Handling Biographical Questions with Implicature

Donghui Feng
Information Sciences Institute
University of Southern California
Marina del Rey, CA, 90292
donghui@isi.edu

Eduard Hovy
Information Sciences Institute
University of Southern California
Marina del Rey, CA, 90292
hovy@isi.edu

Abstract

Traditional question answering systems adopt the following framework: parsing questions, searching for relevant documents, and identifying/generating answers. However, this framework does not work well for questions with hidden assumptions and implicatures. In this paper, we describe a novel idea, a cascading guidance strategy, which can not only identify potential traps in questions but further guide the answer extraction procedure by recognizing whether there are multiple answers for a question. This is the first attempt to solve implicature problem for complex QA in a cascading fashion using N-gram language models as features. We here investigate questions with implicatures related to biography facts in a web-based QA system, PowerBio. We compare the performances of Decision Tree, Naïve Bayes, SVM (Support Vector Machine), and ME (Maximum Entropy) classification methods. The integration of the cascading guidance strategy can help extract answers for questions with implicatures and produce satisfactory results in our experiments.

1 Motivation

Question Answering has emerged as a key area in natural language processing (NLP) to apply question parsing, information extraction, summarization, and language generation techniques (Clark et al., 2004; Fleischman et al., 2003; Echihabi et al., 2003; Yang et al., 2003; Hermjakob et al., 2002; Dumais et al., 2002). Traditional question answering systems adopt the framework of parsing questions, searching for relevant documents, and then pinpointing and generating answers. However, this framework includes potential dangers. For example, to answer the question "when did Beethoven get married?", a typical QA system would identify the question target to be a "Date" and would apply techniques to identify the date Beethoven got married. Since Beethoven never married, this direct approach is likely to deliver wrong answers. The trick in the question is the implicature that Beethoven got married. In the main task of QA track of TREC 2003, the performances of most systems on providing "NIL" when no answer is possible range from only 10% to 30% (Voorhees, 2003).

Just as some questions have no answer, others may have multiple answers. For instance, with "who was Ronald Reagan's wife?", a QA system may give only "Nancy Davis" as the answer. However, there is another correct answer: Jane Wyman. The problem here is the implicature in the question that Reagan only got married once.

An implicature is anything that is inferred from an utterance but that is not a condition for the truth of the utterance (Gazdar, 1979; Levinson, 1983). Implicatures in questions either waste computational effort or impair the performance of a QA system or both. Therefore, when answering questions, it is prudent to identify the questions with implicatures before processing starts.

In this paper, we describe a novel idea to solve the problem: a strategy of cascading guidance. This is the first attempt to solve implicature problem for complex QA in a cascading fashion using N-gram

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 596–603, Vancouver, October 2005. ©2005 Association for Computational Linguistics

language models as features. The cascading guidance part is designed to be inserted immediately before the search procedure to handle questions with implicatures. It can not only first identify the potential "no answer" traps but also identify whether multiple answers for this question are likely.

To investigate the performance of the cascading guidance strategy, we here study two types of questions related to biography facts in a web-based biography QA system, PowerBio. This web-based QA system extracts biographical facts from the web obtained by querying a web search engine (Google in our case). Figure 1 provides the two types of questions we selected, which we refer to as SPOUSE_QUESTION and CHILD_QUESTION.

I. SPOUSE_QUESTION
E.g. Who is <PERSON>'s wife?
Who is <PERSON>'s husband?
Whom did <PERSON> marry?
…
II. CHILD_QUESTION
E.g. Who is <PERSON>'s son?
Who is <PERSON>'s daughter?
Who is <PERSON>'s child?
…

Figure 1. SPOUSE_QUESTION and CHILD_QUESTION

Both types of questions have implicatures to justify the use of the cascading guidance strategy. Intuitively, to answer these questions, we have two issues related to implicatures to clarify:

- Does the person have a spouse/child?
- What's the number of answers for this question? (One or many?)

We therefore create two successive classification engines in the cascading classifier.

