Enhancing Performance of Protein Name Recognizers Using Collocation

Wen-Juan Hou
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan
wjhou@nlg.csie.ntu.edu.tw

Hsin-Hsi Chen
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan
hh_chen@csie.ntu.edu.tw

Abstract

Named entity recognition is a fundamental task in biological relationship mining. This paper employs protein collocates extracted from a biological corpus to enhance the performance of protein name recognizers. Yapex and KeX are taken as examples. The precision of Yapex is increased from 70.90% to 81.94% at the low expense of recall rate (i.e., only decrease 2.39%) when collocates are incorporated. We also integrate the results proposed by Yapex and KeX, and employs collocates to filter the merged results. Because the candidates suggested by these two systems may be inconsistent, i.e., overlap in partial, one of them is considered as a basis. The experiments show that Yapex-based integration is better than KeX-based integration.

1 Introduction

Named entities are basic constituents in a document. Recognizing named entities is a fundamental step for document understanding. In a famous message understanding competition MUC (Darpa, 1998), named entities extraction, including organizations, people, and locations, along with date/time expressions and monetary and percentage expressions, is one of the evaluation tasks. Several approaches have been proposed to capture these types of terms. For example, corpus-based methods are employed to extract Chinese personal names, and rule-based methods are used to extract Chinese date/time expressions and monetary and percentage expressions (Chen and Lee, 1996; Chen, *et al*., 1998). Corpus-based approach is adopted because a large personal name database is available for training. In contrast, rules which have good coverage exist for date/time expressions, so the rule-based approach is adopted.

In the past, named entities extraction mainly focuses on general domains. Recently, large amount of scientific documents has been published, in particular for biomedical domains. Several attempts have been made to mine knowledge from biomedical documents (Hirschman, *et al*., 2002). One of their goals is to construct a knowledge base automatically and to find new information embedded in documents (Craven and Kumlien, 1999). Similar information extraction works have been explored on this domain. Named entities like protein names, gene names, drug names, disease names, and so on, were recognized (Collier, *et al*., 2000; Fukuda, *et al*., 1998; Olsson, *et al.*, 2002; Rindflesch, *et al*., 2000). Besides, the relationships among these entities, e.g., protein-protein, protein-gene, drug-gene, drug-disease, *etc*., were extracted (Blaschke, *et al.*, 1999; Frideman, *et al*., 2001; Hou and Chen, 2002; Marcotte, *et al*., 2001; Ng and Wong, 1999; Park, *et al*., 2001; Rindflesch, *et al*., 2000; Thomas, *et al*., 2000; Wong, 2001).

Collocation denotes two or more words having strong relationships (Manning and Schutze, 1999). The related technologies have been applied to terminological extraction, natural language generation, parsing, and so on. This paper deals with a special collocation in biological domain – say, protein collocation. We will find out those keywords that co-occur with protein names by using statistical methods. Such terms, which are called *collocates* of proteins hereafter, will be considered as restrictions in protein name extraction. To improve the precision rate at the low expense of recall rate is the main theme of this approach.

The rest of the paper is organized as follows. The protein name recognizers used in this study are introduced in Section 2. The collocation method

we adopted is shown in Section 3. The filtering and integration strategies are explained in Sections 4 and 5, respectively. Finally, Section 6 concludes the remarks and lists some future works.

2 Protein Name Recognizers

The detection of protein names presents a challenging task because of their variant structural characteristics, their resemblance to regular noun phrases and their similarity with other kinds of biological substances. Previous approaches on biological named entities extraction can be classified into two types – say, rule-based (Fukuda, *et al*., 1998; Humphreys, *et al.*, 2000; Olsson, *et al*., 2002) and corpus-based (Collier, *et al*., 2000). KeX developed by Fukuda, *et al*. (1998) and Yapex developed by Olsson, *et al*. (2002) were based on handcrafted rules for extracting protein names. Collier, *et al*. (2000) trained a Hidden Markov Model with a small corpus of 100 MEDLINE abstracts to extract names of gene and gene products.

