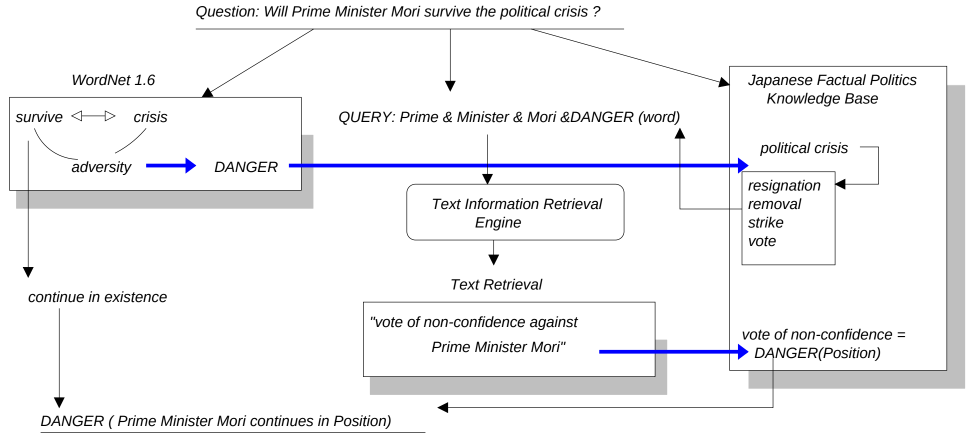

Question: Will Prime Minister Mori survive the political crisis ?
WordNet 1.6
survive
crisis
adversity
DANGER
QUERY: Prime & Minister & Mori &DANGER (word)
Japanese Factual Politics
Knowledge Base
political crisis
resignation
removal
strike
vote
Text Information Retrieval
Engine
Text Retrieval
continue in existence
"vote of non-confidence against
Prime Minister Mori"
vote of non-confidence =
DANGER(Position)
DANGER (Prime Minister Mori continues in Position)

Figure 9: Intentional Structure derived from Lexico-Semantic Knowledge.

bases and/or database access mechanisms. The research proposed in this project focuses on the derivation and usage of pragmatic knowledge that supports the recognition of question implications, also known as implicatures (cf. (Grice, 1975b)).

5.1 Intentional structures Derived from Lexico-Semantic Knowledge

The novel idea of this research is to link computational implicatures, similar to those defined by Grice (Grice, 1975b), to inferences that can be drawn from general lexico-semantic knowledge bases such as WordNet of FrameNet. Incipient work was described in (Sanda Harabagiu and Yukawa, 1996), where a method of using lexico-semantic path for recognizing textual implicatures was presented. To our knowledge, this is the only computational model of implicatures that was developed and tested on a large lexico-semantic knowledge base (e.g. WordNet), enabling successful recognition of implicatures.

The model proposed in (Sanda Harabagiu and Yukawa, 1996) uncovered a relationship between (a) the *coherence* of a text segment; (b) its *cohesion* expressed by the lexical paths and (c) the implicatures that can be drawn, mostly to account for *pragmatic* knowledge. This relationship can be extended across documents and across topics, to learn patterns of textual and Q&A implicatures and the methods of deriving knowledge that enables their recognition.

The derivation of pragmatic knowledge combines information from three different sources:
(1) lexical knowledge bases (e.g. WordNet),
(2) expert knowledge bases that can be rapidly formatted for many domains (e.g. Japanese political knowledge); and
(3) knowledge supported from the textual information available from documents. The methodology of combining these three sources of information is novel.

For question Q^I, the starting point is the concept identified as a cue for the expected answer type through methods described in (Harabagiu et al., 2000). This concept is lexicalized by the verb-object pair *survive-crisis*. Verb survive has four distinct senses in the WordNet 1.6 database, whereas noun crisis has two senses. The polysemy of the expected answer type increases the difficulty of the derivation of pragmatic knowledge, but it does not presupposes the word sense disambiguation of the expression. The information available in the glosses defining the WordNet synsets provides helpful information for expanding the multi-word term defining the expected answer type. By measuring the similarity between the two senses of the noun crisis and the words encountered as objects or prepositional attachments in the glosses of the various senses of the verb survive, we distinguish the noun *adversity* and the example *cancer* as expressing the closest semantic orientation to the first sense of noun crisis. The similarity is measured by counting the number of common hypernyms and gloss concepts of hypernyms of two synsets. Figure 9 illustrates the concepts related to the question Q^I, as derived from WordNet lexico-semantic knowledge base.