For learning, our approach queries the search engine with every person listed in the training set, extracts related features from the documents, and trains the cascading classifiers. For application, when a new question is given, the cascading classifier is applied before activation of the search subsystem. We compare the performances of four popular classification approaches in the cascading classifier, namely Decision Tree, Naïve Bayes, SVM (Support Vector Machine), and ME (Maximum Entropy) classifications.

The paper is structured as follows: related work is discussed in Section 2. We introduce our cascading guidance technique in Section 3, including Decision Tree, Naïve Bayes and SVM (Support Vector Machine) and ME (Maximum Entropy) classifications. The experimental results are presented in Section 4. We discuss related issues and future work in Section 5.

2 Related Work

Question Answering has attracted much attention from the areas of Natural Language Processing, Information Retrieval and Data Mining (Fleischman et al., 2003; Echihabi et al., 2003; Yang et al., 2003; Hermjakob et al., 2002; Dumais et al., 2002; Hermjakob et al., 2000). It is tested in several venues, including the TREC and CLEF Question Answering tracks (Voorhees, 2003; Magnini et al., 2003). Most research efforts in the Question Answering community have focused on factoid questions and successful Question Answering systems tend to have similar underlying pipelines structures (Prager et al., 2004; Xu et al., 2003; Hovy et al., 2000; Moldovan et al., 2000).

Recently more techniques for answer extraction, answer selection, and answer validation have been proposed (Lita et al., 2004; Soricut and Brill, 2004; Clark et al., 2004).

Prager et al. (2004) proposed applying constraint satisfaction obtained by asking auxiliary questions to improve system performance. This approach requires the creation of auxiliary questions, which may be complex to automate.

Ravichandran and Hovy (2002) proposed automatically learning surface text patterns for answer extraction. However, this approach will not work if no explicit answers exist in the source. The first reason is that in that situation the anchors to learn the patterns cannot be determined. Secondly, most of the facts without explicit values are not expressed with long patterns including anchors. For example, the phrase "the childless marriage" gives enough information that a person has no child. But it is almost impossible to learn such surface text patterns following (Ravichandran and Hovy, 2002).

Reported work on question processing focuses mainly on the problems of parsing questions, determining the question target for search subsystem

(Pasca and Harabagiu, 2001; Hermjakob et al., 2000). Saquete et al. (2004) decompose complex temporal questions into simpler ones based on the temporal relationships in the question.

To date, there has been little published work on handling implicatures in questions. Just-In-Time Information Seeking Agents (JITISA) was proposed by Harabagiu (2001) to process questions in dialogue and implicatures. The agents are created based on pragmatic knowledge. Traditional answer extraction and answer fusion approaches assume the question is always correct and explicit answers do exist in the corpus. Reported work attempts to rank the candidate answer list to boost the correct one into top position. This is not enough when there may not be an answer for the question posed.

For biographical fact extraction and generation, Zhou et al. (2004) and Schiffman et al. (2001) use summarization techniques to generate human biographies. Mann and Yarowsky (2005) propose fusing the extracted information across documents to return a consensus answer. In their approach, they did not consider multiple values or no values for biography facts, although multiple facts are common for some biography attributes, such as multiple occupations, children, books, places of residence, etc. In these cases a consensus answer is not adequate.

Our work differs from theirs because we are not only working on information/answer extraction; the focus in this paper is the guidance for answer extraction of questions (or IE task for values) with implicatures. This work can be of great help for immediate biographical information extraction.

We describe details of the cascading guidance technique and investigate how it will help for question answering in Section 3.

3 Cascading Guidance Technique

We turn to the Web by querying a web search engine (Google in our case) to find evidence to create guidance for answer extraction.

3.1 Classification Procedure

The cascading classifier is applied after the name of the person and the answer types are identified. Figure 2 gives the pipeline of the classification procedure.

