Language-independent Probabilistic Answer Ranking for Question Answering

Jeongwoo Ko, Teruko Mitamura, Eric Nyberg
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{jko, teruko, ehn}@cs.cmu.edu

Abstract

This paper presents a language-independent probabilistic answer ranking framework for question answering. The framework estimates the probability of an individual answer candidate given the degree of answer relevance and the amount of supporting evidence provided in the set of answer candidates for the question. Our approach was evaluated by comparing the candidate answer sets generated by Chinese and Japanese answer extractors with the re-ranked answer sets produced by the answer ranking framework. Empirical results from testing on NTCIR factoid questions show a 40% performance improvement in Chinese answer selection and a 45% improvement in Japanese answer selection.

1 Introduction

Question answering (QA) systems aim at finding precise answers to natural language questions from large document collections. Typical QA systems (Prager et al., 2000; Clarke et al., 2001; Harabagiu et al., 2000) adopt a pipeline architecture that incorporates four major steps: (1) question analysis, (2) document retrieval, (3) answer extraction and (4) answer selection. Question analysis is a process which analyzes a question and produces a list of keywords. Document retrieval is a step that searches for relevant documents or passages. Answer extraction extracts a list of answer candidates from the retrieved documents. Answer selection is a process which pinpoints correct answer(s) from the extracted candidate answers.

Since the first three steps in the QA pipeline may produce erroneous outputs, the final answer selection step often entails identifying correct answer(s) amongst many incorrect ones. For example, given the question *"Which Chinese city has the largest number of foreign financial companies?"*, the answer extraction component produces a ranked list of five answer candidates: Beijing (AP880603-0268)[1], Hong Kong (WSJ920110-0013), Shanghai (FBIS3-58), Taiwan (FT942-2016) and Shanghai (FBIS3-45320). Due to imprecision in answer extraction, an incorrect answer ("Beijing") can be ranked in the first position, and the correct answer ("Shanghai") was extracted from two different documents and ranked in the third and the fifth positions. In order to rank "Shanghai" in the top position, we have to address two interesting challenges:

- *Answer Similarity*. How do we exploit similarity among answer candidates? For example, when the candidates list contains redundant answers (e.g., "Shanghai" as above) or several answers which represent a single instance (e.g. "U.S.A." and "the United States"), how much should we boost the rank of the redundant answers?

- *Answer Relevance*. How do we identify relevant answer(s) amongst irrelevant ones? This task may involve searching for evidence of a relationship between the answer

[1] Answer candidates are shown with the identifier of the TREC document where they were found.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 784–791,
Prague, Czech Republic, June 2007.

and the answer type or a question keyword. For example, we might wish to query a knowledge base to determine if "Shanghai" is a city (`IS-A(Shanghai, city)`), or to determine if Shanghai is in China (`IS-IN(Shanghai, China)`).

The first challenge is to exploit redundancy in the set of answer candidates. As answer candidates are extracted from different documents, they may contain identical, similar or complementary text snippets. For example, "U.S." can appear as "United States" or "USA" in different documents. It is important to detect redundant information and boost answer confidence, especially for list questions that require a set of unique answers. One approach is to perform answer clustering (Nyberg et al., 2002; Jijkoun et al., 2006). However, the use of clustering raises additional questions: how to calculate the score of the clustered answers, and how to select the cluster label.

To address the second question, several answer selection approaches have used external knowledge resources such as WordNet, CYC and gazetteers for answer validation or answer reranking. Answer candidates are either removed or discounted if they are not of the expected answer type (Xu et al., 2002; Moldovan et al., 2003; Chu-Carroll et al., 2003; Echihabi et al., 2004). The Web also has been used for answer reranking by exploiting search engine results produced by queries containing the answer candidate and question keywords (Magnini et al., 2002). This approach has been used in various languages for answer validation. Wikipedia's structured information was used for Spanish answer type checking (Buscaldi and Rosso, 2006).