The fact that surviving a political crisis has a dangerous component, indicated by the noun adversity, may also be supported by inferences drawn from an expert knowledge base, showing that a political crisis may be dangerous for political figures in power. However, at this point, the object of the dangerous situation is not specified. But several concepts indicating dangerous political situations can be inferred from the expert knowledge base and used in the query for text evidence. Only when text passages involving Prime Minister Mori are retrieved, clarifications of the situation are brought to attention: a vote of non-confidence against the prime minister is considered. This new information helps inference from the expert knowledge base. The expert knowledge base modeling the

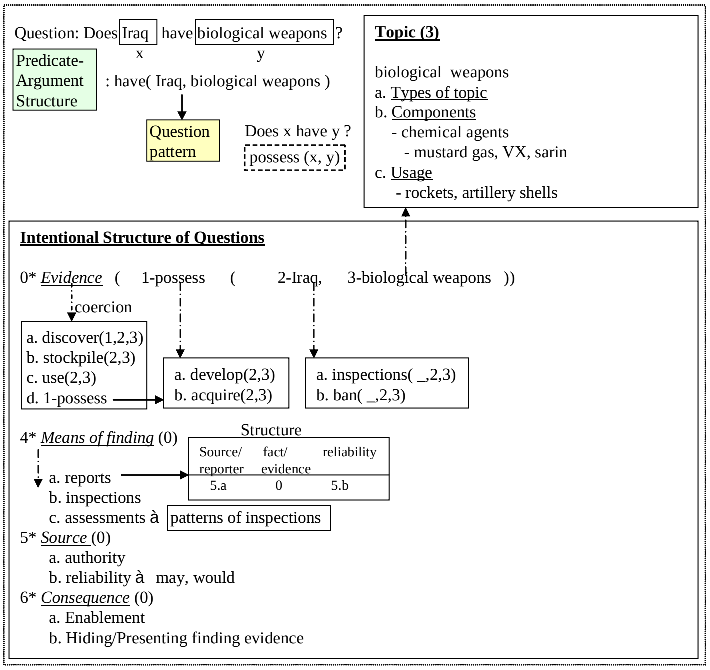


Figure 10: Intentional structure derived from predicate-argument structures.

Japanese factional politics confirms that this is a dangerous situation for the Prime Minister and that in fact his position is in jeopardy. Due to this inference from the expert knowledge base, the concept POSITION replaces noun *existence* from the gloss of the second sense of verb survive, and the pragmatic knowledge required for the interpretation of the implicature is assembled:

The interactions between the three information sources derives the pragmatic knowledge on which relies the implication of the question. The user had an inherent belief that Prime Minister Mori might be replaced, and (s)he queries the Q&A system not only to find information but also to find support for his/her belief. The intentional structure is represented as a set of concepts and the relations that span them, as illustrated in Figure 9.

5.2 Coercion of Intentions

A second method of deriving the intentional structure of a question is based on the predicate-argument structure that is derived from the question and the candidate answers.

Figure 10 illustrates the Intentional Structure of one such question. The structure of the intentions is determined by the predicate-argument structure of the question and by its pattern. Generally, when asking whether X posses Y, we want to find (1) evidence of this fact; (2) we explore different means of finding the information; (3) we are interested in the source of information and (4) the enablers or inhibitors of finding the information as well as the consequences of knowing it are of interest. We assign a different index to each object from the predicate-argument structure, and do the same for each element of the intentional structure. For instance, in Figure 2, source(0) is interpreted as source(index=0) = source(evidence). Another feature of the intentional structure is determined by the coercions that are associated with both forms of indexed objects. For example, the coercion of evidence shows the most typical ways of finding evidence in the context of the topic of the question. Figure 2 lists such possibilities as (a) discovering, (b) stockpiling, (c) using and even (d) possessing. These possibilities are inserted in the context of the topic, since they make use of the indexes for associating meaning to their representations. In fact, option (a) discover(1,2,3) reads as discover(index=1, index=2, index=3) =discover(possesses(Iraq, biological weapons)). Whereas option (b) stockpile(2,3) can be similarly interpreted as stockpile(Iraq, biological weapons). Note that one of the indexed objects is the topic. The structure of the topic is define along three semantic dimensions: (1) hyponyms or examples of other types of the same category as the topic; (2) the meronyms or components; and (3) the functionality or the usage. The derivation of such a large set of intentional structures helped us learn how to coerce pragmatic knowledge. We have developed a probabilistic approach extending the metonymy work of (Lapata and Lascarides, 2003).

