Integrated Computer-Aided Engineering 16 (2009) 271–281
DOI 10.3233/ICA-2009-0316
IOS Press

A supervised learning approach to biological question answering

Ryan T.K. Lin[a], Justin Liang-Te Chiu[a,c], Hong-Jie Dai[a,d], Richard Tzong-Han Tsai[b,*], Min-Yuh Day[a] and Wen-Lian Hsu[a,d]
[a]*Institute of Information Science, Academia Sinica, Taipei, Taiwan*
[b]*Department of Computer Science & Engineering, Yuan Ze University, Taoyuan, Taiwan*
[c]*Department of Computer Science & Information Engineering, National Taiwan University, Taipei, Taiwan*
[d]*Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan*

Abstract. Biologists rely on keyword-based search engines to retrieve superficially relevant papers, from which they must filter out the irrelevant information manually. Question answering (QA) systems can offer more efficient and user-friendly ways of retrieving such information. Two contributions are provided in this paper. First, a factoid QA system is developed to employ a named entity recognition module to extract answer candidates and a linear model to rank them. The linear model uses various semantic features, such as named entity types and semantic roles. To tune the weights of features used by the model, a novel supervised learning algorithm, which only needs small amounts of training data, is provided. Second, a QA system may assign several answers with the same score, making evaluation unfair. To solve this problem, an efficient formula for a mean average reciprocal rank (MARR) is proposed to reduce the complexity of its computation. After employing all effective semantic features, our system achieves a top-1 MARR of 74.11% and top-5 MARR of 76.68%. In comparison of the baseline system, the top-1 and top-5 MARR increase by 9.5% and 7.1%. In addition, the experiment result on test set shows our ranking method, which achieves 55.58% top-1 MARR and 66.99% top-5 MARR, significantly surpasses traditional BM25 and simple voting in performance by averagely 35.23% and 36.64%, respectively.

1. Introduction

When planning a research project, biological researchers are predominantly interested in relevant molecular pathways and underlying mechanisms [5]. Since molecular biology is a rapidly developing and evolving field, it is essential for researchers to be able to effectively search recently published papers. Currently, most of them use keyword-based search engines such as PubMed and Google [31]. However, with the tremendous amount of new biomedical literature being published and the increasing complexity of molecular pathway descriptions, it is becoming much harder to find specific and relevant information about molecular interactions using these tools. Keyword-based information retrieval (IR) is designed to find broadly related passages, not specific answers. When biologists are interested in exactly which protein is involved in a pathway, lots of manual effort is still required to locate the desired terms from search results. Question answering (QA) systems can offer more efficient and user-friendly ways of retrieving such information.

In this paper, a QA system is built, and aimed to answer questions about biomolecular events, such as gene and protein interactions. The answers to such questions mainly consist of short pieces of information such as protein, DNA, RNA, or cell names as well as times and locations. In addition to syntactic features, two effective semantic features, named entity types and semantic roles, are incorporated to help match the question with its corresponding answer phrases contained in retrieved documents. Accordingly, our system is focused on factoid questions, to which the answers are named entities.

*Corresponding author: Richard Tzong-Han Tsai, Department of Computer Science & Engineering, Yuan Ze University, Taoyuan, Taiwan. Tel.: +886 3 4638800 ext. 2367 ext. 7062; Fax: +886 3 4638850; E-mail: thtsai@saturn.yzu.edu.tw.

ISSN 1069-2509/09/$17.00

Ranking potential candidate entities is another important issue. Hu et al. [14] employed a linear model that combines several semantic features to score each candidate in their entity search system. They also proposed a supervised learning approach of estimating the weights associates with these features. Their experiment results showed that the supervised learning approach is much more effective in ranking candidates when the ranking influenced by these semantic features. Therefore, their method is enhanced to apply in the QA task.

In addition, devising an accurate performance measurement for QA is also a problem, especially when two answer candidates are scored equally. Traditional performance metric, such as the top-5 mean reciprocal rank (MRR), may be inaccurate in such cases. An efficient formula for an improved MRR measurement is proposed to reduce the computational complexity in this study. This measurement can also be applied to evaluate QA systems in other domains.

The remainder of this paper is organized as follows: Section 2 contains a review of related works. In Section 3, the system workflow of our QA system, a proposed ranking method, and an improved evaluation measurement are described. Section 4 depicts our datasets, the experiment design, and the experiment results. In Section 5, the experiment results are discussed and the proposed measurement with MRR is compared. Finally, Section 6 summarizes our conclusions.

2. Related works

2.1. Information retrieval systems

2.1.1. Traditional search systems

The objective of traditional IR systems is to identify documents or passages that are relevant to a query. In general, the criterion used to judge relevance is the existence of query terms in the documents or passages. The terms are usually weighted according to their occurrences in the documents or passages by using weighting models, such as TF-IDF [23], BM25 [22], and the Language Model [20], which are unsupervised and do not require labeled data for training.

In contrast, a relatively new trend in IR is to employ supervised learning methods to train ranking functions. Herbrich et al. [12] formulated the IR problem as an ordinal regression model, and proposed a method for training the model on the basis of SVM. Gao et al. [14] conducted discriminate training on a linear IR model and observed a significant improvement in the accuracy of document retrieval as a result.

