A Probabilistic Graphical Model for Joint Answer Ranking in Question Answering

Jeongwoo Ko
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jko@cs.cmu.edu

Luo Si
Department of Computer Science
Purdue University
Lafayette, IN 47907
lsi@cs.purdue.edu

Eric Nyberg
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
ehn@cs.cmu.edu

ABSTRACT

Graphical models have been applied to various information retrieval and natural language processing tasks in the recent literature. In this paper, we apply a probabilistic graphical model for answer ranking in question answering. This model estimates the joint probability of correctness of all answer candidates, from which the probability of correctness of an individual candidate can be inferred. The joint prediction model can estimate both the correctness of individual answers as well as their correlations, which enables a list of accurate and comprehensive answers. This model was compared with a logistic regression model which directly estimates the probability of correctness of each individual answer candidate. An extensive set of empirical results based on TREC questions demonstrates the effectiveness of the joint model for answer ranking. Furthermore, we combine the joint model with the logistic regression model to improve the efficiency and accuracy of answer ranking.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms

Keywords

Answer ranking, probabilistic graphical model, question answering

1. INTRODUCTION

Question Answering (QA) aims at finding exact answers to natural language questions in a large collection of documents. Most QA systems combine document retrieval with question analysis and extraction techniques to identify a set of likely candidates, from which the final answer(s) are selected [21, 4, 9]. Since question analysis, document retrieval and/or answer extraction may produce erroneous results, the selection process can be very challenging, as it often entails identifying relevant answer(s) amongst many irrelevant ones.

For example, given the question *"Which city in China 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 the imprecision in answer extraction, an incorrect answer ("Beijing") can be ranked at the first position. On the other hand, the correct answer ("Shanghai") was extracted from two different documents and ranked at the third and the fifth positions. In order to rank an answer like "Shanghai" in the top position, we have to address two interesting challenges:

- *Answer Relevance.* How do we identify relevant answer(s) amongst irrelevant ones? This task may involve searching for facts in a knowledge base. For example, `IS-IN(Shanghai, China)`, `IS-A(Shanghai, city)`.

- *Answer Similarity.* How do we exploit similarity among answer candidates? For example, when there are redundant answers ("Shanghai", as above) or several answers which represent a single instance (e.g. "Clinton, Bill" and "William Jefferson Clinton") in the candidate list, how much should we boost the rank of the answer candidate? Effective handling of redundancy is also important when identifying a set of novel answers for list or definition questions.

Although many QA systems address these issues separately, there has been little research on generating a probabilistic framework that allows any relevance and similarity features to be easily incorporated.

In our previous work [14], we proposed a probabilistic answer ranking model to address these two challenges. The model used logistic regression to estimate the probability

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

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

that an individual answer candidate is correct given relevance of the answer and the amount of supporting evidence provided by a set of similar answer candidates. The experimental results on TREC factoid questions show that the model significantly improves answer ranking performance for different extraction techniques.

However, this model considered each answer candidate separately, and did not consider correlation of the correctness of answer candidates, which is problematic for generating accurate and comprehensive answers. For example, several similar answers may be ranked high in the final answer list, but a less redundant answer may not be ranked high enough to reach the user's attention. In this paper, we propose a new probabilistic answer ranking framework that uses an undirected graphical model to estimate the joint probability of the correctness of all answer candidates, from which the probability of the correctness of an individual candidate can be inferred. The proposed model considers both the correctness of individual answers as well as their correlation to generate an accurate and comprehensive answer list.

In comparing the previous logistic regression model (which considers answers independently) to the new graphical model (which jointly considers answers), we will refer to the former as an `independent prediction model` and the latter as a `joint prediction model`. Experimental results on the TREC factoid questions [25] show that the joint prediction model significantly improves answer ranking performance over a baseline model, and produces a set of unique answers whose precision is higher than those produced by the independent prediction model. Furthermore, we incorporate the independent prediction model into the joint prediction model for improving the efficiency and accuracy of answer ranking.

This paper is organized as follows: Section 2 describes related work. Section 3 presents the independent prediction model and the joint prediction model. Section 4 lists the features used for the models. In Section 5, we describe our experimental methodology and the results. Finally, Section 6 concludes with suggestions for future research.

