Extracting Functional Annotations of Proteins Based on Hybrid Text Mining Approaches

Jung-Hsien Chiang and Hsu-Chun Yu

Department of Computer Science and Information Engineering

National Cheng Kung University, Tainan 701, Taiwan, ROC

jchiang@mail.ncku.edu.tw; yuhc@ismp.csie.ncku.edu.tw

ABSTRACT

In this report, we present the system that we built for task 2 of the BioCreAtIvE competition, which is based on the MeKE (Medical Knowledge Explorer) system [3] developed earlier. Our system combines the high-precision advantage of a pattern matching approach and the little-human-effort advantage of a sentence classification approach, and creates great potential for achieving higher performance than that of using either one of the two approaches.

1. INTRODUCTION

There has been more and more study toward the application of natural language processing techniques to automatically extract knowledge from biomedical literature [4, 9, 11]. Thus the work has mainly focused on identifying gene names [10] and discovering protein-protein interactions [2, 12] from biomedical articles. The BioCreAtIvE evaluation, whose goal is to provide common benchmarks for the performance of natural language processing systems working on biomedical research literature, was thus set up.

The second task of the BioCreAtIve evaluation addresses the functional curation problem, *i.e.* the automatic assignment of GO (Gene Ontology) [13] annotations to human proteins. Full text articles describing protein functions are lengthy and they have complex article structures, hence in contrast with abstracts, full text is rather difficult to be processed. Besides, sentences reporting protein functions appear in various forms, and thus cannot be handled simply with lexicons. In information extraction techniques, a pattern matching approach has the advantage of high precision but also has the disadvantage of too much human intervention [8]. A sentence classification approach needs little human effort but also sacrifices some precision rate [3, 5]. Our system combines these two approaches to merge both the strengths of the two approaches.

In the second section, we describe our methods of extracting protein functions. In the third section, we present our results in the competition and discuss the discovered phenomena.

2. METHODS

In our system, the documents from the JBC [14] and BMC [1] journals are processed through the following procedure:

(1) Sentence detection and indexing
(2) POS (part of speech) tagging, GO term indexing and protein name indexing
(3) Co-occurrence extraction
(4) Phrase parsing
(5) Pattern matching
(6) Template Screening

There are some extra optional steps executed conditionally, which include:

(1) Sentence transform
(2) Sentence classification
(3) GO variant mining

We describe these steps below.

2.1 Sentence Detection and Indexing

The final submission format of the task requires the original text of the document that proves an annotation. Nevertheless, a sentence is tokenized after sentence detection, *i.e.* the original text would be modified. Hence sentences are first indexed by recording their positions in text, so that the system can return to the original text, and provide the evidence text.

2.2 POS Tagging, GO Term Indexing and Protein Name Indexing

For POS tagging, we adopt Grok [6], an open source natural language processing library written in Java, which is part of the OpenNLP [7] project.

For the identification of GO terms and protein names in text, we use an indexing method instead of a tagging one. Since a word in text may match more than one GO term or protein name, even both, a method of directly tagging the names in text does not work. By contrast, an indexing method that records the positions of names in text works much better.

2.3 Co-occurrence Extraction

In this step, the system extracts sentences with the co-occurrence of a protein name and a GO term. These sentences are taken as candidate sentences that describe protein functions. In regard to implementation, since indices of protein names and GO terms are stored in a database, the system executes a SQL statement to query the sentence IDs

belonging to both a protein name index and a GO term index.

2.4 Phrase Parsing

Due to the inefficiency of full parsing sentences in medical documents, we use a shallow parsing method only on those sentences with the co-occurrence of a protein name and a GO term, which are extracted through the co-occurrence extraction step.

The system recognizes noun and verb phrases using finite automata, which model general forms of phrase constructs. Each head noun is associated with its left modifiers to constitute a noun phrase, and each verb is associated with its auxiliaries and adverbs to constitute a verb phrase. The finite automaton used to recognize noun phrases is shown in Figure 1.

After parsing, a sentence is represented as a sequence of phrases, each of which has the following format:

{<phrase type> {{`<token>` <POS> <slot>}+}},

where <phrase type> is noun phrase ("NP"), verb phrase ("VP") or others ("."); <token> is the original token in a sentence, and each protein name or GO term is viewed as a single token; <POS> is the part of speech of the token; <slot> is "P" when this token is a protein name, "G" when it is a GO term, and "." otherwise.