Aside from the supervised approach introduced in the last paragraph, rule-based approaches are also widely used in many domains. Besnard et al. [1] use an ontology in the form of an IS-A hierarchy to capture explanations based on causal statements. Chen et al. [4] employed a rule-based dependency models for security protocol analysis. Tao et al. [25] proposed a novel computational model for solving retrieval problems by constructing and mining a personalized ontology based on world knowledge and a user's Local Instance Repository. According to these recent achievements in using rule-based approach, this is another possible way to handle IR problems. There are also some works inspire us during our research. Gao et al. [10] describes a new approach to the use of clustering for automatic data detection in semi-structured web pages. Pera et al. [19] provide a correlation-based phrase matching (CPM) model and a fuzzy compatibility clustering (FCC) model. CPM can detect RSS news articles containing phrases that are the same as well as semantically alike, and dictate the degrees of similarity of any two articles. FCC identifies and clusters non-redundant, closely related RSS news articles based on their degrees of similarity and a fuzzy compatibility relation. Yusof et al. [32] proposed a model which combines the functional and structural information to facilitate software component search and retrieval

2.1.2. Entity search systems

Before discussing factoid question answering, a similar task is described – entity search. Entity search (also known as expert search) systems try to identify entities that are strongly associated with query terms. The most studied type of entity is *people*, which has been addressed by [3,6]. However, existing entity search methods only exploit simple features and traditional ranking methods. Hu et al. [14] employed a supervised learning technique to train an entity search model. The results of experiments on several data sets indicate that the method significantly outperforms methods based solely on the co-occurrence of terms.

2.1.3. Question answering systems

The first large-scale evaluation of QA systems was hosted by the Text Retrieval Conference (TREC) in 1999 [30], where the task focused on responding to open domain questions with short passages of 50 to 250 words. In 2003, the evaluation task became more chal-

lenging because it required systems to provide exact answers without redundant information [30].

Given a collection of documents, a QA system should be able to retrieve answers for questions posed in natural language. QA systems are categorized according to the questions they can deal with. One type of question is the factoid question, for which the answer consists of a short factual tidbit of information, such as a date, a location, or a person/organization name.

Usually, a factoid QA system transforms a natural language question into keywords, which are sent to an IR engine to retrieve search results. It then extracts possible answers from the returned documents, and rank them using natural language processing (NLP) features like shallow/full parse, tokenization, and part-of-speech (POS) tagging.

The following are two examples of systems that incorporate the above techniques and steps, as well as several others. These systems, constructed by the Language Computer Corporation (LCC) and the National University of Singapore (NUS), were the most successful participants in recent TREC QA tracks. The LCC system [11] uses the COGEX Logic Prover to verify and extract any lexical relationships between a question and its candidate answers. It achieved the best top-1 MARR (71.3%) at TREC-14. The NUS system, developed by Sun et al. [7] utilizes syntactic information and semantic information generated by the MiniPar dependency parser [16] and the ASSERT semantic role labeler [21], respectively. Since the MiniPar parser does not work well with web documents, the system uses semantic role information to extract reliable answer candidates. Finally, it employs dependency-relation-based answer ranking to verify if the web answer is correct for the context. The NUS system achieved second place (66.6%) in top-1 MRR in TREC-14.

2.2. Traditional performance measurement

There are two commonly used measurements for QA system performance measurement: the top-1 accuracy and the top-5 mean reciprocal rank (MRR). For a question set Q, the, top-1 accuracy reports the average accuracy of the top-1 answers for all the questions. It is defined as follows:

$$\text{top-1 accuracy} = \text{\# of correct top-1 answers / } |Q|$$

The top-5 mean reciprocal rank (MRR) [30] is calculated as follows:

$$r_i = \begin{cases} rank(q_i), rank(q_i) \leqslant 5 \\ \infty, rank(q_i) > 5 \end{cases}$$

$$RR(q_i) = \frac{1}{r_i}$$

$$MRR(Q) = \frac{\sum_{q \in Q} RR(q)}{|Q|},$$

where q_i is the ith question; and *rank*(q_i) is the rank of the first correct answer on the list of answer candidates for q_i. In the first formula above, top-1 MRR uses 1 in place of 5.

However, there may be many answer candidates with the same score and all in the leading five places. This results in multiple answer candidate sequences with different RR scores for a question, but all based on the same score. Although selecting a candidate sequence at random and calculating its RR to represent all sequences may solve this problem, the true system performance may be overestimated or underestimated. To address this problem, a new measurement called the average reciprocal rank (ARR) is proposed and discussed in Section 3.3.

3. Biomedical question answering system

3.1. System workflow

The proposed QA system, BeQA, is comprised of four components, namely Question Processing, Passage Retrieval, Candidate Extraction and Feature Generation, and Answer Ranking, which are described in detail in the following sub-sections.

3.1.1. Question processing

Question processing transforms natural language questions into search keywords and extracts features for answer ranking. In our work, question processing involves five steps: named entity (NE) recognition (NER) [9,24,28,29], semantic role labeling (SRL) [15, 26], question classification, and query modification.

The NER step extracts named entities (NEs), such as protein and gene names, from the original question. Then, the SRL step extracts predicates (e.g., the main verb) and corresponding arguments (e.g., noun phrases) from the question. Both the NEs and the SRL information will be transformed into features and used by the answer ranking module, which is described in the Method section.