2. RELATED WORK

To identify relevant answers from a list of extracted candidates, several answer selection approaches have used external semantic resources. One of the most common approaches relies on WordNet, CYC and gazetteers for answer validation or answer reranking. In this approach, answer candidates are either removed or discounted if they are not found within the resource's hierarchy corresponding to the expected answer type of the question [26, 17, 3, 6]. The Web also has been used for answer reranking by exploiting search engine results produced by queries containing the answer candidate and question keywords [15]. Wikipedia's structured information has been used for answer type checking [2].

Although each of these approaches uses one or more semantic resources to independently support an answer, few have considered the potential benefits of combining resources together as evidence. There was an attempt to combine geographical databases with WordNet for type checking of location questions [23]. However, the experimental results show that the combination did not improve performance because of the increased semantic ambiguity which accompanies broader coverage of location names. This is evidence that the method of combining potential answers may matter as much as the choice of resources.

Collecting evidence from similar answer candidates to boost the confidence of a specific answer candidate is also important for answer selection. As answer candidates are extracted from different documents, they may contain identical, similar or complementary text snippets. Some previous work [19, 11] has used heuristic methods like manually compiled rules to cluster evidence from similar answer candidates. Graph-based clustering was also used to consider non-transitiveness in similarity [11].

Similarity detection is more important in list questions which require a set of unique answers. In many systems, cutoff threshold has been used to select the most probably top N answers [8, 13] or exhaustive search to find all possible candidates has been applied [27].

Although previous work has utilized evidence from similar answer candidates for a specific answer candidate, the algorithms only modeled each answer candidate separately and did not consider both answer relevance and answer correlation to prevent the biased influence of incorrect similar answers. As far as we know, no previous work has jointly modeled the correctness of available answer candidates in a formal probabilistic framework, which is very important for generating an accurate and comprehensive answer list.

3. MODELS

The independent prediction model estimates the probability of correctness of each answer candidate. It considers two factors. The first factor tries to identify relevant answers by estimating the probability $P(correct(A_i)|A_i, Q)$, where Q is a question and A_i is an answer candidate. The second factor tries to exploit answer similarity by estimating the probability $P(correct(A_i)\ |A_i, A_j)$, where A_j is similar to A_i. By combining these two factors together, the independent prediction model estimates the probability of an answer as: $P(correct(A_i)|Q, A_1, ..., A_n)$.

Instead of addressing each answer candidate separately, the joint prediction model estimates the joint probability of correctness of available answer candidates. In particular, the joint model estimates the probability of $P(correct(A_1)$,..., $correct(A_n)|Q, A_1, ..., A_n)$, where n is the number of answer candidates in consideration. The marginal probability of $P(correct(A_i)|Q, A_1, ..., A_n)$ for each individual answer as well as the conditional probability $P(correct(A_i)|correct(A_j)$ $, Q, A_1, ..., A_n)$ can be naturally derived from the joint probability.

3.1 Independent Prediction Model

The independent prediction model [14] directly estimates the probability of correctness of each individual answer candidate. It was implemented with logistic regression.

Figure 1 shows how logistic regression predicts the probability that an answer candidate is correct given multiple answer relevance features and answer similarity features. K1 and K2 are the number of features for answer relevance and answer similarity scores, respectively. n is the number of answer candidates for a question. Each $\mathrm{rel}_k(\mathrm{A}_i)$ is a feature function used to produce an answer relevance score for an individual answer candidate A_i. Each $\mathrm{sim}_k(A_i, A_j)$ is a scoring function used to calculate an answer similarity between A_i and A_j. Each $\mathrm{sim}'_k(A_i)$ represents one similarity feature

$$
\begin{aligned}
&P(correct(A_i)|Q, A_1, ..., A_n) \\
&\quad \approx P(correct(A_i)|rel_1(A_i), ..., rel_{K1}(A_i), sim'_1(A_i), ..., sim'_{K2}(A_i)) \\
&\quad = \frac{exp(\alpha_0 + \sum_{k=1}^{K1} \beta_k rel_k(A_i) + \sum_{k=1}^{K2} \lambda_k sim'_k(A_i))}{1 + exp(\alpha_0 + \sum_{k=1}^{K1} \beta_k rel_k(A_i) + \sum_{k=1}^{K2} \lambda_k sim'_k(A_i))} \\
&\quad where, sim'_k(A_i) = \sum_{j=1(j \neq i)}^{N} sim_k(A_i, A_j).
\end{aligned}
$$