Different taggers have their specific features. KeX was evaluated by using 30 abstracts on SH3 domain and 50 abstracts on signal transduction, and achieved 94.70% precision and 98.84% recall. Yapex was applied to a test corpus of 101 abstracts. Of these, 48 documents were queried from protein binding and interaction, and 53 documents were randomly chosen from GENIA corpus. The performance of tagging protein names is 67.8% precision and 66.4% recall. While the same test corpus was applied to KeX, it got 40.4% precision and 41.1% recall. It reveals that each tagger has its own characteristics. Changing the domain may result in the variant performance. Consequently, how to select the correct molecular entities proposed from the existing taggers is an interesting issue.

3 Statistical Methods for Collocation

The overall flow of our method is shown in Figure 1. To extract protein collocates, we need a corpus in which protein names have been tagged. Thus, we prepare a tagged biological corpus by looking up the protein lexicon in the first step. Then, common stop words are removed and the stemming procedure is applied to gather and group more informative words. Next, the collocation values of proteins and their surrounding words are calculated. Finally, we use these values to tell which neighbouring words are the desired collocates. The major modules are specified in detail in the following subsections.

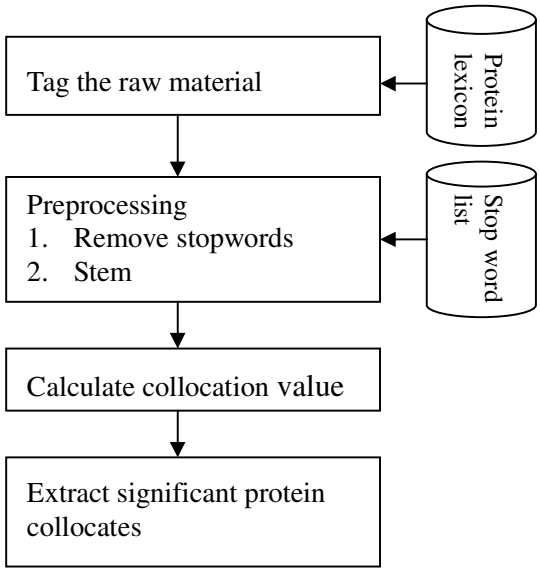

Figure 1. Flow of Mining Protein Collocates

3.1 Step 1: Tagging the Corpus

On the one hand, to calculate the collocation values of words with proteins from a corpus, it is necessary to recognize protein names at first. On the other hand, the goal of this paper deals with performance issue of protein name tagging. Hence, preparing a protein name tagged corpus and developing a high performance protein name tagger seem to be a chicken-egg problem. Because the corpus developed in the first step is used to extract the contextual information of proteins, a completely tagged corpus is not necessary at the first step. Dictionary-based approach for name tagging, i.e., full pattern matching between the dictionary entries and the words in the corpus, is simple. The major argument is its coverage. Those protein names which are not listed in the dictionary, but appear in the corpus will not be recognized. Thus this approach only produces a partial-tagged corpus, but it is enough to acquire contextual information for latter use.

3.2 Step 2: Preprocessing

3.2.1 Step 2.1: Exclusion of Stopwords

Stopwords are common English words (such as preposition "in" and article "the") that frequently appear in the text but are not helpful in discriminating special classes. Because they are distributed largely in the corpus, they should be filtered out. The stopword list in this study was collected with reference to the stoplists of Fox (1992), but the words also appearing in the protein lexicon are removed. For example, "of" is a constituent of the protein name "capsid of the lumazine", so that "of" is excluded from the stoplist. Finally, 387 stopwords were used.

3.2.2 Step 2.2: Stemming

Stemming is a procedure of transforming an inflected form to its root form. For example, "inhibited" and "inhibition" will be mapped into the root form "inhibit" after stemming. Stemming can group the same word semantics and reflect more information around the proteins.

3.3 Step 3: Computing Collocation Statistics

The collocates of proteins are those terms that often co-occur with protein names in the corpus. In this step, we calculate three collocation statistics to find the significant terms around proteins.

Frequency

The collocates are selected by frequency. In order to gather more flexible relationships, here we define a collocation window that has five words on each side of protein names. And then collocation bigrams at a distance are captured. In general, more occurrences in the collocation windows are preferred, but the standard criteria for frequencies are not acknowledged. Hence, other collocation models are also considered.