In the question classification step, hand-crafted patterns are used to identify the target NE type, such as protein, cell, DNA, and RNA, requested by the question (i.e., the answer's NE type). The classified NE type is then sent to the ranking module, which filters out unmatched answer candidates. In the phrase chunking

step, questions are parsed by the GENIA Tagger [3]. Each word in the remaining phrases is then examined and eliminated if it appears on the stop word list. The remaining phrase segments are sent to the passage retrieval module as keywords for a Google search.

Query modification is used to improve recall for queries where Google returns pages. First, using WordNet [18] and Longman's dictionary [2], queries are expanded to generate a list of synonyms and other tenses for the main verb of the question. Then, the web search with the expanded query terms is repeated. If there are still no returned pages, keywords begin to be removed to improve recall.

3.1.2. Passage retrieval

The passage retrieval module is a Google-interfacing program that sends queries to Google and retrieves a collection of web pages, which are sent to the answer extraction module. In the passage retrieval stage, to avoid unnecessary noise, only pages from Google's index of the PubMed database on the NCBI website are retrieved.

3.1.3. Candidate extraction and feature generation

In this stage, two extraction technologies, NER and SRL, are utilized to extract candidate NEs and their corresponding features. NER extracts named entities for answer candidates, and generates features to help match a query with passages containing relevant NEs. The NERBio [28] is employed to identify four types of NE: protein, DNA, RNA, and cell. Biomolecular events in nominal form (e.g., protein expressions), in which the relevant NEs are involved, are also extracted. In our system, each candidate is output with the sentence that contains it, and the sentence is treated as its supporting evidence.

An SRL system – BIOSMILE [8,26] is developed to generate semantic features for answer ranking. SRL can recognize the predicate of a sentence and its corresponding argument phrases, such as the agent, recipient, and location. The argument types recognized by our SRL component and their descriptions are listed in Table 1.

The SRL step also verifies whether answer candidates extracted by our NER component are the expected type. By comparing a candidate's semantic argument type with the expected type, many incorrect candidates can be eliminated to improve the overall accuracy. All the entity candidates along with their features are input to the answer ranking module after completion of the extraction step.

Table 1
Argument types and their descriptions

Type	Description
Arg0	Agent
Arg1	Direct object / Theme / Patient
Arg2-5	Not fixed
ArgM-NEG	Negation marker
ArgM-LOC	Location
ArgM-TMP	Time
ArgM-MNR	Manner
ArgM-EXT	Extent
ArgM-ADV	General-purpose
ArgM-PNC	Purpose
ArgM-CAU	Cause
ArgM-DIR	Direction
ArgM-DIS	Discourse connectives
ArgM-MOD	Modal verb
ArgM-REC	Reflexives and Reciprocals
ArgM-PRD	Marks of secondary predication

3.1.4. Answer ranking

Each NE extracted in the previous step is treated as an answer candidate. The answer ranking module is responsible for calculating each candidate's score. A linear model is employed to calculate a candidate's score based on its features. To estimate feature weights more precisely, a supervised weight tuning procedure, which is described in the next section, is proposed.

3.2. Method

3.2.1. Linear answer ranking method

To calculate an answer candidate's score, a new ranking method called linear answer ranking is proposed. It uses a linear function (combination of features) to calculate the weighted sum of the candidate's features. Each candidate c identified in the candidate extraction step is represented as a binary feature vector $\mathbf{f}_c$. The ith dimension of $\mathbf{f}_c$ (f_{c_i}) indicates whether or not c meets the criterion of the binary feature function f_i, which has a corresponding weight w_i. Therefore, the score of a candidate c is calculated as follows:

$$score(c) = \mathbf{f}_c \bullet \mathbf{w} = \sum_i f_{c_i} w_i,$$

where $\mathbf{w}$ is the weight vector that corresponds to $\mathbf{f}_c$.

3.2.2. Tuning feature weights

To improve the ranking results, a weight tuning procedure is applied. First, the procedure generates all possible weight combinations of the eight features, whose weights have integer values between 1 and 10. This yields 10^8 different combinations. To avoid generating too many weight vectors with the same score, top-

5 MRR, rather than the top-1 accuracy, is utilized to measure the weight of each vector.

Next, for each of the top 20 weight vectors, new vectors are created by adjusting the weights upward or downward by 0.5, or by leaving the weight of a given vector unchanged. This produces $3^n - 1$ new vectors, (n denotes the number of dimensions). The process is then repeated with an upward or downward adjustment of 0.25, and the algorithm iterates repeatedly until the weight decrement reaches 0.125. The combination of eight weights with the highest top-5 MRR scores is taken as the final feature weight set.

3.2.3. Features

The proposed BeQA system employs eight features: Verb_Match (f_{VM}), Argument_Match (f_{ARGM}), NE_Match (f_{NEM}), NE_Similarity (f_{NES}), KeyWord_Similarity (f_{KWS}), Argument_Similarity (f_{ARGS}), Consecutive_Word_Match (f_{CWM}), and Google_Reciprocal_Rank (f_{GRR}). Answer candidate is denoted as c, the query as q, the sentence containing c as s, and the page containing s as$p$. The eight feature types are defined as follows:

$$f_{\mathrm{VM}}(c) = \begin{cases} 1 \text{ if c's verb matches q's main verb} \\ 0 \text{ otherwise} \end{cases}$$

$$f_{\mathrm{ARGM}}(c) = \begin{cases} 1 \text{ if c's semantic role matches the target role} \\ 0 \text{ otherwise} \end{cases}$$