Figure 1: Independent prediction model

$$
P(S_1, ..., S_n) = \frac{1}{Z} exp\left(\sum_{i=1}^{n}\left[(\sum_{k=1}^{K1} \beta_k rel_k(A_i))S_i + \sum_{N(i)}(\sum_{k=1}^{K2} \lambda_k sim_k(A_i, A_{N(i)}))S_i S_{N(i)}\right]\right)
$$

Figure 2: Joint prediction model

for an answer candidate A_i and is obtained by summing N-1 answer similarity scores to represent the similarity of one answer candidate to all other candidates.

The parameters α, $\{\beta_k\}$, and $\{\lambda_k\}$ are estimated from training data by maximizing the log likelihood. In particular, the Quasi-Newton algorithm [16] is used.

After applying the independent prediction model, answer candidates are reranked according to their estimated probability. For factoid questions, the top answer is selected as a final answer to the question. As logistic regression can be used for a binary classification task with a default threshold of 0.5, we may use the model to identify incorrect answers: if the probability of an answer candidate is lower than 0.5, it may be 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 exists in the corpus [24].

3.2 Joint Prediction Model

The joint prediction model estimates the joint probability of all answer candidates, from which the probability of an individual candidate is inferred. This estimation is performed using a probabilistic graphical model.

Graphical models have been applied to solve problems in many different domains such as artificial intelligence, computational biology, image processing, computer vision, information retrieval and natural language processing. A graphical model is either directed or undirected. Directed graphs can be used to represent causal relationships between variables [20]. Undirected graphs can be used to represent correlations between variables [5]. In this paper, we used an undirected graph for the joint prediction model.

Undirected graphical models are defined as a product of cliques (Equation 1). A clique is a complete subgraph whose nodes are fully connected.

$$
P(X) = \frac{1}{Z} \prod_{c \in C} \psi_c(X_c) \tag{1}
$$

where ψ_c is a positive potential function for a clique in the graph, C is a set of cliques in the graph, and Z is a normalization constant.

A Boltzmann machine [10, 12] is a special type of undirected graphical model whose node S_i has a binary value: either {0,1} or {-1,1}. The joint probability of this graph is represented in Equation 2:

$$
P(S) = \frac{1}{Z} exp(\sum_{i<j} \theta_{ij} S_i S_j + \sum_i \theta_{i0} S_i) \tag{2}
$$

where θ_{ij}=0 if nodes S_i and S_j are not neighbors in the graph.

We adapted a Boltzmann machine for answer ranking. Each node S_i in the graph represents an answer candidate A_i and its binary value represents answer relevance (Equation 3). The weights on the edges represent answer similarity between two nodes. If two answers are not similar, the weight between them is 0.

$$
S_i = \begin{cases} 1, & if\ A_i\ is\ correct \\ 0, & otherwise \end{cases} \tag{3}
$$

The joint probability of the model can be represented in Figure 2. Each $\text{rel}_k(A_i)$ is a feature function used to produce an answer relevance score for an individual answer candidate and each $\text{sim}_k(A_i, A_{N(i)})$ is a feature function used to calculate the similarity between an answer candidate A_i and its neighbor answer $A_{N(i)}$. The parameters α, $\{\beta_k\}$, and $\{\lambda_k\}$ are estimated from training data using the Quasi-Netwon algorithm.

As each node has a binary value (either 0 or 1), this model uses the answer relevance scores only when an answer candidate is correct (S_i=1) and uses the answer similarity scores only when two answer candidates are correct (S_i=1 and $S_{N(i)}$=1). If S_i=0, the relevance and similarity scores are ignored. If $S_{N(i)}$=0, the answer similarity scores are ignored. This prevents the biased influence of incorrect similar answers.

As the joint prediction model is based on a probabilistic graphical model, it can support probabilistic inference to identify a set of accurate and comprehensive answers. Fig 3 shows the algorithm for selecting answers using this model. Examples will be found in the next section.