For example, this sentence "These results indicate that Pyk2 is involved in the signal transduction pathway leading to IL-2 production." is parsed as follows:

```
{
{NP {{`These` DT .} {`results` NNS .}}}
{VP {{`indicate` VBP .}}}
{.  {{`that` IN .}}}
{NP {{`Pyk 2` NNP P}}}
{VP {{`is` VBZ .} {`involved` VBN .}}}
{.  {{`in` IN .}}}
{NP {{`the` DT .} {`signal transduction` NNP G}}}
{VP {{`pathway` RB .} {`leading` VBG .}}}
{.  {{`to` TO .}}}
{NP {{`IL` NNP .} {`-` : .}}}
{.  {{`2` LS .}}}
{NP {{`production` NN .}}}
}
```

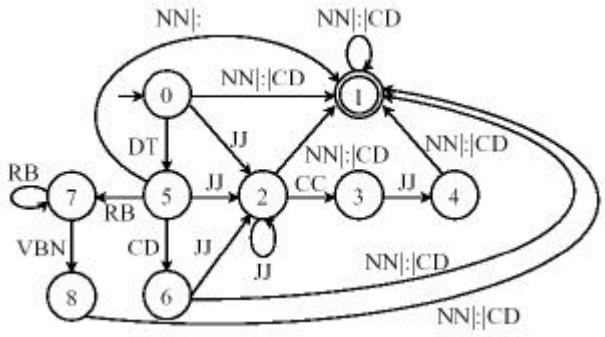

Figure 1. The Finite Automaton for Recognizing Noun Phrases

2.5 Pattern Matching

The first approach we used to extract evidence text of GO annotations for proteins is matching via *phrasal patterns.* Phrasal patterns are used to perform phrase-based pattern matching, which could permit various forms of modifiers in a phrase, so as to achieve robust pattern matching. A phrasal pattern consists of a sequence of phrase constraints, each of which has the same format as that of the phrases of a sentence.

Two pattern examples for the biological process and cellular component types of GO are listed below respectively:

```
{
{NP {{. . P}}}
{VP {{`plays` . .}}}
{NP {{`role` . .}}}
{.  {{. IN .}}}
{NP {{. . G}}}
}

{
{NP {{. . P}}}
{VP {{`localized|colocalizes|immunolocalized` . .}}}
{.  {{. IN|TO .}}}
{NP {{. . G}}}
}
```

2.6 Sentence Transform

Proteins are often reported to relate with more than one function. In such cases, the phrasal patterns usually cannot work. Hence the sentences are transformed firstly, and then pass through the pattern-matching step again to extract templates. Following is an example of the transformation rules:

```
{
{NP {{`both` . .}}}
{.  {{. IN .}}}
{NP {{. . .}}}
{.  {{. CC .}}}
{NP {{. . G}}}
}
->
{
1
4
}
```

where if a sentence matches the condition, it will be transformed by reserving only the 1st and 4th phrases (zero-based). For example, the following phrase-parsed sentence would conform to the above rule:

```
{
{NP {{`PBF` NNP P}}}
{VP {{`is` VBZ .} {`localized` VBN .}}}
{NP {{`both` NN .}}}
{.  {{`in` IN .}}}
{NP {{`the` DT .} {`cytoplasm` NN .}}}
{.  {{`and` CC .}}}
{NP {{`the` DT .} {`nucleus` NNP G}}}
}
```

Hence this sentence is transformed as follows:

```
{
{NP {{`PBF` NNP P}}}
{VP {{`is` VBZ .} {`localized` VBN .}}}
{.  {{`in` IN .}}}
{NP {{`the` DT .} {`nucleus` NNP G}}}
}
```

2.7 Sentence Classification

The second approach we used to extract evidence text of GO annotations for proteins is using a classifier to classify sentences according to whether a sentence describes protein functions or not. We had studied this approach formerly, the details of which are

interpreted in another paper [3].

2.8 GO Variant Mining

It is not sufficient to use GO terms as a lexicon to recognize function descriptions, since article authors describe protein functions in various forms. Consequently, we mine from text those terms that match GO terms partially, and calculate the *edit distance* between each pair of mined term and GO term, i.e. the minimum number of insertions or deletions of tokens necessary to make the two terms equal regardless of the token order. Those terms for which the number of token insertion less than two and the number of token deletion less than two are taken as candidate *GO variants*. For example, *catabolism of phenylalanine* is a variant of *phenylalanine catabolism*. All GO variants mentioned in text are indexed for the purpose of performing the same procedure as that for GO terms.

