BESearch: A Supervised Learning Approach to Search for Molecular Event Participants

Richard Tzong-Han Tsai[1], Hong-Jei Dai[1], Hsi-Chuan Hung[1], Ryan T.K. Lin[1],
Wen-Chi Chou[1], Ying-Shan Su[2], Min-Yuh Day[1], and Wen-Lian Hsu[1],
[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.
[2]Institute of Human Nutrition, Columbia University, New York 10032

Abstract

Biomedical researchers rely on keyword-based search engines to retrieve superficially relevant documents, from which they must filter out irrelevant information manually. Hence, there is an urgent need for a more efficient system to help them rapidly locate specific molecular events and the participants involved in these events. In this paper, we propose a novel search system with a new search interface and answer ranking scheme. Due to the limited number of query types in the Biomedical-specific searches, we employ a form-based interface with various query templates for specifying required information. This can ascertain a user's intentions more accurately than a conventional keyword-based interface. Ranking is another key issue in this type of search. We propose a linear ranking model, trained by a supervised learning algorithm, which combines different features. Two semantic features, named entity types and semantic roles, are incorporated into the model to help match a query with entities in relevant documents. After employing all effective semantic features, our system achieves a Top-1 accuracy of 43.1% and Top-5 MRR of 47.1%. In comparison with the baseline system, Top-1 accuracy and Top-5 MRR increase by 9.5% and 7.1%.

1. Introduction

When planning a research project, molecular biologists are primarily interested in relevant molecular pathways and underlying mechanisms [1]. Since molecular biology is a rapidly developing and changing field, it is essential that researchers are able to obtain accurate search results from newly published literature. Currently, most researchers use keyword-based search engines such as PubMed and Google [2]. However, with the tremendous amount of new biomedical literature being published and the increasing complexity of molecular pathway descriptions, it is becoming harder to find specific and relevant information about molecular interactions using these tools. Keyword-based information retrieval (IR) is more suitable for finding broadly relevant documents, rather than more specific information in those documents. When biologists want to know exactly which proteins are involved in a pathway, they still have to put in a great deal of manual effort to locate the desired terms. To avoid this situation, users must provide more specific and complex information. Even so, a keyword-based interface limits users to describing semantic information about phrases and relations between words and phrases.

There are two other types of information search systems. The first is question answering (QA), which allows users to input well-formed natural language questions and returns concise and clear answers, such as factoid answers, a list of answers, and definitional answers. The main problem with the QA approach stems from the flexible nature of its queries. So far, it has proved difficult for state-of-the-art natural language processing (NLP) technologies to correctly analyze input questions and output candidates; thus, the accuracy of QA systems is far from satisfactory. The situation is even worse in molecular biology due to wide variation in the names of biomedical entities.

The second type comprises keyword-based entity search systems [3, 4]. When given a list of keywords and an NE type, such systems return a ranked list of named entities corresponding to the specified type. There is increasing interest in entity search, and many such systems have been implemented, especially in the newswire and business domains. However, in the biomedical domain, researchers need to input more specific information, such as the interaction in which the entity participates, or the role the entity plays in this interaction. It is not feasible to convey these constraints using keyword-based interfaces.

Due to the limited number of query types along with complex input information in the above mentioned searches, form-based interfaces are more appropriate than keyword-based interfaces. They may be used to ensure that queries are entered in a canonical (i.e., unambiguous) form. Many information systems, especially database systems, use such interfaces. In addition, they are also used by internet search engines. Askjeeves[1] is a successful example of such interfaces applied to question answering.

Ranking potential candidate entities is another important issue. [10] employed a linear model that combines several semantic features to score each candidate. They also proposed a supervised learning approach for estimating the weights associated with these features. Their experiment

[1] http://www.ask.com

results showed that the supervised learning approach is much more effective in ranking candidates when the ranking is influenced by these semantic features.

In this paper, we propose a form-based entity search system that can return a ranked list of the entities participating in a specific molecular event. These forms provide sufficiently detailed slots (and options in each slot) for users to specify their search demand. We incorporate two effective semantic features, named entity types and semantic roles, to help match the query with relevant information contained in retrieved documents.

2. Related work

2.1 Traditional search systems

The objective of conventional information retrieval systems is to identify documents or passages that may be relevant to a query. The criterion used to judge the relevance is the appearance of query terms in the documents or passages. The terms are usually weighted by using models such as TF-IDF [5], BM25 [6], and Language Model [7]. Term weighting methods do not usually require labeled data for training. In this sense, the methods are unsupervised. There is also a new trend in IR that employs supervised learning methods to train ranking functions. [8] formulates the IR problem as an ordinal regression model, and proposes a method for training the model on the basis of SVM. [9] conducted discriminate training on a linear IR model and observed a significant improvement in the accuracy of document retrieval as a result.