Although many QA systems have incorporated individual features and/or resources for answer selection in a single language, there has been little research on a generalized probabilistic framework that supports answer ranking in multiple languages using any answer relevance and answer similarity features that are appropriate for the language in question.

In this paper, we describe a probabilistic answer ranking framework for multiple languages. The framework uses logistic regression to estimate the probability that an answer candidate is correct given multiple answer relevance features and answer similarity features. An existing framework which was originally developed for English (Ko et al., 2007) was extended for Chinese and Japanese answer ranking by incorporating language-specific features. Empirical results on NTCIR Chinese and Japanese factoid questions show that the framework significantly improved answer selection performance; Chinese performance improved by 40% over the baseline, and Japanese performance improved by 45% over the baseline.

The remainder of this paper is organized as follows: Section 2 contains an overview of the answer ranking task. Section 3 summarizes the answer ranking framework. In Section 4, we explain how we extended the framework by incorporating language-specific features. Section 5 describes the experimental methodology and results. Finally, Section 6 concludes with suggestions for future research.

2 Answer Ranking Task

The relevance of an answer to a question can be estimated by the probability P($correct(A_i)\ |A_i, Q$), where Q is a question and A_i is an answer candidate. To exploit answer similarity, we estimate the probability $P(correct(A_i)\ |A_i, A_j)$, where A_j is similar to A_i. Since both probabilities influence overall answer ranking performance, it is important to combine them in a unified framework and estimate the probability of an answer candidate as: $P(correct(A_i)|Q, A_1, ..., A_n)$.

The estimated probability is used to rank answer candidates and select final answers from the list. For factoid questions, the top answer is selected as a final answer to the question. In addition, we can use the estimated probability to classify incorrect answers: if the probability of an answer candidate is lower than 0.5, it is considered to be a wrong answer and is filtered out of the answer list. This is useful in deciding whether or not a valid answer to a question exists in a given corpus (Voorhees, 2002). The estimated probability can also be used in conjunction with a cutoff threshold when selecting multiple answers to list questions.

3 Answer Ranking Framework

This section summarizes our answer ranking framework, originally developed for English answers (Ko

$$
\begin{aligned}
&P(correct(A_i)|Q, A_1, ..., A_n) \\
&\quad \approx P(correct(A_i)|rel_1(A_i), ..., rel_{K1}(A_i), sim_1(A_i), ..., sim_{K2}(A_i)) \\
&\quad = \frac{exp(\alpha_0 + \sum_{k=1}^{K1} \beta_k rel_k(A_i) + \sum_{k=1}^{K2} \lambda_k sim_k(A_i))}{1 + exp(\alpha_0 + \sum_{k=1}^{K1} \beta_k rel_k(A_i) + \sum_{k=1}^{K2} \lambda_k sim_k(A_i))} \\
&\quad where, sim_k(A_i) = \sum_{j=1(j \neq i)}^{N} sim'_k(A_i, A_j).
\end{aligned}
$$

Figure 1: Estimating correctness of an answer candidate given a question and a set of answer candidates

et al., 2007). The model uses logistic regression to estimate the probability of an answer candidate (Figure 1). Each $\text{rel}_k(\text{A}_i)$ is a feature function used to produce an answer relevance score for an answer candidate A_i. Each $\text{sim}'_k(A_i, A_j)$ is a similarity function used to calculate an answer similarity between A_i and A_j. K1 and K2 are the number of answer relevance and answer similarity features, respectively. N is the number of answer candidates.

To incorporate multiple similarity features, each $\text{sim}_k(A_i)$ is obtained from an individual similarity metric, $\text{sim}'_k(A_i, A_j)$. For example, if Levenshtein distance is used as one similarity metric, $\text{sim}_k(A_i)$ is calculated by summing N-1 Levenshtein distances between one answer candidate and all other candidates.

The parameters α, β, λ were estimated from training data by maximizing the log likelihood. We used the Quasi-Newton algorithm (Minka, 2003) for parameter estimation.