2.9 Template Screening

For task 2.1, only one evidence text of each annotation can be returned as the submission result, and for task 2.2, there is a number limiting the number of GO term predictions for each annotation. Consequently, we use a screening strategy to select extracted GO annotations and the text that fits each annotation best, which constitute *templates* in an information extraction task. Since we use a mixture of phrasal pattern and sentence classifier approaches to extract GO annotations, and the phrasal pattern approach gets higher precision and lower recall than those of the sentence classifier approach, the screening strategy gives the templates extracted by the first approach higher priority than those extracted by the second approach. Besides, GO annotations extracted by GO variant indexing are less reliable than those extracted by indexing official GO terms, hence they have relatively low priority.

In summary, the template priority, from high to low, for the screening strategy is as follows:

(1) Templates extracted by GO term indexing and the phrasal pattern approach
(2) Templates extracted by GO term indexing and the sentence classifier approach
(3) Templates extracted by GO variant indexing and the phrasal pattern approach
(4) Templates extracted by GO variant indexing and the sentence classifier approach

3. RESULTS AND DISCUSSION

Since the task allows each participant to submit 3 runs of results, we adopt three different extraction strategies to generate three versions of results. Figure 2 demonstrates the three extraction strategies.

Strategy 1, expected to be most accurate, is adopting the phrasal pattern approach alone. Since only one piece of evidence text can be submitted for each annotation, the strategy randomly selects a template for output.

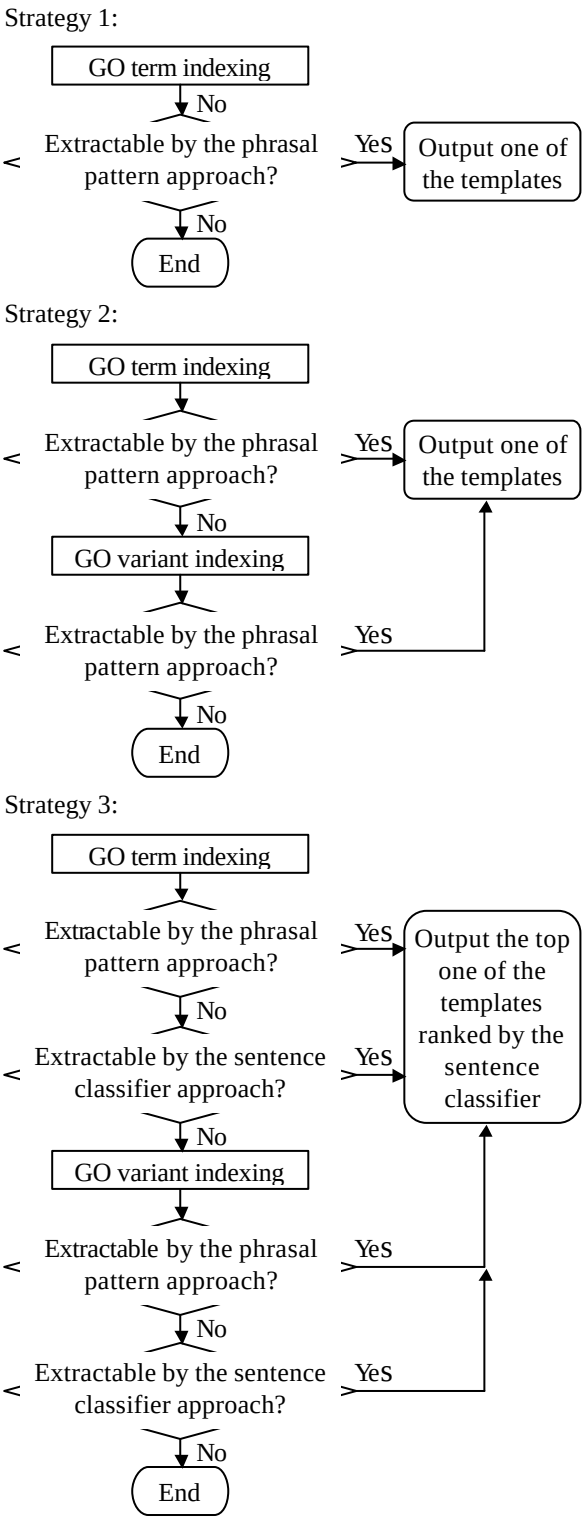

Figure 2. The Three Extraction Strategies We Used in the Competition

Strategy 2, an expansion of Strategy 1, is complementing GO term indexing with GO variant indexing. When an annotation cannot be extracted by GO term indexing, the strategy performs GO variant indexing and then executes the phrasal pattern

approach again for these newly mined GO variants.

