NRC Publications Archive
Archives des publications du CNRC

Literature Mining in Molecular Biology
de Bruijn, Lambertus; Martin, Joel

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Proceedings of the EFMI Workshop on Natural Language Processing in Biomedical Applications, pp. 1-5, 2002

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=a9514d78-a9a6-44ed-ac9a-d1e2ac13f1d6
https://publications-cnrc.canada.ca/fra/voir/objet/?id=a9514d78-a9a6-44ed-ac9a-d1e2ac13f1d6

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research
Council Canada

Conseil national
de recherches Canada

Institute for
Information Technology

Institut de technologie
de l'information

NRC·CNRC

*Literature Mining in Molecular Biology**

De Bruijn, B., and Martin, J.
March 2002

* published in: R. Baud and P. Ruch. Proceedings of the EFMI Workshop on Natural Language Processing in Biomedical Applications. Nicosia Cyprus, pp. 1-5, March 8-9, 2002. NRC 45881.

Copyright 2002 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

Literature mining in molecular biology

Berry de Bruijn and Joel Martin
Institute for Information Technology, National Research Council, Canada
berry.debruijn@nrc.ca; joel.martin@nrc.ca

Abstract:

Literature mining is the process of extracting and combining facts from scientific publications. In recent years, many studies have resulted in computer programs to extract various molecular biology findings using Medline abstracts or full text articles. This article describes the range of techniques that have been applied in literature mining. In doing so, it divides automated reading into four general subtasks: text categorization, named entity tagging, fact extraction and collection-wide analysis. Special attention is given to the domain particularities of molecular biology.

Introduction

With an overwhelming amount of biomedical information available as text, it is natural to ask if it can be read automatically. For several decades, natural language processing (NLP) has been applied in biomedicine to automatically 'read' patient records and has resulted in a growing, but fairly homogeneous body of research. Now with the explosive growth of molecular biology research, there is an overwhelming amount of text of a different sort, journal articles. The text collection in Medline can be mined to learn about a subfield, find supporting evidence for new experiments, or add to molecular biology databases.

Literature mining can be compared to reading and understanding literature but is performed automatically by a computer. Like reading, most literature mining projects target a specific goal. In bioinformatics, examples are:

- Finding protein-protein interactions [a.o. 2, 20, 31],
- Finding protein-gene interactions [26],
- Finding subcellular localization of proteins [5, 6],
- Functional annotation of proteins [1, 23],
- Pathway discovery [10, 18],
- Vocabulary construction [19, 24],
- Assisting BLAST search with evidence found in literature [3],
- Discovering gene functions and relations [27].

With this wide variety of goals, it's not surprising that many different tools have been adopted or invented by the various researchers. Although the approaches differ, they can all be seen as examples of one or more stages of a reading process (Fig 1). A reader first selects what they will read, then identifies important entities and relations between those entities, and finally combines this new information with other articles and other knowledge. This reading process forms the backbone of this article.

Before the sections of this article go into detail of the four stages, a few words about the material on which analyses are done. Most of the studies that work with biomedical literature, use Medline abstracts only. This underlines the immense value of the Medline collection. Its size is now approaching 12 million citations, most of which include abstracts. Our hope is that in future years, more and more initiatives will and can be directed towards the full text of articles. A number of publishers now offer free on-line access to full articles and standards in web lay-out and metatagging are finding their acceptance. Algorithms that scale up better and a continuous increase in affordable computing power are - or will be - ready to tackle that.

Free availability of material is at this moment trapped between two forces. There is the growing pressure from the (noncommercial) scientific community to freely share material. But on the other hand there is a growing pressure on companies to make a profit on the web and therefore to regulate access to material. This matter - interesting as it is though - falls outside the scope of this article.

This article introduces a number of studies on literature mining applied to molecular biology, and takes a look at the range of techniques that have been (or could be) applied to modules within the literature mining process. For a more extensive overview of NLP studies applied to molecular biology - as well as to other biomedical domains - see our on-line, partially annotated bibliography at http://textomy.iit.nrc.ca/cgi-bin/BNLPB_ix.cgi .