Multiple features were used to generate answer relevance scores and answer similarity scores; these are discussed below.

3.1 Answer Relevance Features

Answer relevance features can be classified into knowledge-based features or data-driven features.

1) Knowledge-based features

Gazetteers: Gazetteers provide geographic information, which allows us to identify strings as instances of countries, their cities, continents, capitals, etc. For answer ranking, three gazetteer resources were used: the Tipster Gazetteer, the CIA World Factbook and information about the US states provided by 50states.com. These resources were used to assign an answer relevance score between -1 and 1 to each candidate. For example, given the question "Which city in China has the largest number of foreign financial companies?", the candidate "Shanghai" receives a score of 0.5 because it is a city in the gazetteers. But "Taiwan" receives a score of -1.0 because it is not a city in the gazetteers. A score of 0 means the gazetteers did not contribute to the answer selection process for that candidate.

Ontology: Ontologies such as WordNet contain information about relationships between words and general meaning types (synsets, semantic categories, etc.). WordNet was used to identify answer relevance in a manner analogous to the use of gazetteers. For example, given the question "Who wrote the book 'Song of Solomon'?", the candidate "Mark Twain" receives a score of 0.5 because its hypernyms include "writer".

2) Data-driven features

Wikipedia: Wikipedia was used to generate an answer relevance score. If there is a Wikipedia document whose title matches an answer candidate, the document is analyzed to obtain the term frequency (tf) and the inverse term frequency (idf) of the candidate, from which a tf.idf score is calculated. When there is no matched document, each question keyword is also processed as a back-off strategy, and the answer relevance score is calculated by summing the tf.idf scores obtained from individual keywords.

Google: Following Magnini et al. (2002), a query consisting of an answer candidate and question keywords was sent to the Google search engine. Then the top 10 text snippets returned by Google were analyzed to generate an answer relevance score by computing the minimum number of words between a keyword and the answer candidate.

3.2 Answer Similarity Features

Answer similarity is calculated using multiple string distance metrics and a list of synonyms.

String Distance Metrics: String distance metrics such as Levenshtein, Jaro-Winkler, and Cosine similarity were used to calculate the similarity between two English answer candidates.

Synonyms: Synonyms can be used as another metric to calculate answer similarity. If one answer is synonym of another answer, the score is 1. Otherwise the score is 0. To get a list of synonyms, three knowledge bases were used: WordNet, Wikipedia and the CIA World Factbook. In addition, manually generated rules were used to obtain synonyms for different types of answer candidates. For example, "April 12 1914" and "12th Apr. 1914" are converted into "1914-04-12" and treated as synonyms.

4 Extensions for Multiple Languages

We extended the framework for Chinese and Japanese QA. This section details how we incorporated language-specific resources into the framework. As logistic regression is based on a probabilistic framework, the model does not need to be changed to support other languages. We only retrained the model for individual languages. To support Chinese and Japanese QA, we incorporated new features for individual languages.

4.1 Answer Relevance Features

We replaced the English gazetteers and WordNet with language-specific resources for Japanese and Chinese. As Wikipedia and the Web support multiple languages, the same algorithm was used in searching language-specific corpora for the two languages.

1) Knowledge-based features

The knowledge-based features involve searching for facts in a knowledge base such as gazetteers and WordNet. We utilized comparable resources for Chinese and Japanese. Using language-specific resources, the same algorithms were applied to generate an answer relevance score between -1 and 1.

	#Articles	
Language	Nov. 2005	Aug. 2006
English	1,811,554	3,583,699
Japanese	201,703	446,122
Chinese	69,936	197,447

Table 1: Articles in Wikipedia for different languages

Gazetteers: There are few available gazetteers for Chinese and Japanese. Therefore, we extracted location data from language-specific resources. For Japanese, we extracted Japanese location information from Yahoo[2], which contains many location names in Japan and the relationships among them. For Chinese, we extracted location names from the Web. In addition, we translated country names provided by the CIA World Factbook and the Tipster gazetteers into Chinese and Japanese names. As there is more than one translation, top 3 translations were used.