With the identified person name, we query the search engine (Google) to obtain the top N web pages/documents. A simple data cleaning program only keeps the content texts in the web page, which is broken up into separate sentences. Following that, topic sentences are identified with the keyword topic identification technique. For each topic we provide a list of possible related keywords and any sentences containing both the person's name (or reference) and at least one of the keywords will be selected. The required features are extracted from the topic sentences and passed to the cascading classifier as supporting evidence to generate guidance for answer extraction.

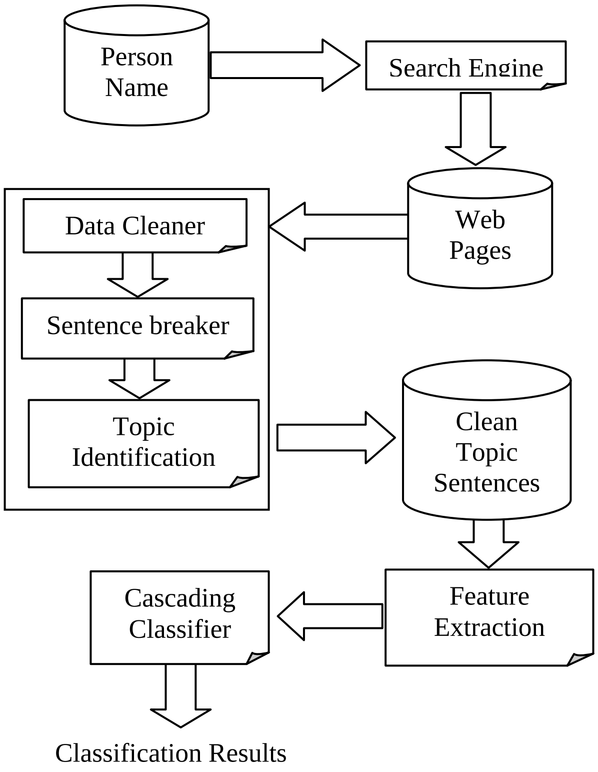

Figure 2. Procedure of Cascading Classifier

3.2 Feature Extraction

Intuitively, sentences elaborating a biographical fact in a given topic should have similar styles (short patterns) of organizing words and phrases. Here, topic means an aspect of biographical facts, e.g., marriage, children, birthplace, and so on. Inspired by this, we consider taking N-grams in sentences as our features. However, N-gram features not closely related to the topic will bring more noise into the system. Therefore, we only take the N-grams within a fixed-length window around the topic keywords for features calculation, and pass them as evidence to cascading classifier.

For N-grams, instead of using the multiplication of conditional probabilities of each word in the N-gram, we only consider the last conditional probability (see below). The reason is that the last conditional probability is a strong sign of the pattern's importance and how this sequence of words is organized. Simply multiplying all the conditional probabilities will decrease the value and require normalization. Realizing that in a set of documents the frequency of each N-gram is very important information, we combine the last conditional probability with the frequency.

The computation for each feature of unigram, bigram and trigram are defined as the following formulas:

$$f_{unigram} = p(w_i) * freq(w_i) \quad (1)$$

$$f_{bigram} = p(w_i \mid w_{i-1}) * freq(w_{i-1}, w_i) \quad (2)$$

$$f_{trigram} = p(w_i \mid w_{i-2}, w_{i-1}) * freq(w_{i-2}, w_{i-1}, w_i) \quad (3)$$

We here investigate four kinds of classifiers, namely Decision Tree, Naïve Bayes, Support Vector Machine (SVM), and Maximum Entropy (ME).

3.3 Classification Approaches

The cascading classifier is composed of two successive parts. Given the set of extracted features, the classification result could lead to different responses to the question, either answering with "no value" with strong confidence or directing the answer extraction model how many answers should be sought.

For text classification, there are several well-studied classifiers in the machine learning and natural language processing communities.

Decision Tree Classification

The Decision Tree classifier is simple and matches human intuitions perfectly while it has been proved efficient in many application systems. The basic idea is to break up the classification decision into a union of a set of simpler decisions based on N-gram features. Due to the large feature set, we use C5.0, the decision tree software package developed by RuleQuest Research (Quinlan, 1993), instead of C4.5.