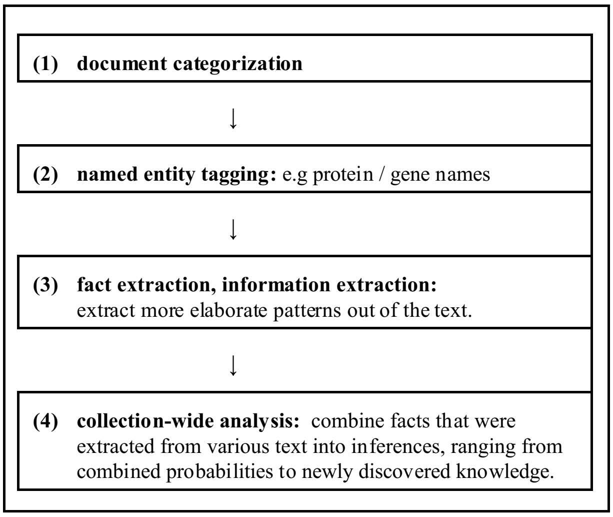
(1) document categorization
↓
(2) named entity tagging: e.g protein / gene names
↓
(3) fact extraction, information extraction: extract more elaborate patterns out of the text.
↓
(4) collection-wide analysis: combine facts that were extracted from various text into inferences, ranging from combined probabilities to newly discovered knowledge.

Figure 1: Text mining as a modular process.

Document categorization

Document categorization, at its most basic, divides a collection of documents into disjoint subsets. This is also known as Document or Text *Classification*, but *categorization* is the most common term. The categories are usually predefined; if they are not, the process is actually document clustering (grouping documents through their superficial characteristics, e.g., [15]). By this definition Information retrieval (IR) is one form of categorization: the collection is divided into two categories of documents, one relevant to the query and one irrelevant. IR algorithms however differ from more specialized categorization algorithms as they use queries rather than teaching from examples.

Document categorization is useful primarily for efficiency reasons. Automated readers, just like human readers, cannot usually spend the time to read all available documents. Having a relevant subset in an early phase can direct future efforts, especially those that are computationally expensive. For example, a text mining system that hunts for subcellular localizations of proteins, might need one minute of processing time per Medline abstract. One can apply that system to all 12 million Medline abstracts and find in retrospect that only, say, 8,900 abstracts returned a valid finding. One could also use a document categorizer that finds, say, 10,000 promising abstracts, and see in retrospect that 8,800 abstracts were useful. A researcher might accept a slight loss of 100 documents with the huge reduction in processing time.

Document categorization can be used to aid human readers by providing a much more accurate, but slower and less flexible, alternative to search engines (e.g. [33]). Other projects explicitly include document categorization but as a module in a larger system (e.g., [19, 30, 35]).

The methods used for document categorization can be borrowed from Machine Learning. Popular methods include Naive Bayes (e.g. [17, 33]), Decision Trees (e.g. [34]), Neural Networks, Nearest Neighbor (e.g. [8]) and Support Vector Machines (e.g. [9, 33]). In all these methods, a collection of precategorized documents is used to train a statistical model of word or phrase use and then the statistical model is applied to uncategorized documents.

Before the training and the actual categorization, there are two preliminary steps: (1) feature extraction, and (2) feature set transformation. The characterizing features of documents can be based on words (most often), word combinations, character sequences or (more rarely) concepts associated with word occurrences. Feature set transformation has two purposes: reducing the size of the feature set, hoping that that will improve efficiency as well as effectiveness, and scaling or weighting the feature set with the purpose of improving the document representation relative to the entire collection. Reduction of the feature set is often done by stemming, eliminating stop words, and eliminating very rare words that burden the classifier more than that they add discrimination power. See for instance [15].