Mean and Variance

The mean value of collocations can indicate how far collocates are typically located from protein names. Furthermore, variance shows the deviation from the mean. The standard deviation of value zero indicates that the collocates and the protein names always occur at exactly the same distance equal to the mean value. If the standard deviation is low, two words usually occur at about the same distance, i.e., near the mean value. If the standard deviation is high, then the collocates and the protein names occur at random distance.

t-test Model

When the values of mean and variance have been computed, it is necessary to know if two words do not co-occur by chance. Moreover, we also have to know if the standard deviation is low enough. In other words, we have to set a threshold in the above approach. To get the statistical confidence that two words have a collocation relationship, *t*-test hypothesis testing is adopted.

The *t*-value for each word *i* is formulated as follows:

$$t_i = \frac{\bar{x}_i - u_i}{\sqrt{s_i^2 / N}}$$

Where

$N = 4n - 15$,

$\bar{x}_i = \frac{n_count_i}{N}$,

$s_i^2 = p_i \times (1 - p_i)$,

$p_i = n_count_i / n$,

$u_i = p_{protein} \times p_i$, and

$p_{protein}$ is the probability of protein.

When α (confidence level) is equal to 0.005, the value of t is 2.576. In the *t*-test model, if the *t*-value is larger than 2.576, the word is regarded as a good collocate of protein with 99.5% confidence.

3.4 Step 4: Extraction of Collocates

We applied the above procedure to a corpus downloaded from the PASTA website in Sheffield University with 1,514 MEDLINE abstracts [http://www.dcs.shef.ac.uk/nlp/pasta]. Of the 4,782 different stemmed words appearing in the collocation windows, there are 541 collocations generated in Step 3. The collocates are not tagged with parts of speech, so that the output may contain nouns, prepositions, numbers, verbs, *etc.*

The collocates extracted in a corpus cannot only serve as conditions of protein names, but also facilitate the relationship discovery between proteins. From the past papers on the extraction of the biological information, such as Blaschke, *et*

al. (1999), Ng, *et al.* (1999), and Ono, *et al.* (2001) *etc*., verbs are the major targets. This is because many of the subjects and the objects related to these verbs are names of genes or proteins. To assure that the collocates selected in Step 3 are verbs, we assign parts of speech to these words. Appendix A lists the collocates and their variations.

4 Filtering Strategies

For protein name recognition, rule-based systems and dictionary-based systems are usually complementary. Rule-based systems can recognize those protein names not listed in a dictionary, but some false entities may also pass at the same time. Dictionary-based systems can recognize those proteins in a dictionary, but the coverage is its major deficiency. In this section, we will employ collocates of proteins mined earlier to help identify the molecular entities. Yapex system (Olsson *et al*., 2002) is adopted to propose candidates, and collocates are served as restrictions to filter out less possible protein names.

The following filtering strategies are proposed. Assume the candidate set M0 is the output generated by Yapex.

- M1: For each candidate in M0, check if a collocate is found in its collocation window. If yes, tag the candidate as a protein name. Otherwise, discard it.
- M2: Some of the collocates may be substrings of protein names. We relax the restriction in M1 as follows. If a collocate appears in the candidate or in the collocation window of the candidate, then tag the candidate as a protein name; otherwise, discard it.
- M3: Some protein names may appear more than once in a document. They may not always co-occur with some collocate in each occurrence. In other words, the protein candidate and some collocates may co-occur in the first occurrence, the second occurrence, or even the last occurrence. We revise M1 and M2 as follows to capture this phenomenon. During checking if there exists a collocate co-occurring with a protein candidate, the candidate without any collocate is kept undecidable instead of definite no. After all the protein names are examined, those undecidable candidates may be considered as protein names when one of their co-occurrences containing any collocate. In other words, as long as a candidate has been confirmed once, it is assumed to be a protein throughout. In this way, there are two filtering alternatives M31 and M32 from M1 and M2, respectively.

To get more objective evaluation, we utilized another corpus of 101 abstracts used by Yapex [http://www.sics.se/humle/projects/prothalt].
Using the test corpus and answer keys supported in Yapex project, the evaluation results on filtering strategies are listed in Table 1.