1. Create an empty answer pool.
2. Estimate the joint probability of all answer candidates: $P(S_1, ..., S_n)$
3. Calculate the marginal probability that an individual answer candidate is correct.

$$P(correct(A_i)|Q, A_1, ..., A_n) \approx \sum_{S_1} ... \sum_{S_{i-1}} \sum_{S_{i+1}} ... \sum_{S_n} P(S_i = 1, S_1, ..., S_{i-1}, S_{i+1}, ..., S_n)$$

4. Choose the answer candidate whose marginal probability is highest, and move it to the answer pool.
5. For the remaining answer candidates,
   5.1. Calculate the conditional probability of individual answers given the chosen answer(s). For example, if A_i is chosen as the first answer, calculate $P(correct(A_j)|correct(A_i), Q, A_1, ..., A_n)$.
   5.2. Calculate the score of each answer candidate from the marginal and conditional probability.

$$Score(A_j) = P(correct(A_j)|Q, A_1, ..., A_n) - \max_i P(correct(A_j)|correct(A_i), Q, A_1, ..., A_n)$$

   5.3. Choose the answer whose $Score(A_j)$ is maximum, and move it to the answer pool.

Figure 3: Algorithm to rank answers from the joint prediction model.

Keeping consistent with the independent prediction model, answer candidates whose marginal probability is lower than 0.5 are removed from the answer list. If only one answer should be provided for any factoid question, the answer whose marginal probability is highest is selected as the final answer to the question.

3.3 Comparison

Both the independent prediction model and the joint prediction model provide a general probabilistic framework to estimate the probability of correctness of an individual answer candidate from answer relevance and similarity features. But the independent prediction model directly estimates the probability of an individual answer and the joint prediction model estimates the joint probability of all answers, from which the probability of correctness of an individual candidate is inferred.

One advantage of the joint prediction model is that it supports probabilistic inference. For example, the question *"Who have been the U.S. presidents since 1993?"* requires a list of person names as the answer. As person names can be represented in several different ways (e.g., "Bill Clinton", "William J. Clinton", "Clinton, Bill"), it is important to find unique names as the final answers. This task can be done by using the conditional probability inferred from the joint prediction model. For example, assume that we have three answer candidates for this question: "William J. Clinton", "Bill Clinton" and "George W. Bush". The probability of correctness of each answer has been calculated by marginalizing the joint probability of all answer candidates.

P(correct(William J. Clinton))= 0.758
P(correct(Bill Clinton)) = 0.755
P(correct(George W. Bush) = **0.617**

In this example, the marginal probability P(correct(Bill Clinton)) and P(correct(William J. Clinton)) are high because "Bill Clinton" and "William J. Clinton" are supporting each other. Based on the marginal probabilities, we first choose the answer candidate A_i whose marginal probability is the highest. In this example, "William J. Clinton" is chosen and added to the answer pool. Then we calculate the conditional probability of the remaining answer candidates given the first answer.

P(correct(Bill Clinton)|correct(William J. Clinton))
= 0.803
P(correct(George W. Bush)|correct(William J. Clinton))
= **0.617**

Even though the marginal probability P(correct(Bill Clinton) is higher than P(correct(George W. Bush)), "Bill Clinton" is not chosen as the second answer because the conditional probability of "Bill Clinton" given "William J. Clinton" is high, which indicates that the answer of "Bill Clinton" tends to be redundant to the answer of "William J. Clinton".

On the other hand, P(correct(George W. Bush) | correct(William J. Clinton)) is much smaller than P(correct(Bill Clinton) | correct(William J. Clinton)). Therefore, according to the algorithm in Figure 3, "George W. Bush" is chosen as the second answer even though its marginal probability is low. In this way we can select the best unique answers from a list of answer candidates.

In terms of efficiency, the independent prediction model has better time performance than the joint prediction model because the joint prediction requires $O(2^N)$ time complexity for calculating the joint probability, where N is the size of the graph (i.e. number of answer candidates). To address this issue, approximate inference (e.g. Markov chain Monte Carlo sampling or variational inference) can be used.