Ontology: For Chinese, we used HowNet (Dong, 2000) which is a Chinese version of WordNet. It contains 65,000 Chinese concepts and 75,000 corresponding English equivalents. For Japanese, we used semantic classes provided by Gengo GoiTaikei[3]. Gengo GoiTaikei is a Japanese lexicon containing 300,000 Japanese words with their associated 3,000 semantic classes. The semantic information provided by HowNet and Gengo GoiTaikei was used to assign an answer relevance score between -1 and 1.

2) Data-driven features

Wikipedia: As Wikipedia supports more than 200 language editions, the approach used in English can be used for different languages without any modification. Table 1 shows the number of text articles in three different languages. Wikipedia's current coverage in Japanese and Chinese does not match its coverage in English, but coverage in these languages continues to improve.

To supplement the small corpus of Chinese documents available, we used Baidu

[2] http://map.yahoo.co.jp/

[3] http://www.kecl.ntt.co.jp/mtg/resources/GoiTaikei

(http://baike.baidu.com), which is similar to Wikipedia but contains more articles written in Chinese. We first search for Chinese Wikipedia. When there is no matching document in Wikipedia, each answer candidate is sent to Baidu and the retrieved document is analyzed in the same way to analyze Wikipedia documents.

The idf score was calculated using word statistics from Japanese Yomiuri newspaper corpus and the NTCIR Chinese corpus.

Google: The same algorithm was applied to analyze Japanese and Chinese snippets returned from Google. But we restricted the language to Chinese or Japanese so that Google returned only Chinese or Japanese documents. To calculate the distance between an answer candidate and question keywords, segmentation was done with linguistic tools. For Japanese, Chasen[4] was used. For Chinese segmentation, a maximum-entropy based parser was used (Wang et al., 2006).

3) Manual Filtering

Other than the features mentioned above, we manually created many rules for numeric and temporal questions to filter out invalid answers. For example, when the question is looking for a year as an answer, an answer candidate which contains only the month receives a score of -1. Otherwise, the score is 0.

4.2 Answer Similarity Features

The same features used for English were applied to calculate the similarity of Chinese/Japanese answer candidates. To identify synonyms, Wikipedia were used for both Chinese and Japanese. EIJIRO dictionary was used to obtain Japanese synonyms. EIJIRO is a English-Japanese dictionary containing 1,576,138 words and provides synonyms for Japanese words.

As there are several different ways to represent temporal and numeric expressions (Nyberg et al., 2002; Greenwood, 2006), language-specific conversion rules were applied to convert them into a canonical format; for example, a rule to convert Japanese Kanji characters to Arabic numbers is shown in Figure 2.

[4]http://chasen.aist-nara.ac.jp/hiki/ChaSen

Original answer string	Normalized answer string
三 千 億 円	3E+11 円
3,000億円	3E+11 円
一九九三年 七月 四 日	1993-07-04
1993 年 **7** 月**4** 日	1993-07-04
四分の一	0.25
５割	50 %

Figure 2: Example of normalized answer strings

5 Experiments

This section describes the experiments to evaluate the extended answer ranking framework for Chinese and Japanese QA.

5.1 Experimental Setup

We used Chinese and Japanese questions provided by the NTCIR (NII Test Collection for IR Systems), which focuses on evaluating cross-lingual and monolingual QA tasks for Chinese, Japanese and English. For Chinese, a total of 550 factoid questions from the NTCIR5-6 QA evaluations served as the dataset. Among them, 200 questions were used to train the Chinese answer extractor and 350 questions were used to evaluate our answer ranking framework. For Japanese, 700 questions from the NTCIR5-6 QA evaluations served as the dataset. Among them, 300 questions were used to train the Japanese answer extractor and 400 questions were used to evaluate our framework.