2.2 Entity search systems

Entity search systems try to identify entities that are strongly associated with query terms. The most studied type of entity is people (also known as expert search), which has been addressed by [3, 4]. However, existing entity search methods only exploit simple features or traditional IR techniques for ranking. Many features may be useful for entity search, including new features that are not used in traditional IR. Therefore, an appropriate approach to entity search should be able to incorporate new features easily. [10] employed a supervised learning method to train an entity search model. The experimental results indicate that the method significantly outperforms methods based solely on co-occurrences.

2.3 Question answering systems

The first large-scale evaluation of QA systems was hosted by the Text Retrieval Conference (TREC) in 1999 [11]. The task focused on responding to open domain questions with short passages of 50 to 250 words. After several years of evolution, the evaluation task became more challenging, requiring exact answers without redundant information [11].

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

Generally, a factoid QA system can transform a natural language question into keywords, send the keywords to an IR engine, retrieve the search results, extract possible answers from the returned documents, and rank them using NLP features like shallow/full parsing, tokenization, and part-of-speech 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 systems in recent TREC QA tracks. The LCC system [12] uses the COGEX Logic Prover to verify and extract any lexical relationships between a question and its candidate answers. It also incorporates eXtended WordNet, SUMO, and other resources in its knowledge base. The LCC system achieved the best top-1 accuracy (71.3%) in TREC-14. The NUS system (Sun et al., 2005) utilizes syntactic and semantic relations, produced by the MiniPar dependency parser [13] and the ASSERT semantic role labeler [14], respectively. To overcome the sparsity of keywords in short questions, the NUS system retrieves related documents from the Web and performs keyword expansion according to the syntactic parsing results. Since its parser does not work well with web documents, it 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 the second place (66.6%) in top-1 accuracy in TREC-14.

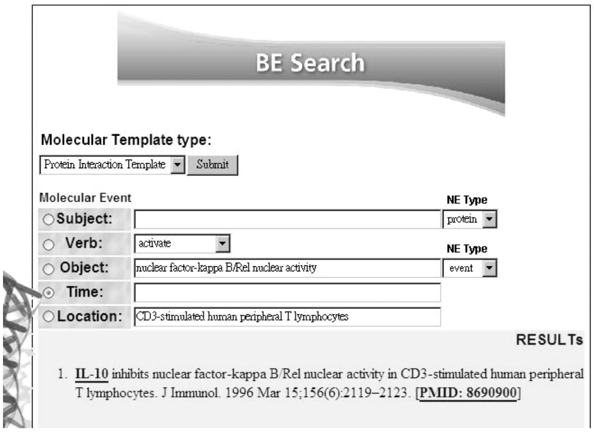
3. System architecture

In this section, we describe the five main components of our form-based biomedical entity search engine, namely, the search interface, the query construction module, the passage retrieval module, the candidate extraction and feature generation module, and the ranking module.

3.1 Search interface

In BESearch, users specify their query by filling query templates. Each template represents one molecular event type. It contains one main verb and several additional arguments (semantic roles). One of these arguments must be specified as the target. Each argument may be designed to either have both a text field and an NE type dropdown list or only a text field. The text field allows users to specify a phrase for the argument. For the target argument, the text field should be left as blank. The NE type dropdown list is used for specifying the NE type of this argument. In BE-Search, the NE type should be protein, DNA, RNA, cell, or molecular event phrase that contains the above four NE types (e.g., protein expression).

The query template shown in Figure 1 represents the protein-protein interaction event. We can see that the main verb is “activate”; the target is the subject and must be a protein name; the object is “nuclear factor-kappa B/Rel nuclear activity” and is specified as an event; the location argument is “CD3-stimulated human peripheral T lymphocytes”; the time argument is not specified.



Molecular Template type: Protein Interaction Template

Molecular Event

○ Subject: NE Type protein

○ Verb: activate

○ Object: nuclear factor-kappa B/Rel nuclear activity NE Type event

◉ Time:

○ Location: CD3-stimulated human peripheral T lymphocytes

RESULTs

1. IL-10 inhibits nuclear factor-kappa B/Rel nuclear activity in CD3-stimulated human peripheral T lymphocytes. J Immunol. 1996 Mar 15;156(6):2119–2123. [PMID: 8690900]

Figure 1. Our form-based query interface

3.2 Query Construction

A query has two usages in our system. First, it is used to retrieve relevant passages. Second, it is used to rank candidate entities. Basically, it is constructed based on the information gathered from the input form.