Naïve Bayes Classification

The Naïve Bayes classifier utilizes Bayes' rule as follows. Supposing we have the feature set $F = \{f_1, f_2, ..., f_n\}$, the probability that person p belongs to a class c is given as:

$$c = \arg\max_{c'} P(c' \mid F) \quad (4)$$

Based on Bayes' rule, we have

$$\begin{aligned} c &= \arg\max_{c'} P(c' \mid F) \\ &= \arg\max_{c'} \frac{P(F \mid c')P(c')}{P(F)} \quad (5) \\ &= \arg\max_{c'} P(F \mid c')P(c') \end{aligned}$$

This was used for both successive classifiers of the cascading engine.

SVM Classification

SVM (Support Vector Machines) has attracted much attention since it was introduced in (Boser et al., 1992). As a special and effective approach for kernel based methods, SVM creates non-linear classifiers by applying the kernel trick to maximum-margin hyperplanes.

Suppose $p_i, i = 1,...,n$ represent the training set of persons, and the classes for classifications are $C = \{c_1, c_2\}$ (for simplicity, we represent the classes with $C = \{-1,1\}$). Then the classification task requires the solution of the following optimization problem (Hsu et al., 2003):

$$\begin{aligned} &\min_{\omega,b,\xi} \ \frac{1}{2}\omega^T\omega + M\sum_{i=1}^{n}\xi_i \\ &subject\ \ to\ \ c_i(\omega^T\phi(p_i) + b) \geq 1 - \xi_i \quad (6) \\ &\xi_i \geq 0 \end{aligned}$$

We use the SVM classification package LIBSVM (Chang and Lin, 2001) in our problem.

ME Classification

ME (Maximum Entropy) classification is used here to directly estimate the posterior probability for classification.

Suppose p represents the person and the classes for classifications are $C = \{c_1, c_2\}$, we have M feature functions $h_m(c, p), m = 1,..., M$. For each feature function, we have a model parameter $\lambda_m, m = 1,..., M$. The classification with maximum likelihood estimation can be defined as follows (Och and Ney, 2002):

$$
\begin{aligned}
P(c \mid p) &= p_{\lambda_1^M}(c \mid p) \\
&= \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(c, p)]}{\sum_{c'} \exp[\sum_{m=1}^{M} \lambda_m h_m(c', p)]} \qquad (7)
\end{aligned}
$$

The decision rule to choose the most probable class is (Och and Ney, 2002):

$$
\begin{aligned}
\hat{c} &= \arg\max_{c} \{P(c \mid p)\} \\
&= \arg\max_{c} \left\{ \sum_{m=1}^{M} \lambda_m h_m(c, p) \right\} \qquad (8)
\end{aligned}
$$

We use the published package YASMET[1] to conduct parameters training and classification. YASMET requires supervised learning for the training of maximum entropy model.

The four classification approaches are assembled in a cascading fashion. We discuss their performance next.

4 Experiments and Results

4.1 Experimental Setup

We download from infoplease.com[2] and biography.com[3] two corpora of people's biographies, which include 24,975 and 24,345 bios respectively. We scan each whole corpus and extract people having spouse information. To create the data set, we manually check and categorize each person as having multiple spouses, only one spouse, or no spouse. Similarly, we obtained another list of persons having multiple children, only one child, and no child. The sizes of data extracted are given in Table 1.

Type	Child	Spouse
No_value	25	20
One_value	35	32
Multiple_values	107	43

Table 1. Extracted experimental data

For the cascading classification, in the first step, when classifying whether a person has a spouse/child or not, we merge the last two subsets with one value and multiple values into one. Table 2 presents the data used for each level of classification.

	class	Child	Spouse
First-level Classification	No_value	25	20
	With_value	142	75
Second-level Classification	One_value	35	32
	Multiple_value	107	43

Table 2. Data set used for classification

To investigate the performances of our cascading classifiers, we divided the two sets into training set and testing set, with half of them in the training set and half in the testing set.