Table 1. Evaluation on Filtering Strategies

	Precision	Recall	F-score
M0	70.90%	69.53%	70.22%
M1	79.18%	56.10%	67.64%
M2	79.29%	56.66%	67.98%
M31	81.97%	66.84%	74.41%
M32	81.94%	67.14%	74.54%

Compared with the baseline model M0, the precision rates of all the four models using collocates were improved more than 8%. The recall rates of M1 and M2 decreased about 13%. Thus, the overall F-scores of M1 and M2 decreased about 2% compared to M0. In contrast, if the decision of tagging was deferred until all the information were considered, then the recall rate decreased only 2% and the overall F-scores of M31 and M32 increased 4% relative to M0. The best one, M32, improved the precision rate from 70.90% to 81.94%, and the F-score from 70.22% to 74.54%. That meets our expectation, i.e., to enhance the precision rate, but not to reduce the significant recall rate.

5 Integration Strategies

Now we consider how to improve the recall rates. Integration strategies based on a hybrid concept are introduced. The basic idea is that different protein name taggers have their own specific features such that they can recognize some tagging objects according to their rules or recognition methods. Among the proposed protein names by different recognizers, there may exist some overlaps and some differences. In other words, a protein name recognizer may tag a protein name

that another recognizer cannot identify, or both of them may accept certain common proteins. The integration strategies are used to select correct protein names proposed by multiple recognizers. In this study, we made experiments on Yapex and KeX because they are freely available on the web.

Because protein candidates are proposed by two named entity extractors independently, they may be totally separated, totally overlap, overlapped in between, overlapped in the beginning, and overlapped in the end. Figure 2 demonstrates these five cases.

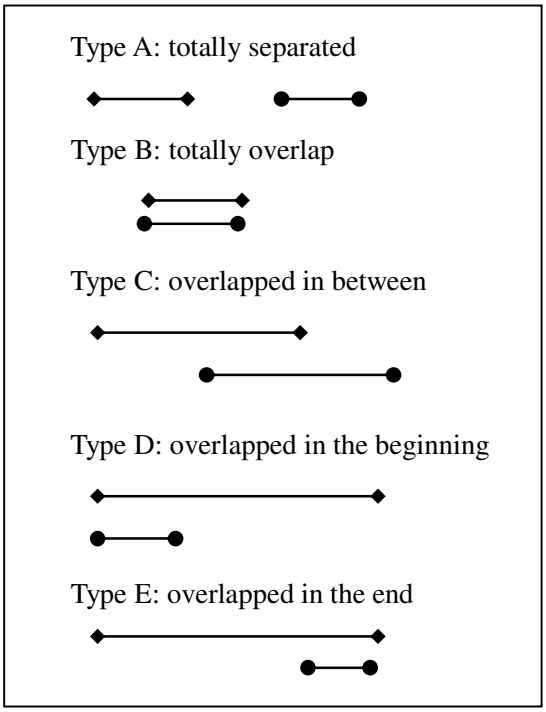

Figure 2. Candidates Proposed by Two Systems

The integration strategies shown as follows combine the results from two sources.

- When the protein names produced from two recognizers are totally separated (i.e., type A), retain them as the protein candidates. This integration strategy postulates that one protein name recognizer may extract some proteins that another one cannot identify.
- When the protein names produced from two recognizers are exactly the same (i.e., type B), retain them as the protein candidates. Because both taggers accept the same protein names, there must exist some special features that fit protein names.
- When the protein names tagged by two taggers have partial overlap (i.e., types C, D and E), two additional integration strategies are employed, i.e., Yapex-based and KeX-based strategies. In the former strategy, we adopt protein names tagged by Yapex as candidates and discard the ones produced by KeX. In contrast, the names tagged by KeX are kept in the latter strategy. The integration strategy is made because each recognizer has its own characteristics, and we do not know which one is performed better in advance.

The above integration strategies put together all the possible protein candidates except the ambiguous cases (i.e., types C, D and E). That tends to increase the recall rate. To avoid decreasing the precision rate, we also employ the collocates mentioned in Section 3 to filter out the less possible protein candidates. Furthermore, to objectively evaluate the performance of the proposed collocates, we employ the same strategies to the same test corpus with some terms suggested by human experts. Total 48 verbal keywords which were used to find the pathway of proteins are used and listed in Appendix B.

Four sets of experiments were designed as follows for Yapex- and KeX-based integration strategies, respectively.

(1)YA and KA: Use the collocates automatically extracted in Section 3 to filter out the candidates as described in Section 4.

(2)YB and KB: Use the terms suggested by human experts for the filtering strategies.