$$f_{\mathrm{NEM}}(c) = \begin{cases} 1 \text{ if c's NE type matches the target type} \\ 0 \text{ otherwise} \end{cases}$$

$$f_{\mathrm{NES}}(c,q,s) = \frac{\text{\# of NEs in } s \text{ that match NEs in } q}{\text{\# of NEs in } q}$$

$$f_{\mathrm{KWS}}(c,q,s) = \text{\# of keywords in } s \text{ that match keywords in } q \,/\, \text{\# of keywords in } q$$

$$f_{\mathrm{ARGS}}(c,q,s) = \text{\# args in } s \text{ that match arguments in } q \text{ excluding the target argument} \,/\, \text{\# args in } q \text{ exclusing the target argument}$$

$$f_{\mathrm{CWM}}(q,s) = \text{\# of consecutive words in } s \text{ that match consecutive words in } q \,/\, \text{\# of keywords in } q$$

$$f_{\mathrm{GRR}}(p) = \text{ the Google reciprocal rank of } p$$

The first three features listed above are binary features. The next three represent the similarity between q and s. The seventh feature denotes keywords adjacent to c match that those of q; and the last feature is p's Google reciprocal rank. The values of the last five features range between 0 and 1.

3.3. An improved evaluation measurement

As mentioned in Section 2.2, the traditional QA measurement approach may overestimate or underestimate a QA system's performance. In this section, a new measurement, top-k ARR, which avoids the problem, is described.

The top-k ARR score is the average of all possible ranked lists' top-k MRR scores. It is defined as follows:

$$r_i(s) = \begin{cases} rank(q_i), rank(q_i) \leqslant k \\ \infty, rank(q_i) > k \end{cases}$$

$$\text{top} - k\ ARR(q_i) = \frac{\sum_{s \in S} \frac{1}{r_i(s)}}{|S|}$$

$$\text{top} - k\ MARR(Q) = \frac{\sum_{q \in Q} \text{top} - k\ ARR(q)}{|Q|},$$

where S is the set of all possible ranked lists, including all the answer candidates for q_i; and s is one list in S.

To further explain the differences between top-k MRR and MARR, the following example is used to compare the results of the RR and ARR methods. Suppose there are three answer candidates A, B, and C, all of which have the highest score. However, only A is the correct answer candidate. Using the RR measurement, different ranked list containing A, B, and C are produced. There are $3! = 6$ lists, as shown in Table 2. The RR score for each list is shown in the last column, and the way to obtain it is introduced in Section 2.2. In this case, the QA system has a one-sixth probability of getting one of these sequences. Consequently, each run of multiple experiments may produce different evaluation results.

However, by using top-k ARR, all the top-k RR scores can be summed, and then divide by $|S|$. Therefore, the result is

$$\text{top-5 ARR}(q_i) = (1+1+1/2+1/3+1/2+1/3)/6 = 11/18,$$

which is a fixed value. In contrast to the RR method, ARR can evaluate the QA systems' performance precisely.

Table 2
All possible sequences

Sequence	Top 1	Top 2	Top 3	RR
1	**A**	B	C	1
2	**A**	C	B	1
3	B	**A**	C	1/2
4	B	C	**A**	1/3
5	C	**A**	B	1/2
6	C	B	**A**	1/3

However, the above ARR method has two limitations. (1) If the value of $|S|$ is very large, calculating the ARR score directly is inefficient; for example, if 170 answer candidates have the highest score, 170! permutations need to be expanded totally. (2) Technologically, there are no numerical data types that can fit such a large value.

To solve the above problems in calculating the ARR score, the following efficient formula is proposed in the [17]:

$$ARR(q_i) = \sum_{r}^{\min(r+m-1,5)} \frac{n(m-r)!(m-n)!}{r(m-n-r+1)!m!}$$

where m represents the number of answer candidates with the same score; n denotes the number of correct answer candidates with the same score; and r indicates the highest rank over all correct answer candidates in all possible ranked list.

Although the above formula provides an efficient way to calculate the ARR score, it still has a problem. Consider the situation of r larger than 1 and m is equal to n (This is possible since maybe all of the answer candidates with same score are correct), then the item $m-n-r+1$ at the denominator will be less than zero, making its factorial undefined. To fix this problem, this formula is revised as follows:

$$\text{top}-k\ ARR(q_i) = \sum_{t=r}^{\min(r+(m-n),k)} \frac{n(m-n)!(m-(t-r)-1)!}{t(m-n-(t-r))!m!}$$

Using the above formula, the top-5 ARR score is calculated as follows ($r = 1$, $m = 3$, $n = 1$):

$$\sum_{t=1}^{3} \frac{1(3-1)!(3-(t-1)-1)!}{t(3-1-(t-1))!3!} = 11/18.$$

Here, the score is equal to that of previous ARR method. It provides a convenient way of calculating a large number of answer candidates with the same score.

Table 3
Thirty common biomolecular verbs

activate	phosphorylate	express	mediate	promote
affect	decrease	increase	modulate	reduce
alter	differentiate	induce	mutate	regulate
associate	transactivate	inhibit	encode	repress
bind	dnhance	interact	prevent	signal
stimulate	suppress	block	transform	trigger

4. Results

4.1. Dataset

To the best of our knowledge, there are no well-established online factoid QA systems dedicated to the biomolecular domain. Hence, it is difficult to obtain a representative set of user queries for use as a benchmark. To create a question set, biologists in our laboratory referred to the TREC Genomics Track [13] to choose appropriate abstracts and generate candidate questions.