4. FEATURES

This section summarizes the features used to generate answer relevance scores and answer similarity scores (more details on the features found in [14]).

4.1 Answer Relevance Features

Multiple external resources were used as answer relevance features. Each answer relevance feature produces a relevance score which predicts whether or not an answer candidate is a relevant answer to the question.

4.1.1 Knowledge-based features

The knowledge-based features involve searching for facts in a knowledge base such as gazetteers and WordNet.

Gazetteers: Gazetteers provide geographic information, which allows us to identify strings as instances of countries, their cities, continents, capitals, etc. To identify relevant answers, three gazetteer resources were used: the Tipster Gazetteer, the CIA World Factbook (https://www.cia.gov/cia/publications/factbook/index.html) 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 "What continent is Togo on?", the candidate "Africa" receives a score of 1.0 because gazetteers can answer this question. The candidates "Asia" receive a score of 0.5 because it is a continent name in gazetteers and matches to the expected answer type of the question. But "Ghana" receives a score of -1.0 because it is not a continent in gazetteers. A score of 0 means the gazetteers did not contribute to the answer selection process for that candidate.

For some numeric questions, range checking was used to validate numeric questions [22]. For example, given the question *"How many people live in Italy?"*, if an answer candidate is within ± 10% of the population stated in the CIA World Factbook, it receives a score of 1.0. If it is in the range of 20%, its score is 0.5. If it significantly differs by more than 20%, it receives a score of -1.0. The threshold may vary based on when the document was written and when the census was taken[2].

WordNet: WordNet[7] was used to identify answer relevance in a manner analogous to 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*. For the question "What is the capital of Uruguay?", the candidate "Montevideo" receives a score of 1.0 because WordNet contains this information. As with the gazetteer score, a score of 0 means that WordNet does not contribute to the answer ranking process for a candidate.

4.1.2 Data-driven features

Wikipedia and Google were used in a data-driven approach by calculating tf.idf and word distance.

Wikipedia: To generate an answer relevance score, Wikipedia documents as well as Wikipedia's structured information were used. A query consisting of an answer candidate is sent to Wikipedia. If there is a Wikipedia document whose title matches the 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 sent to Wikipedia as a back-off strategy, and the answer relevance score is calculated by summing the tf.idf scores of each keyword. To calculate word frequency, the TREC Web Corpus[3] was used as a large background corpus.

Google: Following Magnini et al. [15], a query consisting of an answer candidate and question keywords was sent to the Google search engine. The top 10 text snippets returned from Google were then analyzed to generate an answer relevance score by computing the word distance between a keyword and the answer candidate.

4.2 Answer Similarity Features

The similarity between two answer candidates was measured with a string distance metric and a list of synonyms.

String Distance Metric: There are several different string distance metrics to calculate the similarity of short strings. We used Levenshtein distance for the experiments reported here.

Synonyms: Synonymity can be used as another metric in calculating answer similarity. If one answer is a synonym of another answer, their similarity score is 1. Otherwise the score is 0. To build a list of synonyms, three knowledge bases were used: WordNet, the CIA World Factbook and Wikipedia.

In addition, manually generated rules are used to obtain synonyms for different types of answer candidates.

- Dates are converted into the ISO 8601 date format (YYYY-MM-DD) (e.g., "April 12 1914" and "12th Apr. 1914" are converted into "1914-04-12" and considered as synonyms).
- Temporal expressions are converted into the HH:MM:SS format (e.g., "six thirty five p.m." and "6:35 pm" are converted into "18:35:xx" and considered as synonyms).
- Numeric expressions are converted into scientific notation (e.g, "one million" and "1,000,000" are converted into "1e+06" and considered as synonyms).
- Representative entities are converted into country names when the expected answer type is `COUNTRY` (e.g., "the Egyptian government" is changed to "Egypt" and "Clinton administration" is changed to "U.S.").

5. EXPERIMENTS

This section describes our experiments to compare the performance of the joint prediction model, the independent prediction model and the baseline algorithm. The JAVELIN QA system [19] was used for the evaluation.

5.1 Experimental Setup

A total of 1818 questions from the TREC8-12 QA evaluations were used as the testbed, and 5-fold cross validation was used to evaluate the models.