In the first usage, all input words (except stop words) from the query template are put on a list. These words and the designated verb are then sent to the Google search engine. If Google returns zero pages, query modification will be executed as follows. First, the WordNet and Longman’s dictionary is used to generate a list of synonyms and other tenses for the query’s main verb. Then, the web search is repeated with the expanded query terms.

In the second usage, all text fields are tagged by NERBio [15], our biomedical NE recognizer. All tagged NEs are added to the query’s data structure.

3.3 Passage Retrieval

The passage retrieval kernel is a Google-interfacing program which can send queries to Google and return a collection of documents. The content of retrieved web pages is sent to the answer extraction module. At this stage, we only retrieve pages from Google's index of the PubMed database on the NCBI website to avoid unnecessary noise.

3.4 Candidate Extraction and Feature Generation

This module is responsible for extracting candidate NEs and their corresponding features. It employs two extraction technologies: named entity recognition (NER) and semantic role labeling (SRL). NER is used for extracting candidate NEs. It can also generate features to help match the query with passages containing the relevant NE. Our NER system, which is trained on the JNLPBA training set, can identify four NE types: protein, DNA, RNA, and cell. The F-score of our NER system [15] on the JNLPBA test set is 74.0%. In addition, the molecular events in nominal form (e.g., protein expression), in which these NEs are involved, are also extracted. In our system, each candidate is output with the sentence containing it, which is treated as its supporting evidence.

In addition, we use SRL to generate semantic features for ranking. SRL can recognize the predicate of a sentence and its corresponding argument phrases, such as the agent, patient, and location. The argument types and descriptions are listed in Table 1. Furthermore, it can verify whether answer candidates extracted by NER are of the expected type. The F-score of our SRL system [16], which operates fully automatically, is 69.7% on the GENIA corpus. By comparing a candidate's semantic argument type with the expected type, we can eliminate many incorrect candidates and improve the overall accuracy. All the entity candidates along with their features are delivered to the ranking module after extraction has been completed.

Table 1. Argument types and their descriptions

Type	Regular Expression
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

3.5 Ranking

Extracted NEs and their features are sent to the answer ranking module, where their scores are calculated. The details are given in Section 4.1.

4. Method

4.1 Our linear ranking function

To score a candidate entity, our ranking module uses a linear function (combination of features) to calculate the weighted sum of each candidate's features to score the candidates. 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 if c matches the criterion of the binary feature function f_i, which has a corresponding weight w_i. Therefore, the score of candidate c is calculated as follows:

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

where **w** is the weight vector that corresponds to **f.**

4.2 Evaluation measurement

We report two measurements. Suppose the question set is Q, and there are n questions in Q. The first, top-1 accuracy, basically reports the average accuracy of the top-1 answers of all the questions. It is defined as follows:

top-1 accuracy = # of correct top answers / n

The second measurement is top-5 mean reciprocal rank (MRR). It is defined as follows:

$$\mathrm{RR}(q_i) = \frac{1}{\text{rank of first correct answer for } q_i}$$

$$\mathrm{MRR}(Q) = \frac{\sum_{i=1}^{n} \mathrm{RR}(q_i)}{n}, \text{where } q_i \in Q$$

In addition to evaluating the testing results, MRR is also used for selecting the final weight vector. The details are given in the next section.

4.3 Tuning feature weights

To improve ranking, we perform a weight tuning procedure. The procedure first generates all possible weight combinations of the seven features (details of these features are described in Section 4.4), where the weights have integer values between 1 and 10 inclusively, or 10^7 different combinations. To avoid too many weight vectors with the same score, we use the top-5 MRR as the measurement of each weight vector, instead of the top-1 accuracy.

Next, for each of the top 20 weight vectors, new vectors are created by changing the weights upward or downward by 0.5 or leaving a weight unchanged—for a given vector, this produces 3^n-1 new vectors (n: dimension number). The process is then repeated with an upward or downward change of 0.25, and the algorithm is iterated repeatedly until the weight decrement reaches 0.125. We take the weight vector with the highest top-5 MRR as the final one.