Lapata and Lascarides report a model of interpretation of verbal metonymy as the point distribution $P(e, o, v)$ of three variables: the metonymy verb v, its object, and the sought after interpretation i. For example a verb → object relation that needs to be metonymycally interpreted, is $enjoy \rightarrow movie$. In this case v = $enjoy$, o = $movie$ and $i \in \{making, watching, directing\}$. The variables of the distribution re ordered as $<i, v, o>$ to help factoring $P(i, v, o) = P(i) \cdot P(v|i) \cdot P(o|i, v)$. Each of the probabilities $P(i)$, $P(v|i)$ and $P(o|i, v)$ can be estimated using maximum likelihood. As it is illustrated in Figure 10, we have extended this model to account for: (1) coercion of topic information; (2) coercion of evidence of a fact; (3) interpretation of predicate and (4) interpretation of arguments. Since the verb → object relation translates in one of the predicate-argument relations, we have coerced the predicate interpretations in the same way as (Lapata and Lascarides, 2003), but we allowed for any predicate-argument relation. Argument coercions were produced by searching the most likely predicates that used the same arguments. The topic model also incorporated topic signatures, similar to these reported in (E.H. Hovy and Ravichandran, 2002).

6 Conclusions

In this paper we have described the problem of interpreting the question intentions and proposed two methods of generating the intentional structure of questions. The first method is based on lexico-semantic chains between concepts that are related to the question. The second method generates intentional structures by using the predicate-argument structures of questions and the topic representation of questions. To derive both forms of intentional structures, we have relied on information available from WordNet and on the parsing of questions and answers in predicate-argument structures. Our experiments show that the intentional structure may determine a different interpretation of the question, and thus different keywords can be used to retrieve the answers. Answer extraction also depends on the semantic relations between the coerced interpretations of predicates and arguments. By selecting a set of 100 questions for test, we have evaluated the correctness of the extracted answers when (1) no intentional knowledge was coerced; (2) implicatures were derived from lexico-semantic knowledge and (3) intentional structures were derived based on predicate-argument structures. An increase of 8structures and one of 22the impact of each element of the intentional structure on the Q/A processing.

References

John Chen and Owen Rambow. 2003. Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the ACL, Sapporo, Japan.*

Chin-Yew Lin E.H. Hovy, U. Hermjakob and Deepak Ravichandran. 2002. Using knowledge to facilitate pinpointing of factoid answers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002).*

Daniel Gildea and Julia Hockenmaier. 2003. Identifying semantic roles using combinatory categorial grammar. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistic*, 28(3):254–288.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02), Philadelphia, PA.*

King-Shy Goh, Edward Chang, and Kwang-Ting Cheng. 2001. SVM binary classifier ensembles for image classification. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 395–402.

Paul H. Grice. 1975a. Logic and conversation. In P. Cole and New York J.L. Morgan (ed.), Academic Press, editors, *Syntax and Sematics Vol.3:Speech Acts*, pages 41–58.

Paul J. Grice. 1975b. *Syntax and Semantics Vol.3:Speech Acts*. P. Cole and J. Morgan, editors.

Kadri Hacioglu, Sameer Pradhan, Wayne Ward, Jim Martin, and Dan Jurafsky. 2003. Shallow semantic parsing using support vector machines. Technical report.

Sanda Harabagiu, Marius Paşca, and Steven Maiorano. 2000. Experiments with open-domain textual question answering. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000), pages 292–298, Saarbrucken, Germany,.*

Sanda M. Harabagiu, Dan I. Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2001. The role of lexico-semantic feedback in open-domain textual question-answering. In *Meeting of the ACL*, pages 274–281.

S. Harabagiu, D. Moldovan, C. Clark, M. Bodwen, J. Williams, and J. Bensley. 2003. Answer mining by combining extraction techniques with abductive reasoning. In *Notebook of the Twelveth Text REtrieval Converence (TREC-2003)*, pages 46–53.

A. Ittycheriah, M. Franz, W. Zhu, and A. Ratnaparkhi. 2000. IBM's statistical question answering system. In *Proceedings of the 9th Text REtrieval Conference, Gaithersburg, MD.*

T. Joachims. 1999. T. Joachims, Making large-Scale SVM Learning Practical. In B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press., editor, *Advances in Kernel Methods - Support Vector Learning*.

Maria Lapata and Alex Lascarides. 2003. A probabilistic account of logical metonymy. *Computational Linguistics*, 29:2:263–317.

Wendy Lehnert. 1978. The process of question answering. In *Lawrence Erlbaum Assoc., Hillsdale*.

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM-2003)*.

Dan Moldovan Sanda Harabagiu and Takashi Yukawa. 1996. Testing gricean constraints on a wordnet-based coherence evaluation system. In *Working Notes of the AAAI-96 Spring Symposium on Computational Implicature, Stanford, CA*.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41th Annual Conference of the Association for Computational Linguistics (ACL-03)*, pages 8–15.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.