(3)YA-C and KA-C: If Yapex and KeX recommend the same protein names (i.e., type B), regard them as protein names without consideration of collocates. Otherwise, use the collocates proposed in this study to make filtering.

(4)YB-C and KB-C: Similar to (3) except that the collocates are replaced by the terms suggested by human experts.

The experimental results are listed in Tables 2 and 3. The tendency M32>M31>M2>M1 is still kept in the new experiments. The strategy of delaying the decision until clear evidence is found is workable. The performances of YA, YA-C, KA, and KA-C are better than the performances of the

corresponding models (i.e., YB, YB-C, KB, and KB-C). It shows that the set of collocates proposed by our system is more complete than the set of terms suggested by human experts. Compared with the recall rate of M0 in Table 1 (i.e., 69.53%), the recall rates of both Yapex- and KeX-based integration are increased, i.e., 77.52% and 70.60%, respectively. That matches our expectation. However, the precision rates are decreased more than the increase of recall rates. In particular, the F-score of KeX-based integration strategy is 4.70% worse than that of the baseline M0. It shows that KeX performed not well in this test set, so it cannot recommend good candidates in the integration stage. Moreover, the F-scores of M31 and M32 of YA and YA-C are better than that of M0 in Table 1. It reveals that Yapex performed better in this test corpus, so that we can enhance the performance by both the filtering and integration strategies. Nevertheless, the models in Tables 2 and 3 still cannot compete to M32 in Table 1. The reason may be some heuristic rules used in Yapex are modified from KeX (Olsson *et al*., 2002).

Table 2. Evaluation Results on Yapex-based Integration Strategy

YA	Precision	Recall	F-score
M0	61.98%	77.52%	69.75%
M1	64.97%	62.82%	63.90%
M2	65.02%	63.53%	64.28%
M31	65.94%	74.26%	70.10%
M32	65.90%	74.62%	70.26%
YB			
M1	66.79%	44.30%	55.55%
M2	66.79%	44.81%	55.80%
M31	70.20%	65.06%	67.63%
M32	70.19%	65.51%	67.85%
YA-C			
M1	65.76%	69.18%	67.47%
M2	65.88%	69.84%	67.86%
M31	65.39%	75.43%	70.41%
M32	65.38%	75.69%	70.54%
YB-C			
M1	68.92%	58.09%	63.51%
M2	68.78%	58.49%	63.64%
M31	69.07%	69.08%	69.13%
M32	69.07%	69.63%	69.35%

Table 3. Evaluation Results on KeX-based Integration Strategy

KA	Precision	Recall	F-score
M0	60.43%	70.60%	65.52%
M1	63.82%	56.61%	60.22%
M2	63.52%	57.22%	60.37%
M31	64.39%	65.56%	64.98%
M32	64.03%	65.92%	64.98%
KB			
M1	67.56%	41.20%	54.38%
M2	66.99%	41.71%	54.35%
M31	69.57%	55.70%	61.64%
M32	69.25%	56.26%	62.76%
KA-C			
M1	64.72%	63.17%	63.95%
M2	64.44%	63.68%	64.06%
M31	63.83%	66.79%	65.31%
M32	63.49%	67.04%	65.27%
KB-C			
M1	69.57%	55.60%	62.59%
M2	69.15%	56.10%	64.06%
M31	68.36%	60.22%	64.29%
M32	68.09%	60.78%	64.44%

6 Concluding Remarks

This paper shows a fully automatic way of mining collocates from scientific text in the protein domain, and employs them to improve the performance of protein name recognition successfully. The same approach can be extended to other domains like gene, DNA, RNA, drugs, and so on. The collocates extracted from a domain corpus are also important keywords for pathway discovery, so that a systematic way from basic named entities finding to complex relationships discovery can be established.

Applying filtering strategy only demonstrates better performance than applying both filtering and integration strategies together in this paper. One of the possible reasons is that the adopted systems are similar, i.e., both systems are rule-based, and some heuristic steps used in one system are inherited from another. The effects of combining different types of protein name taggers, e.g., rule-based and corpus-based, will be investigated in the future.

Acknowledgements

Part of research results was supported by National

Science Council under the contract NSC-91-2213-E-002-088. We also thank Dr. George Demetriou in the Department of the Computer Science of the University of Sheffield, who kindly supported the resources in this work.