4.4 Features

Our QA system currently employs 7 features: NE_Match (f_{NEM}), Verb_Match (f_{VM}), Argument_Match (f_{ARGM}), NE_Similarity (f_{NES}) and KeyWord_Similarity (f_{KWS}), Argument_Similarity (f_{ARGS}), and Google reciprocal rank (f_{GRR}). We denote the entity candidate as c, the query as q, and the sentence containing c as s, and the page containing s as p. The first three features listed above are binary features and c's properties. The next three are the similarity between q and s, and the last feature is p's Google reciprocal rank. The values of the last four features range between 0 and 1. We define the seven feature types as follows:

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

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

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

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

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

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

f_{GRR} (c, p)= the Google reciprocal rank of p

5. Experiment

5.1 Dataset

To the best of our knowledge, there is no appropriate benchmark for evaluating a biomedical entity search system. Therefore, we asked four biologists to generate candidate queries. An independent committee composed of several biologists takes responsible for selecting 200 queries for training our system and the other 200 for evaluating. Each query is generated based on a sentence that contains at least one of the protein-protein interaction verbs listed in Table 2. All these sentences are randomly selected from a Medline abstract. After the queries are selected, the corresponding relevant entities are labeled by the four biologists. Our queries focused on four classes: protein, DNA, RNA, and cell

(cell line and cell type combined). The numbers of these four answer classes are 279, 71, 5, and 45, respectively.

Table 2. Protein-protein interaction verbs

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

5.2 Experiment design

We designed several experiments to find the best settings for our system. We set the baseline system to include four base features: f_{NEM}, f_{VM}, f_{NES}, and f_{KWS}. The maximum returned pages (MRP) value is initially set to 10. First, we test the effectiveness of our query-modification methods. In order to examine the benefit of using SRL and Google rankings, we further compare the features related to them by adding f_{ARGM}, f_{ARGS}, and f_{GRR}, into the baseline configuration. The baseline and these three configurations are denoted as Baseline, ARGM, ARGS, and GRR, respectively. We then incorporate all the features into the "All" configuration. Next, we use the best configuration of features from the development set on the test set and compare its performance with that of the development set. To further explore the impact of MRP, we examine the performance of all configurations in MRP, ranging from 2 to 14.

5.3 Experiment results

Table 3 shows the improvement brought by using our two query modification methods. The top-1 accuracy and top-5 MRR increase by 14.0% and 14.9%, respectively, in our baseline configuration.

Table 3. Improvement by query modifications

Config.	top-1 Acc. (%)	top-5 MRR (%)
w/o modification	19.6	25.1
with modification	33.6	40.0

Table 4 shows performance comparison of using f_{ARGM}, f_{ARGS}, f_{GRR}, and all features. GRR achieves the same performance as Baseline because f_{GRR}'s weight is 0. When applied individually, f_{ARGM} is the most effective feature. f_{ARGS} can also improve performance when used alone or with other features. With all features incorporated, the top-1 accuracy and top-5 MRR are 43.1% and 47.1%, respectively.

In Table 5, we list the actual weights of the ALL configuration, as determined by our training procedure.

As shown in Figures 2 and 3, most target entities relevant to the query can be found in the first 10 pages. We can see that when MRP is greater than 10, the top-1 accuracy and top-5 MRR values either slow down or stop increasing.

Table 4. Comparison of using different features

Config.	*base*	f_{ARGM}	f_{ARGS}	f_{GRR}	top-1 Acc.	top-5 MRR
Baseline	+				33.6	40.0
ARGM	+	+			41.2	46.2
ARGS	+		+		35.3	41.3
GRR	+			+	33.6	40.0
ALL	+	+	+	+	43.1	47.1

Table 5. Weights of the All configuration

Feature	f_{NEM}	f_{VM}	f_{NES}	f_{KWS}	f_{ARGM}	f_{ARGS}	f_{GRR}
Weight	9.8	2.0	4.2	3.0	10.8	0.8	0