An independent committee composed of several other biologists then selected 400 questions from among the generated candidates. Next, the questions are divided into two sets, a development set and a test set, each containing 200 questions. The answer types of the 400 biomolecular event questions cover four NE classes, namely protein, DNA, RNA, and cell (including cell line and cell type). Furthermore, each question is based on one of 30 common biomolecular verbs selected by Chou et al. [27], shown in Table 3.

Six example selected questions with SRL annotations are shown as follows:

1. [$_{\mathrm{R-Arg0}}$Which protein] [$_{\mathrm{predicate}}$increases] [$_{\mathrm{Arg1}}$levels of active nuclear NF-kappa B complex]?
2. [$_{\mathrm{R-AM-LOC}}$In which type of cell] does [$_{\mathrm{Arg0}}$human immunodeficiency virus type 1 Nef protein] [$_{\mathrm{predicate}}$inhibit] [$_{\mathrm{Arg1}}$NF-kappa B induction]?
3. [$_{\mathrm{R-Arg1}}$The transcription of which gene] is [$_{\mathrm{predicate}}$ enhanced] [$_{\mathrm{Arg0}}$by recombinant OTF-2 protein]?
4. [$_{\mathrm{AM-LOC}}$In human T lymphocytes,] [$_{\mathrm{R-Arg1}}$which protein] is [$_{\mathrm{predicate}}$induced] [$_{\mathrm{Arg0}}$by ALD]?
5. [$_{\mathrm{R-Arg0}}$Which protein] [$_{\mathrm{predicate}}$regulates] [$_{\mathrm{Arg1}}$ monocyte migration and activation]
6. [$_{\mathrm{R-Arg1}}$Which mRNA] is [$_{\mathrm{predicate}}$increased] [$_{\mathrm{Arg0}}$ by EBNA-2 expression in Daudi cells]?

Table 4
Comparison of different features

Config.	f_{ARGM}	f_{ARGS}	f_{CWM}	f_{GRR}	top-1 MARR (%)	top-5 MARR (%)
Baseline					57.94	58.07
ARGM	+				63.79	65.37
ARGS		+			62.46	64.16
CWM			+		64.47	66.08
GRR				+	59.71	61.38
ALL	+	+	+	+	74.11	76.68

Table 5
The actual tuned weights

Feature	f_{VM}	f_{NEM}	f_{NES}	f_{KWS}	f_{ARGM}	f_{ARGS}	f_{CWM}	f_{GRR}
Weight	1.0	7.8	2.5	3.0	10.8	1.0	7.7	1.0

Table 6
Comparison of best results in the development and test sets

Dataset	top-1 MARR (%)	top-5 MARR (%)
Development set	74.11	76.68
Test set	55.58	66.99

Table 7
Comparison of the proposed method with two popular ranking methods

Configuration	top-1 MARR (%)	top-5 MARR (%)
BM25	30.00	40.00
Simple voting	10.70	22.70
Our method	55.58	66.99

4.2. Experiment design and results

The following three experiments are designed to evaluate the performance of BeQA. In each experiment, the query results of all configurations are cached to eliminate the influence of updates on Google's index.

The first experiment uses the development set to tune the best weight combination for the proposed features, and then applies the combination to the test set. The second experiment compares the proposed ranking method with two popular ranking methods. Finally, an experiment is conducted to assess the impact of the number of pages returned by a Google search.

4.2.1. Experiment 1

Verb Match(f_{VM}), NE Similarity(f_{NES}), NE Match (f_{NEM}), and Key-Word Similarity(f_{KWS}) are taken as the features of the baseline system. To assess the contribution of each feature, its related features are compared by adding Argument Match(f_{ARGM}), Argument Similarity(f_{ARGS}), Consecutive Word Match(f_{CWM}) and Google Reciprocal Rank(f_{GRR}) to the baseline configuration individually. The five configurations are Baseline, ARGM, ARGS, CWM, and GRR. Furthermore, all the features are incorporated into the sixth configuration, ALL. Table 4 shows the performance comparison of f_{ARGM}, f_{ARGS}, f_{CWM}, f_{GRR} and ALL. When applied individually, f_{CWM} and f_{ARGM} are the most effective features; f_{ARGS} can also improve the performance when used alone or with other features. By incorporating all the features, the top-1 MARR and top-5 MARR results are 74.11% and 76.68%, respectively.

For the development set, the actual tuned weights determined by our tuning procedure for the ALL configuration are shown in Table 5.

In Table 6, the actual tuned weights are applied to the test set, and the performance of the best configurations in the development and test sets are compared.

4.2.2. Experiment 2

For the comparison experiment, two traditional ranking functions are used: BM25 [22] and simple-voting [14]. BM25 is commonly used as an information retrieval function to rank passages. In this experiment, the function is used along with heuristics to rank NEs. The BM25 function is utilized to score the passages in the beginning. Then, the highest ranked passage containing at least one NE of the target type is examined, and the NE closest to the predicate is considered the top-1 answer. The simple-voting method ranks entities based on the number of supported passages. As shown in Table 7, our QA system significantly outperforms the compared methods.