As one example, the Support Vector Machine (SVM) is a relatively new but promising technique for pattern categorization and it has been successfully applied to text (see e.g. [9]). In an SVM, documents are represented as points in a vector space, where the dimensions are the selected features. Based on the training document vectors, the SVM finds the (unique) hyperplane that minimizes the expected generalization error. It does this by maximizing the shortest distance between any of the training examples and the hyperplane. Only some of training vectors will finally define the position of the hyperplane so these are called the 'support vectors'. After the training phase, classification of new documents is a fast process. For biological literature, only few results have been reported. Wilbur [33] used an SVM in combination with a Naive Bayes classifier to construct a boosted system for text categorization. In our own project, we have been applying SVM to various classes of Medline abstracts with good results. A more detailed publication is in preparation, but accuracy is just short of 90% (precision=recall cut-off point). Advantages of SVM include its good and robust performance, and resistance to overfitting the data.

The usual evaluation metric for document categorization tasks is accuracy (in multi-class systems), and the twin-metrics recall and precision (for binary class systems). It is often possible to tweak the system for better precision at the cost of recall or better recall at the cost of precision, so that a task-specific setting can be reached. In evaluation, this makes it possible to plot results in ROC curves. N-fold cross validation is the method of choice for evaluation.

Named entity tagging

The main reason why we read an article is to find out what it says. Similarly, the goal of Information Extraction is to fill in a database record with specific information from the article. The first level of this task is to identify what entities or objects the article mentions. This is called named entity tagging, where

'raw' sentence:	The interleukin-1 receptor (IL-1R) signaling pathway leads to nuclear factor kappa B (NF-kappaB) activation in mammals and is similar to the Toll pathway in Drosophila.
tagged sentence:	The <protein>**interleukin-1 receptor**</protein> (<protein>**IL-1R**</protein>) signaling pathway leads to <protein>**nuclear factor kappa B**</protein> (<protein>**NF-kappaB**</protein>) activation in mammals and is similar to the <protein>**Toll**</protein> pathway in <organism>*Drosophila*</organism>.

Figure 2: an example of named entity tagging on protein and organism names.

the beginning and end of entities might be marked with SGML or XML tags - see fig. 2.

In molecular biology, most of the entities are molecules, such as RNA, genes and proteins, and these entities have many aliases. The lack of naming conventions make this task particularly difficult. Molecule names are invented on a daily basis and conventions, if they exist, may differ between subdisciplines. Two molecules may share names, with only the context to distinguish between the gene and the protein. Even if names are not shared, a substring of an entity name might be a legitimate, but different entity. Tagging 'protein kinase 2' might be an adequate tag in a certain sentence, but 'protein kinase 2 alpha' might be even better.

All techniques suggested for finding named entities use some form of character-by-character or word-by-word pattern to identify the entities. In some of these techniques, the patterns are designed by hand. In others, the patterns are learned from examples that are provided by an expert. Then when a new article is encountered, each string of characters or words is scanned looking for close matches to the learned patterns.

The simplest, manual, approach is to take advantage of *string regularity* and write patterns to capture the known naming conventions, such as a 'p' preceding or succeeding a gene name (see e.g. [11]). Other reliable rules are possible that identify certain words with letters and digits.

A similar approach is *lexicon based* that uses name lists to tag terms, or likely components of entity names [16, 20]. The success of this approach depends on the availability and the coverage of such lists, as well as on their stability over time.

A final manual approach is *context based*. In this method, a dictionary of sentence contexts is compiled that suggest likely molecule names. For instance, in a sentence that shows the pattern "<protein A> inhibits <unknown string>", a rule can dictate that the unknown string is a candidate protein name.

The learning methods, on the other hand, are applied when it is deemed impossible, inaccurate or too slow to manually compile the string regularities and lexicon and context dictionaries. Hishiki et al. [13] use a machine learning module to identify which sequences of n characters are likely to be a part of a molecule name. The most likely ones are the string regularities. New sequences are then scored by the system's past experience with such sequences.

Hidden Markov Models [4] can learn a lexicon and context as well by computing the probability that a sequence of specific words surround or constitute a molecule name. The expert just has to identify examples, while the HMM learns the patterns to apply to new sequences of words.