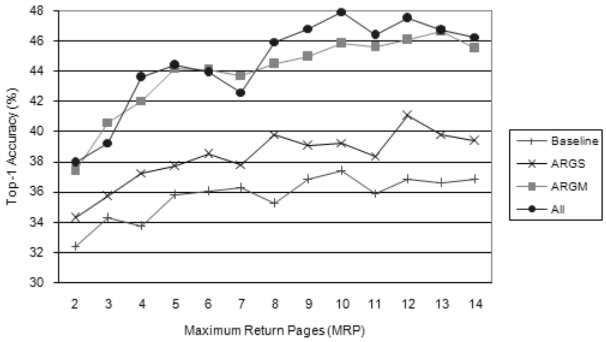


Figure 2. Top-1 accuracy over MRP

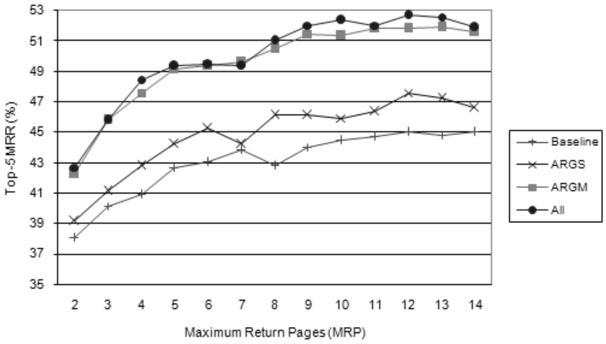


Figure 3. Top-5 MRR over MRP

6. Conclusion

In this paper, we present an entity search system that offers biologists another way to obtain the information they need. We adopt a form-based search interface and design a variety of query templates that allow users to specify required information. Compared to other IR systems with similar functionality (such as question answering systems that require users to input well-formed natural language questions and keyword-based entity search systems), the form-based interface avoids the difficulties of processing natural language input. It also provides more accurate information than keyword-based search approaches due to clear user intention and the limited number of query types in this problem. Ranking is another key issue in this type of search. Most biomedical information retrieval systems use cooccurrence-based approaches, which lack the ability to combine different types of useful semantic features. In contrast, we propose a linear model that incorporates different features and a supervised learning algorithm to train that model. This has the advantage of requiring small amounts of training data, reducing labor-intensive annotation. Two semantic features, NE types and semantic roles, are incorporated into the model. They help match queries with entities in retrieved documents. After employing all effective semantic features, our system significantly increases the Top-1 accuracy and Top-5 MRR by 9.5% and 7.1%, respectively.

7. Acknowledgement

This research was supported in part by the thematic program of Academia Sinica under Grant AS 95ASIA02, the thematic program of Academia Sinica under Grant AS 94B003, and the National Science Council under Center of Excellence Grant NSC 95-2752-E-001-001-PAE.

8. References

[1] K. B. Cohen and L. Hunter, "Natural Language Processing and Systems Biology," in *Artificial Intelligence and Systems Biology, Springer Series on Computational Biology*, W. Dubitzky and F. Azuaje, Eds.: Springer, 2005.

[2] X. Yang, G. Zhou, J. Su, and C. Tan., "Improving Noun Phrase Coreference Resolution by Matching Strings," presented at IJCNLP-2004, 2004.

[3] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, "Expertise Identification using Email Communications," presented at CIKM-03, 2003.

[4] G. V. Cormack and T. R. Lynam, "Statistical Precision of Information Retrieval Evaluation," presented at SIGIR-06, Seattle, Washington, 2006.

[5] G. Salton, J. Allan, and C. Buckley, "Approaches to Passage Retrieval in Full Text Information Systems," presented at SIGIR-93, 1993.

[6] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," presented at TREC-95, 1995.

[7] J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval.," *SIGIR-98*, pp. 275-281, 1998.

[8] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*. Cambridge, MA.: MIT Press, 2000, pp. 115-132.

[9] J. Gao, H. Qi, X. Xia, and J.-Y. Nie, "Linear discriminant model for information retrieval," *SIGIR-05.*, 2005.

[10] G. Hu, J. Liu, H. Li, Y. Cao, J.-Y. Nie, and J. Gao, "A Supervised Learning Approach to Entity Search," 2006.

[11] E. M. Voorhees, "Overview of the TREC 2001 Question Answering Track," presented at TREC-01, 2001.

[12] P. Morarescu, C. Bejan, and S. Harabagiu, "Shallow Semantics for Relation Extraction," presented at IJCAI-05, 2005.

[13] D. Lin, "Dependency-based Evaluation of MINIPAR," presented at.Workshop on the Evaluation of Parsing Systems, 1998.

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

[15] 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*, vol. 7, 2006.

[16] R. T.-H. Tsai, W.-C. Chou, Y.-C. Lin, W. Ku, Y.-S. Su, T.-Y. Sung, and W.-L. Hsu, "BIOSMILE: Adapting Semantic Role Labeling for Biomedical Verbs: An Exponential Model Coupled with Automatically Generated Template Features," presented at BioNLP-2006, New York, 2006.

[17] J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus--semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19 Suppl 1, pp. i180-2, 2003.