References

Blaschke, C., Andrade, M.A., Ouzounis, C. and Valencia, A. (1999) "Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions," *Proceedings of 7th International Conference on Intelligent Systems for Molecular Biology*, pp. 60-67.

Chen, H.H. and Lee, J.C. (1996) "Identification and Classification of Proper Nouns in Chinese Texts," *Proceedings of 16th International Conference on Computational Linguistics*, pp. 222-229.

Chen, H.H.; Ding, Y.W. and Tsai, S.C. (1998) "Named Entity Extraction for Information Retrieval," *Computer Processing of Oriental Languages*, Special Issue on Information Retrieval on Oriental Languages, **12**(1), 1998, pp. 75-85.

Collier, N., Park, H.S., Ogata, N., Tateishi, Y., Nobata, C. and Ohta, T. (1999) "The GENIA project: Corpus-based Knowledge Acquisition and Information Extraction from Genome Research Papers," *Proceedings of the Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'99)*, June.

Collier, N., Nobata, C. and Tsujii J.I. (2000) "Extracting the Names of Genes and Gene Products with a Hidden Markov Model," *Proceedings of 18th International Conference on Computational Linguistics*, pp. 201-207.

Craven, M. and Kumlien, J. (1999) "Constructing Biological Knowledge Bases by Extracting Information from Text Sources, *Proceedings of 7th International Conference on Intelligent Systems for Molecular Biology*, pp. 77-86.

DARPA (1998) *Proceedings of 7th Message Understanding Conference*.

Fox, C. Lexical Analysis and Stoplists. In *Information Retrieval: Data Structures and Algorithms,* Frakes, W. B. and Baeza-Yates, R., ed., Prentice Hall, 102-130, 1992.

Friedman, C., Kra, P., Yu, H., Krauthammer, M. and Rzhetsky, A. (2001) "GENIES: A Natural Language Processing System for the Extraction of Molecular Pathways from Journal Articles," *Bioinformatics*, **17(S1)**, pp. 74-82.

Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. (1998) "Toward Information Extraction: Identifying Protein Names from Biological Papers," *Proceedings of Pacific Symposium on Biocomputing*, pp. 707-718.

Hirschman, L., Park, J.C., Tsujii, J., Wong, L. and Wu, C.H. (2002) "Accomplishments and Challenges in Literature Data mining for Biology," *Bioinformatics*, **18**(12), pp. 1553-1561.

Hou, W.J. and Chen, H.H. (2002) "Extracting Biological Keywords from Scientific Text," *Proceedings of 13th International Conference on Genome Informatics*, pp. 571-573.

Humphreys, K., Demetriou, G. and Gaizauskas, R. (2000) "Two Applications of Information Extraction to Biological Science Journal Articles: Enzyme Interactions and Protein Structures," *Proceedings of Pacific Symposium on Biocomputing*, **5**, pp. 502-513.

Manning, C.D. and Schutze, H. (1999) *Foundations of Statistical Natural Language Processing*, The MIT Press.

Marcotte, E.M., Xenarios, I. and Eisenberd, D. (2001) "Mining Literature for Protein-protein Interactions," *Bioinformatics*, **17(4)**, pp. 359-363.

Ng, S.-K. and Wong, M. (1999) "Toward Routine Automatic Pathway Discovery from On-line Scientific Text Abstracts," *Proceedings of 10th International Conference on Genome Informatics*, pp. 104-112.

Olsson, F., Eriksson, G., Franzen, K., Asker, L. and Liden P. (2002) "Notions of Correctness when Evaluating Protein Name Taggers," *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 765-771.

Ono, T., Hishigaki, H., Tanigami, A., and Takagi, T. "Automated Extraction of Information on Protein-Protein Interactions from the Biological Literature," *Bioinformatics*, **17(2)**, pp.155-161.

Park, J.C., Kim, H.S., and Kim, J.J. (2001) "Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorial Grammar," *Proceedings of Pacific Symposium on Biocomputing*, **6**, pp. 396-407.

Rindflesch, T.C., Tanabe, L., Weinstein, J.N. and Hunter, L. (2000) "EDGAR: Extraction of Drugs, Genes, and Relations from Biomedical Literature," *Proceedings of Pacific Symposium on Biocomputing*, **5**, pp. 517-528.