4.2 Empirical Results

For each situation of the two questions, when the answer type has been determined to be the child or spouse of a person, we send the person's name to Google and collect the top N documents. As described in Figure 2, topic sentences in each document are selected by keyword matching. A window with the length of *w* is applied to the sentence. All word sequences in the window are selected for feature calculation. We take all the three N-gram language models (unigram, bigram, and trigram) in the window for feature computation. Table 3 gives the sizes of the bigram feature sets for first-level classification as we take more and more documents into the system.

Top N Docs	Child	Spouse
1	3468	1958
10	27733	12325
20	46431	27331
30	61057	36637
40	76687	43771
50	87020	50868
60	96393	61632
70	108053	67712
80	118947	73306
90	130526	77370
100	139722	82339

Table 3. Sizes of feature sets

As described in Section 3, the feature values are applied in the classifiers. Tables 4 and 5 give the best performances of the 4 classifiers in the two situations when we select the top N articles using N-gram probability for feature computation.

Due to the large size of the feature set, C5.0, SVM, and ME packages will not work at some

[1] http://www.fjoch.com/YASMET.html
[2] http://www.infoplease.com/people.html
[3] http://www.biography.com/search/index.jsp

point as more documents are encountered. The Naïve Bayes classification is more scalable as we use intermediate file to store probability tables.

Precision	First-level Classification	Second-level Classification
C5.0	82.90%	65.70%
Naïve Bayes	87.80%	72.86%
SVM	84.15%	75.71%
ME	86.59%	75.71%

Table 4. Precision scores for child classification

Precision	First-level Classification	Second-level Classification
C5.0	80.90%	56.80%
Naïve Bayes	83.00%	59.46%
SVM	78.72%	54.05%
ME	78.72%	51.35%

Table 5. Precision scores for spouse classification

Feature	# of times identified (out of 75)	$p(w_i\|w_{i-2,}w_{i-1})$
and his wife	35	0.6786
her husband ,	33	0.3082
and her husband	26	0.5476
was married to	20	0.8621
with his wife	14	0.875
her second husband	13	0.6667
her marriage to	13	0.5
ex - wife	12	0.3333
ex - husband	11	0.6667
her first husband	10	0.75
second husband ,	10	1
his first wife	8	0.3333
first husband ,	7	0.6667
second wife ,	7	0.3333
his first marriage	5	0.1667
s second wife	5	0.75

Table 6. Example trigram features for second-level classification for Spouse (one or multiple values)

The feature set has a large number of features. However, not all of them will be used for each person. We studied the number of times features are identified/used in the training and testing sets and their probabilities. Table 6 presents a list of some trigram features for second-level classification (one or multiple values) for Spouse. Obviously, indicating features have a large probability as expected. The second column gives the number of times the feature is used out of the training and testing set (75 persons in total).

Will more complex N-gram features work better?

Intuitively, being less ambiguous, more complex N-gram features carry more precise information and therefore should work better than simple ones. We studied the performances for different N-gram language model features. Below are the results of Naïve Bayes first-level classification for Child, using different N-gram features.

Top N Docs	Unigram	Bigram	Trigram
1	34.78%	54.35%	**67.39%**
10	30.48%	79.27%	**86.59%**
20	26.83%	82.93%	**85.37%**
30	24.39%	81.71%	**86.59%**

Table 7. Comparisons of classification precisions using different N-gram features for child

From Table 7, we can infer that bigram features work better than unigram features, and trigram features work better than bigrams when we select different numbers of top N documents. Trigram features actually bring enough evidence in classification. However, when we investigated 4-grams language features in the collected data, most of them are very sparse in the feature space of all the cases. Applying 4-grams or higher may not help in our task.

Will more data/documents help?

The performance of corpus-based statistical approaches usually depends on the size of corpus. A traditional view for most NLP problems is that more data will help to improve the system's performance. However, for data collected from a search engine, this may not be the case, since web data is usually ambiguous and noisy. We therefore investigate the data size's effect on system performance. Figure 3 gives the precision curves of the Naïve Bayes classifier for the first-level classification for Child.