Strategy 3, an expansion of Strategy 2, is further complementing the phrase pattern approach with the sentence classifier approach. When an annotation cannot be extracted by the phrasal pattern approach, the strategy executes the sentence classifier approach. In this strategy, we use the scores computed by the sentence classifier to rank the extracted templates, and then the top one is taken as output.

Table 1 shows the results obtained from our system in the competition. In each part of the task, we adopt strategies 1 to 3 in runs 1 to 3 respectively. For part 1, the absolute numbers of perfect and general results for runs 1 to 3 increase progressively, which agrees with the expectation that the recall rates of strategies 1 to 3 should increase progressively. The relative number of perfect results of run 1 is greater than that of run 2, which agrees with expectation, but the corresponding number of run 3 is greatest, which may be the effect of template ranking by the sentence classifier. There is a similar phenomenon for the results of part 2 as well.

From the results, we can know that the combination of the phrasal pattern and sentence classifier approaches is a promising direction of achieving high extraction performance. The GO variant mining step also contributes to raise the recall rate.

Table 1. Results Obtained from Our System in the Competition

Part / Run		Evaluation[1]	Perfect[2]	General[3]
Part 1	Run 1	70	33 (47.14%)	5 (7.14%)
	Run 2	89	41 (46.07%)	7 (7.87%)
	Run 3	251	125 (49.80%)	13 (5.18%)
Part 2	Run 1	28	9 (32.14%)	3 (10.71%)
	Run 2	41	14 (34.15%)	1 (2.44%)
	Run 3	41	14 (34.15%)	1 (2.44%)

[1]Evaluation: the number of results that the evaluators checked.
[2]Perfect: absolute and relative numbers of perfect results, which mean that for both GO and protein the evaluation was correct.
[3]General: absolute and relative numbers of general results, which mean that the protein evaluation was correct and the GO evaluation was generally.

4. CONCLUSION

In this work, we propose a mixture of phrasal pattern and sentence classifier approaches to perform the automatic assignment of GO annotations to proteins. This strategy has been shown to be very advantageous for achieving high performance. We also use the GO variant mining method to search for potential GO variants. This method can broaden the coverage of GO term indexing.

5. REFERENCES

[1] BioMed Central. http://www.biomedcentral.com/.

[2] Blaschke,C., Andrade,M.A., Ouzounis,C. and Valencia,A. (1999) Automatic extraction of biological information from scientific text: protein-protein Interactions. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*. pp. 60-67.

[3] Chiang,J.-H. and Yu,H.-C. (2003) MeKE: discovering the functions of gene products from biomedical literature via sentence alignment. *Bioinformatics*, **19**, 1417-1422. http://ismp.csie.ncku.edu.tw/~yuhc/meke/.

[4] Chiang,J.-H., Yu,H.-C. and Hsu,H.-J. (2004) GIS: a biomedical text-mining system for gene information discovery. *Bioinformatics*, **20**, 120-121. http://iir.csie.ncku.edu.tw/~yuhc/gis/.

[5] Craven,M. and Kumlien,J. (1999) Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*. pp. 77-86.

[6] Grok. http://grok.sourceforge.net/.

[7] OpenNLP. http://opennlp.sourceforge.net/.

[8] Regev,Y., Finkelstein-Landau,M., Feldman,R., Gorodetsky,M., Zheng,X., Levy,S., Charlab,R., Lawrence,C., Lippert,R.A., Zhang,Q. and Shatkay,H. (2002) Rule-based extraction of experimental evidence in the biomedical domain: the KDD Cup 2002 (task 1) *ACM SIGKDD Explorations Newsletter*, **4**, 90-92.

[9] Shah,P.K., Perez-Iratxeta,C., Bork,P. and Andrade,M.A. (2003) Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, **4**:20. http://www.biomedcentral.com/1471-2105/4/20.

[10] Tanabe,L. and Wilbur,W.J. (2002) Tagging gene and protein names in biomedical text. *Bioinformatics*, **18**, 1124-1132.

[11] Tao,Y.-C. and Leibel,R.L. (2002) Identifying functional relationships among human genes by systematic analysis of biological literature. *BMC Bioinformatics*, **3**:16. http://www.biomedcentral.com/1471-2105/3/16.

[12] Temkin,J.M. and Gilder,M.R. (2003) Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, **19**, 2046-2053.

[13] The Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25-29. http://www.geneontology.org/.

[14] The Journal of Biological Chemistry. http://www.jbc.org/.