Thomas, J., Milward, D., Ouzounis, C., Pulman, S., and Carroll, M. (2000) "Automatic Extraction of Protein Interactions from Scientific Abstracts," *Proceedings of Pacific Symposium on Biocomputing*, **5**, pp. 538-549.

Wong, L. (2001) "PIES, a Protein Interaction Extraction System," *Proceedings of Pacific Symposium on Biocomputing*, **6**, pp. 520-531.

Appendix A. Collocates mined from corpus

act (-, -ed, -ing, -ion, -ive, -ivities, -ivity, -s),

activat (-e, -ed, -es, -ing, -ion, -or) , adopt (-,ed, -s), affect (-, -ed, -ing, -s), allow (-, -ed, -s), analy (-sed, -ses, -sis, -zed, -zing), appear (-, -s), arrange (-d, -ment), assembl (-ing, -y), associat (-e, -ed, -ion), bas (-e, -ed, -is), belong (-, -ing, -s), bind (-, -ing, -s) / bound, bond (-, -ed, -ing, -s), bridge (-, -d, -s), calculat (-ed, -ion), called, carr (-ied, -ier, -ies), cataly (-sed, -ses, -stic, -ze, -zed, -zes, -zing), cause (-, -d, -s), center (-, -ed) / centre (-, -s), chang (-e, -ed, -es, -ing), characteriz (-ation, -e, -ed, -es, -ing), charg (-e, -ed), class (-, -es, -ified, -ifying), cleav (-e, -ed, -es, -ing), clos (-e, -ed, -ing), coil (-, -ed), compar (-e, -ed, -ing, -ison, -isons), complex (-, -ed, -es), composed, compris (-es, -ing), conclu (-de, -ded, -sion, -sions), conserved, consist (-, -ed, -ent, -ing, -s), constitut (-e, -ed, -es), contact (-, -s), contain (-, -ed, -ing, -s), coordinat (-e, -ed, -es, -ion), correlat (-e, -ed), correspond (-, -ing), crystal (-, -lize, -lized, -lizes, -s), cycl (-e, -es, -ing), define (-d, -s), demonstrat (-e, -ed, -es, -ing), depend (-, -ent, -ing), derived, describe (-, -d), design (-, -ed, -ing), detail (-, -ed, -s), determin (-ation, -ations, -e, -ed, -es, -ing), differ (-ence, -ences, -s), diffract (-ing, -ion), digest (-ed, -s), dimer (-, -ic, -ization, -ize), direct (-, -ed, -s), discuss (-, -ed), display (-, -s), disrupt (-, -ed, -ing, -s), effect (-, -s), encod (-e, -ed, -ing), enhanc (-e, -ed, -er, -es, -ing), exhibit (-, -ed, -s), exist (-, -s), explain (-, -ed, -ing, -s), express (-ed, -ing), extend (-, -ed), facilitat (-e, -es, -ing), finding / found, fold (-, -ed, -ing, -s), form (-, -ed, -ing, -s), function (-, -al, -ing, -s), groove (-, -s), hydroly (-sis, -zed, -zes), identif (-ied, -ies, -y), implicat (-e, -ed, -ions), inactiv (-ated, -ates, -e), includ (-ed, -es, -ing), indicat (-e, -ed, -es, -ing), induc (-e, -ed, -es, -ing), inhibit (-, -ed, -ing, -ion, -or, -ors, -s), initiat (-ed, -es), insert (-, -ed, -ing), interact (-, -ing, -ion, -ions, -s), involv (-e, -ed, -es, -ing), isolated, lack (-, -s), lead (-, -ing, -s), ligand (-, -ed, -s), like, link (-, -ed, -ing), located, loop (-, -ing, -s), mediat (-e, -ed, -es, -ing), model (-, -ed, -ing, -s), modul (-ate, -ates, -ating, -e, -es), mutat (-ed, -ions), observ (-e, -ed), obtain (-, -ed), occup (-ied, -ies), occur (-, -red, -s), organiz (-ation, -ed), oxidiz (-ed, -ing), phosphorylate (-d, -s), play (-, -s), position (-, -ed, -ing, -s), predict (-, -ed, -ing), presen (-ce, -ted, -ting), produc (-e, -ed, -es, -ing), promot (-e, -er, -es, -ing), proposed, proton (-, -ated, -s), provid (-e, -ed, -es, -ing), purif (-ied, -y), react (-, -ion, -tive, -s), recogni (-tion, -zed, -zes, -ing), reduc (-ed, -es, -ing, -tase, -tion), refined, regulat (-e, -ed, -es, -ing, -ion, -ory), relat (-ed, -es, -ive), repeat (-, -ed, -s), replaced, report (-, -ed), represent (-, -ed, -ing, -s), requir (-e, -ed, -es, -ing), resembl (-e, -ed, -es, -ing), resol (-ution, -ve), result (-, -ed, -ing, -s), reveal (-, -ed, -s), select (-ed, -ive, -ively), sequence (-, -d, -s), serve (-, -s), shape (-, -d), share (-, -d, -s), show (-, -n, -s), signal (-, -ing, , -ling, -s), sol (-ution, -ved), stabili (sed, -ty, -ze, -zed, -zes, -zing), stimulat (-e, -ed, -es, -ion, -ory), strain (-, -s), strand (-, -ed, -s), structur (-al, -ally, -e, -ed, -es), stud (-ied, -ies, -y, -ying), substitut (-e, -es, -ion, -ions), substrate (-, -s), suggest (-, -ed, -ing, -ion, -s), support (-, -ing, -s), switch (-, -es), synthesi (-s, -ze, -zed), target (-, -ed, -ing, -s), transfer (-, -red), transport (-, -s), understand (-, -ing) / understood, unexpected, us (-e, -ing)