Except for the case of top 1, where the top document alone may not contain too much useful information on selected topics, precision scores only have slight variations for increasing numbers of documents. For bigram features, over the top 50 through top 70 documents, the precision scores even get a little worse.

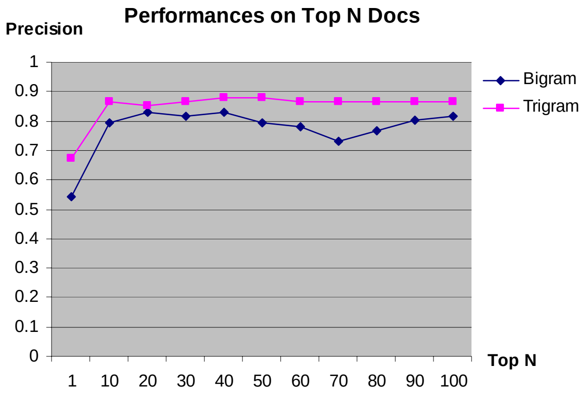

Figure 3. Performance on top N documents

4.3 Examples

Equipped with the cascading guiding strategy, we are able to handle questions containing implicatures. In our system, when we can determine the answer type is child or spouse, the cascading guiding system will help the answer extraction part to extract answers from the designated corpus. Figure 4 gives two examples of the strategy.

Q1: *Who is Sophia Smith's spouse?*
Classified: <NO_SPOUSE>
Answer: *She did not marry.*

Q2: *Who is John Ritter's child?*
Classified: <HAVING_CHILD>
Classified: <MULTIPLE_VALUES>
…

Figure 4. Classification Example for question

For the first question, the classifier recognizes there is no spouse for the target person and returns information for the answer generation. The features used here are the first-level classification result for SPOUSE_QUESTION. For the second question, the classifier recognizes the target person has a child first, followed by recognizing that the answer has multiple values. In this way, the strategy integrated to the question answering system can improve the system's performance by handling questions with implicatures.

5 Discussion and Future Work

Questions may have implicatures due to the flexibility of human language and conversation. In real question-answering systems, failure to handle them may either waste huge computation cost or impair system's performance. The traditional QA framework does not work well for questions containing implicatures. We describe a novel idea in this paper to identify potential traps in biographical questions and recognize whether there are multiple answers for a question.

Question-Answering systems, even when focused upon biographies, have to handle many facts, such as birth date, birth place, parents, training, accomplishments, etc. These values can be extracted using typical text harvesting approaches. However, when there are no values for some biographical information, the task becomes much more difficult because text seldom explicitly states a negative. For example, the following two questions require schools attended:

- *Where did <person> graduate from?*
- *What university did <person> attend?*

Our program scanned the two corpora of bios and found only 2 out 49320 bios explicitly stating that the subject never attended any school. Therefore, for some types of information, it will be much harder to identify null values through evidence from text. Some more complicated reasoning and inference may be required. Classifiers for some biographical facts may need to incorporate extra knowledge from other resources. The inherent relations between biography facts can also be used to validate each other. For example, the relations of marriage and child, birth place and childhood home, etc. may provide clues for cross-validation. We plan to investigate these problems in the future.

Acknowledgements

We wish to thank the anonymous reviewers for their helpful feedback and corrections. Also we thank Lei Ding, Feng Pan, and Deepak Ravichandran for their valuable comments on this work.

References

Boser, B.E., Guyon, I. and Vapnik, V. 1992. A training algorithm for optimal margin classifiers. *Proceedings of the ACM COLT 1992.*

Chang, C. and Lin, C. 2001. LIBSVM -- A library for support vector machines. Software available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm/*

Chu-Carroll, J., Czuba, K., Prager, J., and Ittycheriah, A. 2003. In question answering, two heads are better than one. *Proceedings of HLT-NAACL-2003*.

Clark, S., Steedman, M. and Curran, J.R. 2004. Object-extraction and question-parsing using CCG. *Proceedings EMNLP-2004*, pages 111-118, Barcelona, Spain.

Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. 2002. Web question answering: is more always better? *Proceedings of SIGIR-2002*.

Echihabi, A. and Marcu, D. 2003. A noisy channel approach to question answering. *Proceedings of ACL-2003*.

Fleischman, M., Hovy, E.H., and Echihabi, A. 2003. Offline strategies for online question answering: answering questions before they are asked. *Proceedings of ACL-2003.*

Gazdar, G. 1979. *Pragmatics: Implicature, presupposition, and logical form.* New York: Academic Press.

Harabagiu, S. 2001. Just-In-Time Question Answering. *Invited talk in Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium 2001.*

Hermjakob, U., Echihabi, A., and Marcu, D. 2002. Natural language based reformulation resource and web exploitation for question answering. *Proceedings of TREC-2002*.

Hermjakob, U., Hovy, E.H., and Lin, C. 2000. Knowledge-based question answering. *TREC-2000*.

Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M., and Lin, C. 2000. Question answering in Webclopedia. *Proceedings of TREC-2000*.

Hovy, E.H., Hermjakob, U., Lin, C., and Ravichandran, D. 2002. Using knowledge to facilitate factoid answer pinpointing. *Proceedings of COLING-2002*.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. 2003. A Practical Guide to Support Vector Classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Levinson, S. 1983. *Pragmatics*. Cambridge University Press.

Lita L.V. and Carbonell, J. 2004. Instance-based question answering: a data driven approach. *Proceedings of EMNLP 2004*.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, F., Rijke, M. 2003. The Multiple Language Question Answering Track at CLEF 2003. *CLEF 2003*: 471-486.

Mann, G. and Yarowsky, D. 2005. Multi-field information extraction and cross-document fusion. *Proceedings of ACL-2005*.

Moldovan, D., Clark, D., Harabagiu, S., and Maiorano, S. 2003. Cogex: A logic prover for question answering. *Proceedings of ACL-2003*.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., and Rus, V. 2000. The structure and performance of an open-domain question answering system. *Proceedings of ACL-2000.*

Nyberg, E. et al. 2003. A multi strategy approach with dynamic planning. *Proceedings of TREC-2003*.

Och, F. J.and Ney, H. 2002. Discriminative training and maximum entropy models for statistical machine translation. *Proceedings of ACL 2002* pp. 295-302.

Pasca, M. and Harabagiu, S. 2001. High Performance Question/Answering. *Proceedings of SIGIR-2001*.

Prager, J. M., Chu-Carroll, J., and Czuba, K.W.. 2004. Question answering using constraint satisfaction. *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04).*

Quinlan, J. R. 1993. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.

Ravichandran, D. and Hovy, E.H. 2002. Learning Surface Text Patterns for a Question Answering System. *Proceedings of ACL-2002.*

Saquete, E., Martínez-Barco, P., Muñoz, R., and Vicedo, J.L. 2004. Splitting complex temporal questions for question answering systems. *Proceedings of ACL'04.*

Schiffman, B., Mani, I., and Concepcion, K.J. 2001. Producing biographical summaries: combining linguistic knowledge with corpus statistics. *Proceedings of ACL/EACL-2001*.

Soricut, R. and Brill, E. 2004. Automatic question answering: beyond the factoid. *Proceedings of HLT/NAACL-2004*, Boston, MA.

Voorhees, E.M. 2003. Overview of the trec 2003 question answering track. *Proceedings of TREC-2003*.

Xu, J., Licuanan, A., Weischedel, R. 2003. TREC 2003 QA at BBN: Answering Definitional Questions. *Proceedings of TREC 2003*.

Yang, H., Chua, T.S., Wang, S., and Koh, C.K. 2003. Structured use of external knowledge for eventbased open domain question answering. *Proceedings of SIGIR-2003*.

Zhou, L., Ticrea, M., and Hovy, E.H. 2004. Multi-document biography summarization. *Proceedings of EMNLP-2004.*