Of course, the methods do not have to be used in isolation. Friedman et al. [10] used string regularity as well as a lexicon to tag protein and gene names. Also, the methods can be improved by filtering the text. Some researchers prefer to apply part-of-speech tagging to help the Named entity tagging task, so that only (whole) noun phrases are considered as candidate molecule names. The popular part-of-speech taggers or shallow parsers appear to be flexible enough to handle the specialized biological language. For instance, EngCG was used by Hishiki et al. [13] and by Yakushiji et al. [36].

For protein name tagging, accuracies as high as around 95% have been reported [11], but care should be given to the test set composition. It is known that for some organisms or some protein subdomains, the nomenclature is fairly rigidly standardized and excellent tagging accuracy can be reached there. Likewise, experiments with lower results should not be discarded without close scrutiny of the application domain: it might be that the study concentrates on a trickier problem.

Fact extraction

Readers do not understand text if they merely know the entities. They must also grasp the interactions or relationships between those entities. Fact extraction is the identification of entities and their relations. To have a machine do this correctly for arbitrary relationships would require a full natural language intelligence, something that is many years away. There are several approximations that have been tried, from purely statistical co-occurrence to imperfect parsing and coreference resolution.

The simplest approach to capture entity relationships is to search for sentences that mention two entities of the right sort frequently enough. For example, the frequent cooccurrence of two specific protein names with a verb that indicates a molecular interaction might be enough to guess the existence of such an interaction. Craven [5] had his system find sentences where a protein name occurred together with a subcellular location. The effect of accidental cooccurrence could be minimized by requiring frequent corroboration of any pairing.

Another approach that increases the reliability of discovered relationships searches for fixed regular linguistic templates [20, 31]. For example, the system might search for a specific interaction verb while verifying that the surrounding context is parsable in a correct syntactic structure and with entity names in the allocated positions - taking any (negative) modifiers into account - and only then assume the interaction between the substances to be sufficiently proven. The main disadvantage of this approach is that the templates must be constructed manually. Also, many relationships that do not match the template will be missed, but a few good patterns (even when they have low recall) might extract a good number of facts out of a large corpus.

Some linguistic templates can be learned, for instance using a Hidden Markov Model [22]. This requires a corpus with annotated patterns - something that is harder to find or more labour-intensive to construct than a named entity annotated corpus. The expert must mark both the entities and which of several relations applies between those entities. There are clear advantages, no need to explicitly craft rules, better 'portability', and possibly greater overall recall.

Finally, even though automated understanding is not fully possible, important relationships can be discovered by performing a full syntactic parse, where relations between syntactic components are inferred [25, 36]. This approach is similar to the template searching except that it is not domain specific and attempts to identify many or all relationships in a sentence. Park [21] illustrates the syntactical complexities and pitfalls of sentences in biomedical documents.

As an alternative to developing a literature mining system from scratch, some groups have adapted systems or modules of earlier developed systems. They were originally conceived for other bioinformatics tasks (Jake, Kleisli [18, 35]), for other medical domains (e.g. MedLEE [10] or for general use (e.g. Highlight [31], LaSIE [14]).

Collection wide analysis

Thinking new thoughts and using what is known, requires integrating information between documents. This opens the door to knowledge discovery, where combined facts form the basis of a novel insight. The well-known Swanson study [28, 29] on the relation between Raynauds disease and fish oil, was a starting point of formal literature-based knowledge discovery. Weeber et al. [32] discuss an automated replication of that study and similar ones.

Other studies have addressed knowledge discovery in molecular biology (see [5, 10]). As an example: from document 1 you were able to extract relation A implies B; from document 2 you deduced that B implies C. So you might want to study whether A implies C, for which you have found no previous evidence in the literature.

Blaschke et al. [2] used a large number of automatically extracted facts on protein-protein interactions to automatically graph an interaction map for the drosophila cell cycle. This is one illustration where the system abstracts many articles and leaves it to the researcher to make inferences based on the output graph.