Both the Chinese and Japanese answer extractors use maximum-entropy to extract answer candidates based on multiple features such as named entity, dependency structures and some language-dependent features.

Performance of the answer ranking framework was measured by average answer accuracy: the number of correct top answers divided by the number of questions where at least one correct answer exists in the candidate list provided by an extractor. Mean Reciprocal Rank (MRR5) was also used to calculate the average reciprocal rank of the first correct answer in the top 5 answers.

The baseline for average answer accuracy was calculated using the answer candidate likelihood scores provided by each individual extractor; the

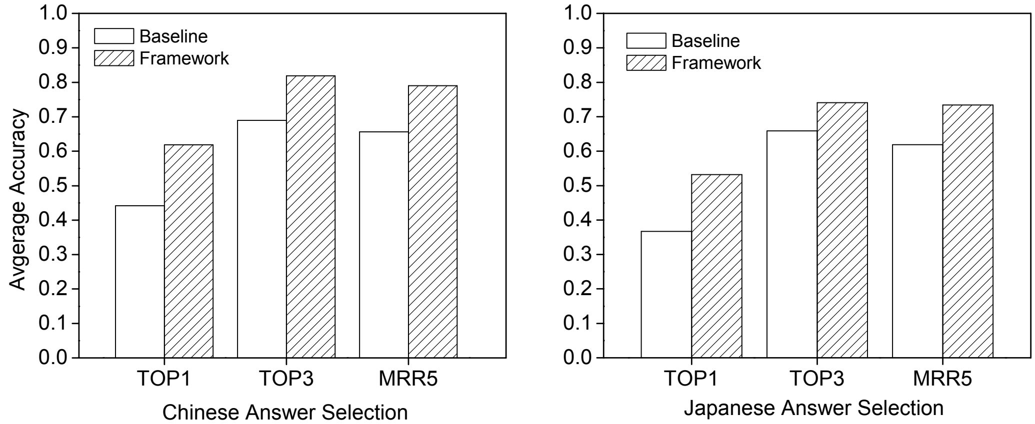

Figure 3: Performance of the answer ranking framework for Chinese and Japanese answer selection (TOP1: average accuracy of top answer, TOP3: average accuracy of top 3 answers, MRR5: average of mean reciprocal rank of top 5 answers)

answer with the best extractor score was chosen, and no validation or similarity processing was performed.

3-fold cross-validation was performed, and we used a version of Wikipedia downloaded in Aug 2006.

5.2 Results and Analysis

We first analyzed the average accuracy of top 1, top3 and top 5 answers. Figure 3 compares the average accuracy using the baseline and the answer selection framework. As can be seen, the answer ranking framework significantly improved performance on both Chinese and Japanese answer selection. As for the average top answer accuracy, there were 40% improvement over the baseline (Chinese) and 45% improvement over the baseline (Japanese).

We also analyzed the degree to which the average accuracy was affected by answer similarity and relevance features. Table 2 compares the average top answer accuracy using the baseline, the answer relevance features, the answer similarity features and all feature combinations. Both the similarity and the relevance features significantly improved answer selection performance compared to the baseline, and combining both sets of features together produced the best performance.

We further analyzed the utility of individual relevance features (Figure 4). For both languages, filtering was useful in ruling out wrong answers. The impact of the ontology was more positive for Japanese; we assume that this is because the Chinese ontology (HowNet) contains much less information overall than the Japanese ontology (Gengo GoiTaikei). The comparative impact of Wikipedia was similar. For Chinese, there were many fewer Wikipedia documents available. Even though we used Baidu as a supplemental resource for Chinese, this did not improve answer selection performance. On the other hand, the use of Wikipedia was very helpful for Japanese, improving performance by 26% over the baseline. This shows that the quality of answer relevance estimation is significantly affected by resource coverage.