To better understand how the performance of the models varies for different extraction techniques, we tested the answer ranking models with three JAVELIN answer extraction components:

[2]The ranges used here were found to work effectively, but were not explicitly validated or tuned.

[3]http://ir.dcs.gla.ac.uk/testcollections/wt10g.html

Table 1: Performance of baseline, the independent prediction model (IP) and the joint prediction model (JP) when ranking the top 10 answer candidates produced by each individual extractor.

	FST			LIGHT			SVM		
	Baseline	IP	JP	Baseline	IP	JP	Baseline	IP	JP
TOP1	0.691	0.873	0.870	0.404	0.604	0.605	0.282	0.532	0.536
TOP3	0.906	0.950	0.966	0.619	0.706	0.756	0.507	0.629	0.675
MRR	0.868	0.936	0.952	0.592	0.699	0.729	0.482	0.618	0.652

Table 2: Average precision when ranking the top 10 answer candidates.

Average	FST			LIGHT			SVM		
Precision	Baseline	IP	JP	Baseline	IP	JP	Baseline	IP	JP
at rank1	0.691	0.873	0.870	0.404	0.604	0.605	0.282	0.532	0.536
at rank2	0.381	0.420	0.463	0.292	0.359	0.383	0.221	0.311	0.339
at rank3	0.260	0.270	0.297	0.236	0.268	0.280	0.188	0.233	0.248
at rank4	0.174	0.175	0.195	0.201	0.222	0.222	0.167	0.193	0.199
at rank5	0.117	0.117	0.130	0.177	0.190	0.190	0.150	0.167	0.170

- FST: an answer extractor based on finite state transducers that incorporate a set of extraction patterns (both manually created and generalized patterns), and are trained for each answer type;
- LIGHT: an extractor that selects answer candidates using a non-linear distance heuristic between the keywords and an answer candidate;
- SVM: an extractor that uses Support Vector Machines to discriminate between correct and incorrect answers based on local semantic and syntactic context.

Answer ranking performance is measured by the 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. Three measures are used: TOP1, TOP3, MRR. TOP1 is the average accuracy of the top ranked answers. TOP3 is the average of correct answers ranked in the top 3 positions[4]. MRR is the average of mean reciprocal rank of the top 5 answers.

Even though the data set contains only factoid questions, approximately 36% of the questions have more than one correct answer, and the average number of correct answers for those questions is 5. Especially, location, person name, numeric and temporal questions tend to have more than one correct answer. For example, given the question "Where is the tallest roller coaster located?", there are three answers in the TREC corpus: Cedar Point, Sandusky, Ohio. All of them are correct, although they represent geographical areas of increasing generality. Some questions require more than one correct answer. For example, for the question "Who is the tallest man in the world?", the correct answers are "Monjane, Gabriel Estavao, Robert Wadlow, AliNashnush, Barman". In addition, there are some list questions (e.g. "Name one of the major gods of Hinduism."). Therefore, we evaluate the average precision of the top 5 answers in order to see how effectively the joint prediction model can identify unique answers. The average precision is calculated by counting the number of unique correct answers among the top N answers. For example, when the first two answers are "William J. Clinton" and "George Bush", and the third answer is "Clinton, Bill", the precision at rank 3 is 2/3.

The baseline was calculated with the answer candidate scores provided by each individual extractor; the answer with the best extractor score was chosen, and no validation or similarity processing was performed. For Wikipedia, we used data downloaded in November 2005 (1,811,554 articles).

5.2 Results and Analysis

This section compares the performance of the joint prediction model with the independent prediction model. Furthermore, we incorporate the independent prediction model into the joint prediction model to improve the efficiency of the joint prediction.

5.2.1 *Joint Prediction Model*

As the joint prediction model is based on a graphical model, it requires $O(2^N)$ time complexity where N is the size of the graph (i.e. number of answer candidates). Therefore, we tested it only with the top 10 answer candidates provided by each individual answer extractor [5].

Table 1 shows the performance of the joint prediction model, compared with the independent prediction model and the baseline when ranking the top 10 answer candidates. As for TOP1, the joint prediction model significantly improved performance over the baseline for all three extractors. This shows the effectiveness of the probabilistic graphical model for selecting the most relevant answer. When compared with the independent prediction model, the joint prediction model performed as well as the independent prediction model in ranking the relevant answer at the top position.