Less ambitious goals have still benefitted from collection-wide analyses. One notable application is using collection redundancy to compensate for recall limitations of both statistical and structural methods (e.g. [5]). A high precision/fair recall algorithm such as the typical structural one should have a pretty good confidence in any fact that did get extracted. Facts that were missed in one document might get extracted from another if the fact is redundant. If higher recall with fair precision algorithm is achieved - something that statistical methods tend to do - the combined confidence from various redundant instances might be enough to accept an extracted fact (e.g. [1]).

Apart from findings from other documents in the collection, external sources might help the text analysis. Analogous to clinical settings where medical thesauri and classification schemes (MeSH, ICD, Snomed, ULMS) are used to support text algorithms, database structures in biology (such as GenBank, SwissProt) can be applied towards the correct analysis of abstracts or full text. Craven [5] used Yeast Protein Database data, Krauthammer [16] used BLAST for protein name tagging; Hatzivassiloglou [12] mentions validation across other publications and existing knowledge.

With higher hopes on collection-wide analyses, the scalability of algorithms becomes a more urgent issue. Considering the current size of Medline (close to 12 million articles) and the growing corpus on molecular biology, practical algorithms should scale up well. The ever-increasing power of computers helps in that respect too.

Concluding remarks

With such a wide variety of current applications and techniques, the future is only likely to be even wider open. On-line access of molecular databases will augment the knowledge component in literature mining systems. Large-scale statistical methods will continue to challenge the position of the more syntax-semantics oriented approaches, although both will hold their own place. Literature mining systems will move closer towards the human reader, supporting subtasks of reading in a more interactive and flexible way. For instance by doing text categorization and named entity tagging on-the-fly, working with training material that can easily be edited and augmented.

Written language will always remain only semistructured -and we see that as a benefit. Literature mining adds to written language the promise of making translations onto structures that we do not yet foresee. Therefore, these methods will continue to be fruitful even when some of the molecular biology challenges are solved.

References

1. Andrade MA, Valencia A: Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system. Proc Int Conf Intell Syst Mol Biol 1997;5(2):25-32.
2. Blaschke C, Andrade MA, Ouzounis C, Valencia A: Automatic extraction of biological information from scientific text: protein-protein interactions. Proc Int Conf Intell Syst Mol Biol 1999;30A(2):60-7.
3. Chang JT, Raychaudhuri S, Altman RB: Including biological literature improves homology search.

Pac Symp Biocomput 2001;24(1):374-83.