	Baseline	Rel	Sim	All
Chinese	0.442	0.482	0.597	0.619
Japanese	0.367	0.463	0.502	0.532

Table 2: Average top answer accuracy of individual features (Rel: merging relevance features, Sim: merging similarity features, ALL: merging all features).

When comparing the data-driven features with the knowledge-based features, the data-driven features (such as Wikipedia and Google) tended to increase performance more than the knowledge-based features (such as gazetteers and WordNet).

Table 3 shows the effect of individual similarity features on Chinese and Japanese answer selec-

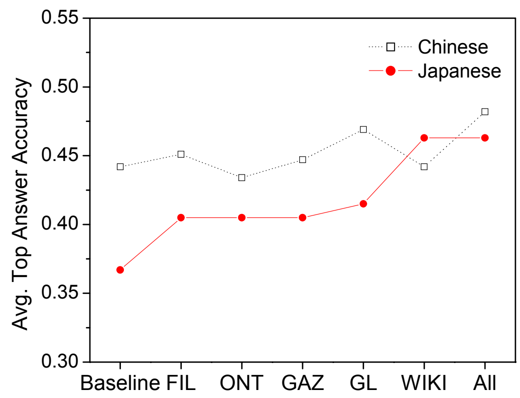

Figure 4: Average top answer accuracy of individual answer relevance features.(FIL: filtering, ONT, ontology, GAZ: gazetteers, GL: Google, WIKI: Wikipedia, ALL: combination of all relevance features)

	Chinese		Japanese	
	0.3	0.5	0.3	0.5
Cosine	**0.597**	**0.597**	0.488	0.488
Jaro-Winkler	0.544	0.518	0.410	0.415
Levenshtein	0.558	0.544	0.434	0.449
Synonyms	0.527	0.527	0.493	0.493
All	0.588	0.580	**0.502**	0.488

Table 3: Average accuracy using individual similarity features under different thresholds: 0.3 and 0.5 ("All": combination of all similarity metrics)

tion. As some string similarity features (e.g., Levenshtein distance) produce a number between 0 and 1 (where 1 means two strings are identical and 0 means they are different), similarity scores less than a threshold can be ignored. We used two thresholds: 0.3 and 0.5. In our experiments, using 0.3 as a threshold produced better results in Chinese. In Japanese, 0,5 was a better threshold for individual features. Among three different string similarity features (Levenshtein, Jaro-Winkler and Cosine similarity), cosine similarity tended to perform better than the others.

When comparing synonym features with string similarity features, synonyms performed better than string similarity in Japanese, but not in Chinese. We had many more synonyms available for Japanese

	Data-driven features	All features
Chinese	0.606	0.619
Japanese	0.517	0.532

Table 4: Average top answer accuracy when using data-driven features v.s. when using all features.

and they helped the system to better exploit answer redundancy.

We also analyzed answer selection performance when combining all four similarity features ("All" in Table 3). Combining all similarity features improved the performance in Japanese, but hurt the performance in Chinese, because adding a small set of synonyms to the string metrics worsened the performance of logistic regression.

5.3 Utility of data-driven features

In our experiments we used data-driven features as well as knowledge-based features. As knowledge-based features need manual effort to access language-specific resources for individual languages, we conducted an additional experiment only with data-driven features in order to see how much performance gain is available without the manual work. As Google, Wikipedia and string similarity metrics can be used without any additional manual effort when extended to other languages, we used these three features and compared the performance.

Table 4 shows the performance when using data-driven features v.s. all features. It can be seen that data-driven features alone achieved significant improvement over the baseline. This indicates that the framework can easily be extended to any language where appropriate data resources are available, even if knowledge-based features and resources for the language are still under development.

6 Conclusion

In this paper, we presented a generalized answer selection framework which was applied to Chinese and Japanese question answering. An empirical evaluation using NTCIR test questions showed that the framework significantly improves baseline answer selection performance. For Chinese, the performance improved by 40% over the baseline. For Japanese, the performance improved by 45% over the baseline. This shows that our probabilistic framework can be easily extended for multiple languages by reusing data-driven features (with new corpora) and adding language-specific resources (ontologies, gazetteers) for knowledge-based features.