TOP3 and MRR show the performance of the three algorithms when they return multiple answers for each question. It can be seen that the joint prediction model performed better than the independent prediction model because it could identify unique correct answers by estimating conditional probability.

To further investigate how much the joint prediction model could identify comprehensive results, we analyzed the aver-

[4]If at least one correct answer exists among the top 3 answers, the score is 1. otherwise, the score is 0.

[5]We calculated the marginal and conditional probability using exact inference (brute force enumeration).

Table 3: Performance of IP and the efficient JP (EJP) when ranking all answer candidates produced by each individual extractor.

	FST			LIGHT			SVM		
	Baseline	IP	EJP	Baseline	IP	EJP	Baseline	IP	EJP
TOP1	0.691	0.880	0.874	0.404	0.624	0.637	0.282	0.584	0.583
TOP3	0.906	0.947	0.960	0.619	0.756	0.772	0.507	0.723	0.747
MRR	0.868	0.935	0.950	0.592	0.737	0.751	0.482	0.702	0.724

Table 4: Average precision of IP and the efficient JP (EJP) when ranking all answer candidates.

Average	FST			LIGHT			SVM		
Precision	Baseline	IP	EJP	Baseline	IP	EJP	Baseline	IP	EJP
at rank1	0.691	0.880	0.874	0.404	0.624	0.637	0.282	0.584	0.583
at rank2	0.381	0.414	0.548	0.292	0.377	0.541	0.221	0.350	0.498
at rank3	0.260	0.269	0.377	0.236	0.274	0.463	0.188	0.255	0.424
at rank4	0.174	0.178	0.259	0.201	0.220	0.399	0.167	0.203	0.366
at rank5	0.117	0.118	0.181	0.177	0.191	0.349	0.150	0.175	0.319

age precision at top 5 answers. Table 2 shows the average precision of the three models. It can be seen that the joint prediction model produced the answer list whose average precision is higher than the independent prediction model. This is additional evidence that the joint prediction model can produce a better comprehensive answer list.

5.2.2 *Efficient Joint Prediction Model*

Efficiency is one issue for the joint prediction model because it requires $O(2^N)$ time complexity for calculating the joint probability. Table 5 shows the average number of answer candidates provided by individual extractors. LIGHT and SVM often return more than 30 answer candidates per question. Even though FST returns a small number of answer candidates, it sometimes returns more than 30 answer candidates for a question. This requires more than $O(2^{30})$, which is intractable with exact inference.

Table 5: The average number of answer candidates per question.

Answer extractor	Average # of answers
FST	4.19
LIGHT	36.93
SVM	38.70

To address this issue, approximate inference may be used (e.g., Markov chain Monte Carlo sampling or variational inference). Instead of using inexact variational inference or slow sampling methods, we propose an efficient version of the joint prediction model by preselecting the answer candidates with the independent prediction model.

In this approach, we first apply the independent prediction model with all the candidates provided by an answer extractor. Then we choose the top 10 answer candidates returned from the independent prediction model as the input to the joint prediction model. Finally, we run the joint prediction model with the top 10 answers.

Table 3 compares the performance of the efficient joint prediction model with the independent prediction model. It shows that the efficient joint prediction model performed as well as the independent prediction model when selecting the top relevant answer. When comparing TOP3 and MRR, the efficient joint prediction model performed better than the independent prediction model because it could identify unique correct answers by estimating the conditional probability.

We also analyzed the average precision of the independent prediction model and the efficient joint prediction model. Table 4 shows that the efficient joint prediction model performed much better than the independent prediction model. For example, the efficient joint prediction model improved the average precision at rank 2 by 33% (FST), 43% (LIGHT) and 42% (SVM) over independent prediction. This is quite a significant improvement over the original joint prediction model because the original joint prediction model improved the average precision at rank 2 by just 10% (FST), 6% (LIGHT) and 9% (SVM).

As for the average precision at rank 5, the extended joint prediction model improved performance by 53% (FST), 83% (LIGHT) and 82% (SVM) over the independent prediction model. When comparing performance gain in the three different extraction techniques, we had less improvement for FST because it produces an average of 4.19 answer candidates (as shown in Table 5).