4.2.3. Experiment 3

To explore the impact of maximum returned page (MRP), the performance of all configurations in MRP is examined, ranging from 2 to 14. As shown in Fig. 1, most target entities relevant to the query can be found in the first 13 pages. The top-1 MARR and top-5 MARR values increase slowly after the maximum number of returned pages reaches 13.

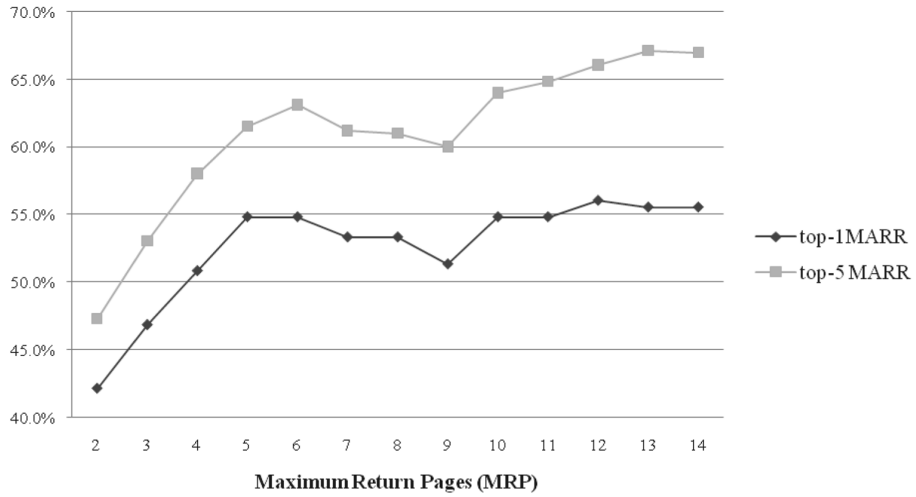

Fig. 1. top-1 and top-5 MARR over MRP.

5. Discussion

Compared to BM25, which depends solely on co-occurrence statistics, our method ranks candidates more intuitively and naturally with semantic information. In this section, some examples are provided to demonstrate the effectiveness of each proposed syntactic or semantic feature, and an additional experiment that compares MRR with MARR is described.

5.1. The effects of using f_{CWM}

Question:
Which protein inhibits the synthesis of Ig mRNA?
Answer:

Baseline	CWM	Passage
16	20.17	These findings demonstrate that [*TGF-beta*] decreases B lymphocyte Ig secretion by inhibiting the synthesis of Ig mRNA and inhibiting the switch from the membrane form to the secreted forms of mu and gamma mRNA.
16	18.50	Transforming growth factor-beta suppresses [*human B lymphocyte Ig*] production by inhibiting synthesis and the switch from the membrane form to the secreted form of Ig mRNA.

The f_{CWM} feature reinforces the keyword match feature by considering consecutive matches between questions and passages. The experiment result shows that f_{CWM} increases accuracy by approximately 7%. The above example is to demonstrate f_{CWM}'s effectiveness. The first and second columns show the scores of the Baseline and CWM configurations, respectively. The third column shows answer candidates (the phrases in brackets) and their passages. In the baseline configuration, both candidates achieve a score of 16. However, after adding f_{CWM}, the first candidate achieves a better score than the second. This is because the first passage's CWM value is higher than that of the second. In the first passage, there is a consecutive five-word match, "the synthesis of Ig mRNA", which is underlined. In the second passage, the length of the consecutive match "Ig mRNA" is only three words. This example demonstrates that f_{CWM} is very useful for disambiguating candidates with similar contexts.

5.2. The effects of using f_{ARGM}

Here an example is given to illustrate how f_{ARGM} significantly enhances the performance of the top-1 and top-5 MARR. In the question, "Which protein interacts with the alpha subunit of TFIIA?", the semantic roles are tagged in addition to named entity tagging, so the question becomes:

> $[_{\mathrm{R-Arg0}}$Which protein$]$ $[_{\mathrm{predicate}}$interacts$]$ $[_{\mathrm{Arg1}}$with the alpha subunit of TFIIA$]$?

In this question, the target role is Arg0 because "which" locates at R-Arg0. Therefore, our QA system searches for an Arg0 protein. The following is the answer sentence corresponding to the above question:

> First, $[_{\mathrm{Arg0}}$Tax$]$ was found to $[_{\mathrm{predicate}}$interact$]$ $[_{\mathrm{Arg1}}$with the 35-kDa (alpha) subunit of TFIIA$]$ $[_{\mathrm{ArgM-LOC}}$in the yeast two-hybrid interaction system$]$.

Although the requested answer is a protein, our QA system successfully identifies the "Tax" protein rather than "TFIIA" since "Tax" locates at an Arg0 argument, which matches the target role (Arg0).