4. Collier, Nobata and Tsujii: Extracting the names of genes and gene products with a Hidden Markov Model.COLING 2000 conference proceedings, pp. 201-207
5. Craven M: Learning to Extract Relations from MEDLINE. AAAI-99 Workshop on Machine Learning for Information Extraction - July 19, 1999, Orlando Florida
6. Craven M, Kumlien J: Constructing biological knowledge bases by extracting information from text sources. Proc Int Conf Intell Syst Mol Biol 1999:77-86.
7. de Bruijn B, Martin J, Wolting C, and Donaldson I: Extracting sentences to justify categorization. ASIST Annual Meeting, Washington DC, Nov 2001 pp 460-457
8. de Bruijn LM, Hasman A, Arends JW: Automatic SNOMED classification--a corpus-based method. Comput Methods Programs Biomed 1997 Sep;54(1-2):115-22.
9. Dumais S: Using SVMs for text categorization. IEEE Intelligent Systems 13(4), 1998 pp 21-23.
10. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A: GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001 Jun;17 Suppl 1:S74-S82.
11. Fukuda K, Tamura A, Tsunoda T, Takagi T: Toward information extraction: identifying protein names from biological papers. Pac Symp Biocomput 1998;33(2):707-18.
12. Hatzivassiloglou V, Duboue PA, Rzhetsky A: Disambiguating proteins, genes, and RNA in text: a machine learning approach. Bioinformatics 2001 Jun;17 Suppl 1:S97-S106.
13. Hishiki T, Collier N, Nobata C, Okazaki-Ohta T, Ogata N, Sekimizu T, Steiner R, Park HS, Tsujii J: Developing NLP Tools for Genome Informatics: An Information Extraction Perspective. Genome Inform Ser Workshop Genome Inform 1998;9:81-90.
14. Humphreys K, Demetriou G, Gaizauskas R: Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. Pac Symp Biocomput 2000;12(4):505-16.
15. Iliopoulos I, Enright AJ, Ouzounis CA: Textquest: document clustering of Medline abstracts for concept discovery in molecular biology. Pac Symp Biocomput 2001:384-95.
16. Krauthammer M, Rzhetsky A, Morozov P, Friedman C: Using BLAST for identifying gene and protein names in journal articles. Gene 2000 Dec 23;259(1-2):245-52.
17. Marcotte EM, Xenarios I, Eisenberg D: Mining literature for protein-protein interactions. Bioinformatics 2001 Apr;17(4):359-363.
18. Ng SK, Wong M: Toward Routine Automatic Pathway Discovery from On-line Scientific Text Abstracts. Genome Inform Ser Workshop Genome Inform 1999;10:104-112.
19. Ohta Y, Yamamoto Y, Okazaki T, Uchiyama I, Takagi T: Automatic construction of knowledge base from biological papers. Proc Int Conf Intell Syst Mol Biol 1997;5(2):218-25.
20. Ono T, Hishigaki H, Tanigami A, Takagi T: Automated extraction of information on protein-protein interactions from the biological literature. Bioinformatics 2001 Feb;17(2):155-61.
21. Park JC: Using combinatory categorial grammar to extract biomedical information. IEEE Intelligent Systems16(6):62-67.
22. Ray S, Craven M: Representing Sentence Structure in Hidden Markov Models for Information Extraction. IJCAI 2001:1273-1279.
23. Renner A, Aszodi A: High-throughput functional annotation of novel gene products using document clustering. Pac Symp Biocomput 2000:54-68.
24. Rindflesch TC, Hunter L, Aronson AR: Mining molecular binding terminology from biomedical text. Proc AMIA Symp 1999;34(12):127-31.
25. Rindflesch TC, Tanabe L, Weinstein JN, Hunter L: EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac Symp Biocomput 2000:517-28.
26. Sekimizu T, Park HS, Tsujii J: Identifying the Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts. Genome Inform Ser Workshop Genome Inform 1998;9:62-71.
27. Shatkay H, Edwards S, Wilbur WJ, Boguski M: Genes, themes and microarrays: using information retrieval for large-scale gene analysis. Proc Int Conf Intell Syst Mol Biol 2000;8:317-28.
28. Swanson DR.: Fish oil, Raynaud's syndrome, and undiscovered public knowledge. Perspect Biol Med 1986 Autumn; 30(1):7-18
29. Swanson DR.: Medical literature as a potential source of new knowledge. Bull Med Libr Assoc 1990 Jan;78(1):29-37
30. Tanabe L, Scherf U, Smith LH, Lee JK, Hunter L, Weinstein JN: MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling. Biotechniques 1999 Dec; 27(6): 1210-1217.
31. Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M: Automatic extraction of protein interactions fromscientific abstracts. Pac Symp Biocomput 2000:541-52.
32. Weeber M, Klein H, Aronson AR, Mork JG, de Jong-van den Berg LT, Vos R: Text-based discovery in biomedicine: the architecture of the DAD-system. Proc AMIA Symp 2000;35(20): 903-7.
33. Wilbur WJ: Boosting naive Bayesian learning on a large subset of MEDLINE. Proc AMIA Symp 2000:918-22.
34. Wilcox A, Hripcsak G: Classification algorithms applied to narrative reports. Proc AMIA Symp 1999;20(1-2):455-9.
35. Wong L: PIES, a protein interaction extraction system. Pac Symp Biocomput 2001:520-31.
36. Yakushiji A, Tateisi Y, Miyao Y, Tsujii J: Event extraction from biomedical papers using a full parser. Pac Symp Biocomput 2001:408-19.