In our previous work, we evaluated the performance of the framework for English QA using questions from past TREC evaluations (Ko et al., 2007). The experimental results showed that the combination of all answer ranking features improved performance by an average of 102% over the baseline. The relevance features improved performance by an average of 99% over the baseline, and the similarity features improved performance by an average of 46% over the baseline. Our hypothesis is that answer relevance features had a greater impact for English QA because the quality and coverage of the data resources available for English answer validation is much higher than the quality and coverage of existing resources for Japanese and Chinese. In future work, we will continue to evaluate the robustness of the framework. It is also clear from our comparison with English QA that more work can and should be done in acquiring data resources for answer validation in Chinese and Japanese.

Acknowledgments

We would like to thank Hideki Shima, Mengqiu Wang, Frank Lin, Justin Betteridge, Matthew Bilotti, Andrew Schlaikjer and Luo Si for their valuable support. This work was supported in part by ARDA/DTO AQUAINT program award number NBCHC040164.

References

D. Buscaldi and P. Rosso. 2006. Mining Knowledge from Wikipedia for the Question Answering task. In *Proceedings of the International Conference on Language Resources and Evaluation*.

J. Chu-Carroll, J. Prager, C. Welty, K. Czuba, and D. Ferrucci. 2003. A Multi-Strategy and Multi-Source Approach to Question Answering. In *Proceedings of Text REtrieval Conference*.

C. Clarke, G. Cormack, and T. Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of SIGIR*.

Zhendong Dong. 2000. Hownet: http://www.keenage.com.

A. Echihabi, U. Hermjakob, E. Hovy, D. Marcu, E. Melz, and D. Ravichandran. 2004. How to select an answer string? In T. Strzalkowski and S. Harabagiu, editors, *Advances in Textual Question Answering*. Kluwer.

Mark A. Greenwood. 2006. Open-Domain Question Answering. Thesis.

S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunsecu, R. Girju, V. Rus, and P. Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In *Proceedings of TREC*.

V. Jijkoun, J. van Rantwijk, D. Ahn, E. Tjong Kim Sang, and M. de Rijke. 2006. The University of Amsterdam at CLEF@QA 2006. In *Working Notes CLEF*.

J. Ko, L. Si, and E. Nyberg. 2007. A Probabilistic Framework for Answer Selection in Question Answering. In *Proceedings of NAACL/HLT*.

B. Magnini, M. Negri, R. Pervete, and H. Tanev. 2002. Comparing statistical and content-based techniques for answer validation on the web. In *Proceedings of the VIII Convegno AI*IA*.

T. Minka. 2003. A Comparison of Numerical Optimizers for Logistic Regression. Unpublished draft.

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

E. Nyberg, T. Mitamura, J. Carbonell, J. Callan, K. Collins-Thompson, K. Czuba, M. Duggan, L. Hiyakumoto, N. Hu, Y. Huang, J. Ko, L. Lita, S. Murtagh, V. Pedro, and D. Svoboda. 2002. The JAVELIN Question-Answering System at TREC 2002. In *Proceedings of Text REtrieval Conference*.

J. Prager, E. Brown, A. Coden, and D. Radev. 2000. Question answering by predictive annotation. In *Proceedings of SIGIR*.

E. Voorhees. 2002. Overview of the TREC 2002 question answering track. In *Proceedings of Text REtrieval Conference*.

M. Wang, K. Sagae, and T. Mitamura. 2006. A Fast, Accurate Deterministic Parser for Chinese. In *Proceedings of COLING/ACL*.

J. Xu, A. Licuanan, J. May, S. Miller, and R. Weischedel. 2002. TREC 2002 QA at BBN: Answer Selection and Confidence Estimation. In *Proceedings of Text REtrieval Conference*.