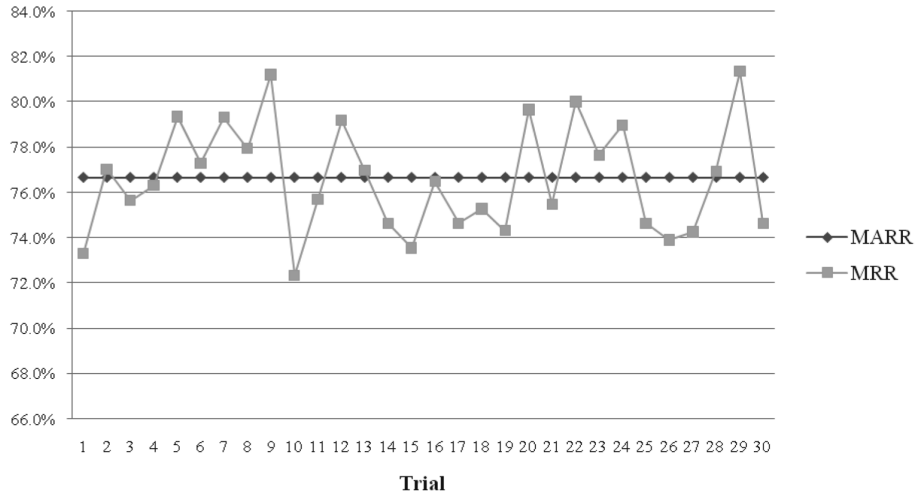

Fig. 2. Comparison between MRR and MARR in 30 trials.

5.3. The effects of using f_{ARGS}

The experiment results show that using f_{ARGS} alone can improve the performance by 4.52% in (top-1 MARR) and 6.09% in top-5 MARR). Using the feature cooperatively with other features (mainly f_{ARGM}) yields similar improvements. Compared to ARGM, ARGS tends to find answer sentences that have predicate-argument structures similar to the question. This finding suggests that these two features work independently. The following is a question and its corresponding answer that exclusively retrieved by employing f_{ARGS}. The question and answer have three common arguments (Arg0, Arg1, and ArgM-LOC).

Question:

[$_{\mathrm{R-Arg1}}$The expression of which protein] is [$_{\mathrm{predicate}}$inhibited] by [$_{\mathrm{Arg0}}$IL-10] [$_{\mathrm{ArgM-LOC}}$in activated human monocytes]?

Answer text:

[$_{\mathrm{Arg0}}$Interleukin-10 (IL-10), like IL-4], is known to [$_{\mathrm{predicate}}$inhibit] [$_{\mathrm{Arg1}}$cytokine expression] [$_{\mathrm{ArgM-LOC}}$ in activated human monocytes].

5.4. Comparing MRR with MARR

To compare MARR with MRR, using the same dataset, 30 additional experiments are conducted on the ALL configuration described in Section 4.2.1. From the results shown in Fig. 2, MARR yields a stable evaluation result, while MRR arbitrarily changes in response to each experiment. The results demonstrate that the proposed method can evaluate any QA system precisely and avoid the same score problem that makes MRR inaccurate.

6. Conclusion

Our proposed QA system provides biologists with another way to obtain the information they need. Compared with general IR systems, which retrieve all possible documents rather than an exact answer, a QA system retrieves a specific answer from a limited number of pages. In this paper, a more reliable method is designed to select the suitable answer from candidates, and incorporates syntactic and semantic features, including NE matching (NEM), verb matching (VM), SRL argument matching (ARGM), NE similarity (NES), keyword similarity (KWS), SRL argument similarity (ARGS), consecutive word matching (CWM), and Google reciprocal rank (GRR). To tune the optimal weight of each feature, a supervised learning algorithm is applied to adjust the feature weights reliably. Our experiments show that with the syntactic feature, CWM, the top-1 and top-5 MARR can be improved by 6.53% (from 57.94% to 64.47%) and 8.01% (from 58.07% to 66.08%,) respectively. Because the SRL system is used to label the semantic structures of input queries and candidate passages, our QA system can benefit from semantic information. For example, with the ARGM feature, the top-1 MARR improves by 5.86% and the top-5 MARR by 7.3%. After combining all the syntactic and semantic features, the proposed BeQA system outperforms traditional ranking functions, such as BM25 and simple voting, by about 25.58% top-1 MARR and 26.99% top5-MARR on the test set. After combining all the syntactic and semantic features, the performance of the BeQA system achieves 74.11% top-1 MARR and 76.68% top-5 MARR.

Notwithstanding our system considers eight features with different weights, it still suffers from the same score problem that affects widely used measurement methods. To resolve the problem, a new measurement called the average reciprocal rank (ARR), which is the average of all possible RR score sequences, is proposed. However, expanding all permutations to calculate the ARR is inefficient, so an efficient formula is further proposed and the equality of the results is demonstrated.

In future work our plans are to (1) increase the variety of answer types by including more NE classes, such as diseases and viruses; (2) expand our corpus sources from short abstracts to full papers or other authoritative biomedical digital libraries; and (3) link the extracted answers directly to other databases or resources to provide biologists with related information in a fast and efficient manner.

Acknowledgments

This research was supported in part by the National Science Council of Taiwan under grant NSC96-2752-E-001-001-PAE and NSC97-2218-E-155-001, as well as the Thematic Program of Academia Sinica under grant AS95ASIA02.

References

[1] P. Besnard, M.O. Cordier and Y. Moinard, Ontology-based inference for causal explanation, *Integrated Computer-Aided Engineering* **15**(4) (2008), 351–367.

[2] B. Boguraev, T. Briscoe, J. Carroll, D. Carter and C. Grover, The derivation of a grammatically indexed lexicon from the Longman Dictionary of Contemporary English, *Proceedings of the 25th conference on Association for Computational Linguistics* (1987), 193–200.

[3] C.S. Campbell, P.P. Maglio, A. Cozzi and B. Dom, Expertise identification using email communications, *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (2003), 528–531.

[4] Q. Chen, S. Zhang and Y.P.P. Chen, Rule-based dependency models for security protocol analysis, *Integrated Computer-Aided Engineering* **15**(4) (2008), 369–380.