This additional analysis on average precision shows clearly that the revised version of the joint prediction model can generate more comprehensive results in an efficient way.

6. CONCLUSIONS

In this paper, we proposed a new probabilistic answer ranking model based on a probabilistic graphical model and evaluated its performance by comparing it with an existing independent prediction model.

Even though the independent prediction and joint prediction models both provide a general probabilistic framework for estimating the probability of an individual answer candidate from answer relevance and similarity features, they differ in how they estimate the probability. The independent prediction model directly estimates the probability of correctness of an individual answer candidate. On the other hand, the joint prediction model uses an undirected graph to estimate the joint probability of correctness of available answer candidates. From the joint probability, we can infer the marginal probability of the correctness of an individual candidate and the conditional probability of correctness for different answers. This enables better answer ranking results for a more accurate and comprehensive answer list.

An extensive set of empirical results on TREC questions shows that the joint prediction model significantly improved answer ranking performance and is better at finding a unique set of correct answers (e.g. for a list-type question). Furthermore, we also extended the joint prediction model to improve its algorithmic efficiency by utilizing the outputs produced by the independent prediction model.

As far as we know, this is the first research work that proposes a formal probabilistic framework to jointly model the correctness and correlation of answer candidates in question answering. We plan to evaluate this new answer ranking model with other types of questions such as list and definition questions in the future. As definition questions require long text answers, different features should be used for answer ranking. Possible relevance features include question keyword inclusion and predicate structure match [18]. For answer similarity, we intend to explore other novelty measurements (e.g., in Allan et al. [1]). We also plan to apply approximate inference algorithms like variational inference to implement the joint prediction model.

7. ACKNOWLEDGMENTS

This work was supported in part by ARDA/DTO Advanced Question Answering for Intelligence (AQUAINT) program award number NBCHC040164.

8. REFERENCES

[1] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of SIGIR*, 2003.

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

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

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

[5] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

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

[7] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[8] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, J. Williams, and J. Bensley. Answer mining by combining extraction techniques with abductive reasoning. In *Proceedings of TREC*, 2003.

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

[10] G. Hinton and T. Sejnowski. Learning and relearning in Boltzmann machines. In Rumelhart, editor, *Parallel Distributed Processing*, pages pp. 282–317. MIT Press, 1986.

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

[12] M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In M. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Publishers, Boston, 1998.

[13] B. Katz, J. J. Lin, D. Loreto, W. Hildebrandt, M. Bilotti, S. Felshin, A. Fernandes, G. Marton, and F. Mora. Integrating web-based and corpus-based techniques for question answering. In *TREC*, pages 426–435, 2003.

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

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

[16] T. Minka. A Comparison of Numerical Optimizers for Logistic Regression. Unpublished draft, 2003.

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

[18] E. Nyberg, T. Mitamura, R. Frederking, M. Bilotti, K. Hannan, L. Hiyakumoto, J. Ko, F. Lin, V. Pedro, and A. Schlaikjer. JAVELIN I and II Systems at TREC 2005. In *Proceedings of Text REtrieval Conference*, 2005.

[19] E. Nyberg, T. Mitamura, R. Frederking, V. Pedro, M. Bilotti, A. Schlaikjer, and K. Hannan. Extending the javelin qa system with domain semantics. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, 2005.

[20] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

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

[22] J. Prager, J. Chu-Carroll, K. Czuba, C. Welty, A. Ittycheriah, and R. Mahindru. IBM's PIQUANT in TREC2003. In *Proceedings of TREC*, 2003.

[23] S. Schlobach, M. Olsthoorn, and M. de Rijke. Type checking in open-domain question answering. In *Proceedings of European Conference on Artificial Intelligence*, 2004.

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

[25] E. Voorhees. Overview of the TREC 2003 question answering track. In *Proceedings of Text REtrieval Conference*, 2003.

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

[27] H. Yang, H. Cui, M. Maslennikov, L. Qiu, M.-Y. Kan, and T.-S. Chua. QUALIFIER In TREC-12 QA Main Task. In *TREC*, pages 480–488, 2003.