Appendix B. Terms suggested by an expert

accompan (-ied, -ies, -y, -ying), activat (-e, -ed, -es, -ing, -ion, -or, -ors, -ory), affect (-, -ed, -ing, -s), aggregat (-e, -ed, -es, -ing, -ion), assembl (-e, -ed, -es, -ing, -y), associat (-e, -ed, -es, -ing, -ion), attract (-, -ed, -ing, -ion, -s), bind (-, -ing, -s) / bound, catalys (-e, -ed, -es, -ing, -tic), catalyz (-e, -ed, -es, -ing), cluster (-, -ed, -ing, -s), communicat (-e, -ed, -es, -ing, -ion), complex (-, -ed, -es, -ing), construct (-, -ed, -ing, -ion, -s), control (-, -ed, -ing, -led, -ling, -s), cooperat (-e, -ed, -es, -ing, -ion, -or, -ors), correlat (-e, -ed, -es, -ing, -ion), coupl (-e, -ed, -es, -ing), crosslink (-, -ed, -ing, -s), deglycosylat (-e, -ed, -es, -ing, -ion, -ory), demethylat (-e, -ed, -es, -ing, -ion, -ory), dephosphorylat (-e, -ed, -es, -ing, -ion, -ory), effect (-, -ed, -ing, -s), eliminat (-e, -ed, -es, -ing, -ion), enabl (-e, -ed, -es, -ing), enhanc (-e, -ed, -er, -es, -ing), glycosylat (-e, -ed, -es, -ing, -ion, -ory), group (-, -ed, -ing, -s), help (-, -ed, -ing, -s), hinder (-, -ed, -ing, -s), inactivat (-e, -ed, -es, -ing, -ion, -or, -ors, -ory), inhibit (-, -ed, -ing, -ion, -or, -ors, -ory, -s), integrat (-e, -ed, -es, -ing, -ion), interact (-, -ed, -ing, -ion, -s), link (-, -ed, -ing, -s), methylat (-e, -ed, -es, -ing, -ion), obstacl (-e, -ed, -es, -ing), participat (-e, -ed, -es, -ing, -ion), phosphorylat (-e, -ed, -es, -ing, -ion, -ory), prim (-e, -ed, -es, -ing), process (-, -ed, -es, -ing), react (-, -ed, -ing, -ion, -or, -ors, -ory, s), regulat (-e, -ed, -es, -ing, -ion, ,-or, -ory), relat (-e, -ed, -es, -ing, -ion), signal (-, -ed, -ing, , -led, -ling, -s), stimulat (-e, -ed, -es, -ing, -ion, ,-or, -ory), suppress (-, -ed, -es, -ing, -ion), transduc (-e, -ed, -es, -ing, -tion, ,-tor, -tory), trigger (-, -ed, -ing, -s)