[5] K.B. Cohen and L. Hunter, Natural Language Processing and Systems Biology, *Artificial Intelligence Methods and Tools for Systems Biology* (2005).

[6] G.V. Cormack and T.R. Lynam, Statistical precision of information retrieval evaluation, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2006), 533–540.

[7] H. Cui, R. Sun, K. Li, M. Kan and T. Chua, *Question Answering Passage Retrieval using Dependency Relations* (2005), 400–407.

[8] H.-J. Dai, C.-H. Huang, R.T.K. Lin, R.T.-H. Tsai and W.-L. Hsu, BIOSMILE web search: a web application for annotating biomedical entities and relations, *Nucleic Acids Research* **36**(Web Server issue) (2008), W390–W398.

[9] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning and G. Sinclair, Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web, *Joint Workshop on Natural Language Processing in Biomedicine and Its Applications at Coling* (2004).

[10] X. Gao, L.P.B. Vuong and M. Zhang, Detecting data records in semi-structured web sites based on text token clustering, *Integrated Computer-Aided Engineering* **15**(4) (2008), 297–311.

[11] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl and P. Wang, *Employing Two Question Answering Systems in TREC-2005*, in The Fourteenth Text REtrieval Conference (TREC-14), 2005.

[12] R. Herbrich, T. Graepel and K. Obermayer, Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers: MIT Press, Pages, 2000.

[13] W. Hersh, A.M. Cohen, P. Roberts and H.K. Rekapalli, TREC 2006 genomics track overview, *The Fifteenth Text Retrieval Conference* (2006).

[14] G. Hu, J. Liu, H. Li, Y. Cao, J.Y. Nie and J. Gao, A Supervised Learning Approach to Entity Search, *Lecture Notes in Computer Science* **4182** (2006), 54.

[15] Y. Kogan, N. Collier, S. Pakhomov and M. Krauthammer, Towards Semantic Role Labeling & IE in the Medical Literature, *AMIA Annual Symposium Proceedings* **2005** (2005), 410.

[16] D. Lin, *Dependency-based Evaluation of MINIPAR*, in.Workshop on the Evaluation of Parsing Systems, 1998.

[17] R.T.K. Lin, J. Liang-Te Chiu, H.J. Dai, M.Y. Day, R.T.H. Tsai and W.L. Hsu, *Biological Question Answering with Syntactic and Semantic Feature Matching and an Improved Mean Reciprocal Ranking Measurement*, 2008, 184–189.

[18] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K.J. Miller, Introduction to WordNet: An On-line Lexical Database*, *International Journal of Lexicography* **3**(4) (2004), 235–244.

[19] M.S. Pera and Y.K. Ng, Utilizing phrase-similarity measures for detecting and clustering informative RSS news articles, *Integrated Computer-Aided Engineering* **15**(4) (2008), 331–350.

[20] J.M. Ponte and W.B. Croft, A Language Modeling Approach to Information Retrieval., *SIGIR-98* (1998), 275–281.

[21] S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J.H. Martin and D. Jurafsky, Support vector learning for semantic argument classification, *Machine Learning* **60**(1) (2005).

[22] S.E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford and A. Payne, *Okapi at TREC-4*, in TREC-95, 1995.

[23] G. Salton, J. Allan and C. Buckley, *Approaches to Passage Retrieval in Full Text Information Systems*, in SIGIR-93, 1993, 49–58.

[24] B. Settles, Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets, *COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (*NLPBA/BioNLP*) (2004).

[25] X. Tao, Y. Li and R. Nayak, A knowledge retrieval model using ontology mining and user profiling, *Integrated Computer-Aided Engineering* **15**(4) (2008), 313–329.

[26] R.T.H. Tsai, W.C. Chou, Y.S. Su, Y.C. Lin, C.L. Sung, H.J. Dai, I.T.H. Yeh, W. Ku, T.Y. Sung and W.L. Hsu, BIOSMILE: A semantic role labeling system for biomedical verbs using a

maximum-entropy model with automatically generated template features, *BMC Bioinformatics* **8**(1) (2007), 325.

[27] R.T.H. Tsai, H.J. Dai, H.C. Hung, R.T.K. Lin, W.C. Chou, Y.S. Su, M.Y. Day and W.L. Hsu, BESearch: A Supervised Learning Approach to Search for Molecular Event Participants, *The 2007 IEEE International Conference on Information Reuse and Integration* (2007), 412–417.

[28] R.T.H. Tsai, C.L. Sung, H.J. Dai, H.C. Hung, T.Y. Sung and W.L. Hsu, NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition, *BMC Bioinformatics* (2007).

[29] Y. Tsuruoka, Bidirectional inference with the easiest-first strategy for tagging sequence data, *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (2005), 467–474.

[30] E. Voorhees and D. Tice, The TREC-8 question answering track evaluation, *Text Retrieval Conference TREC* **8**(2000).

[31] X. Yang, J.S. Guodong Zhou and C.L. Tan, Improving Noun Phrase Coreference Resolution by Matching Strings, *Natural Language Processing-IJCNLP 2004: First International Joint Conference* (2005), 22–31.

[32] Y. Yusof and O.F. Rana, Combining structure and function-based descriptors for component retrieval in software digital libraries, *Integrated Computer-Aided Engineering* **15**(4) (2008), 279–296.