Semantic Annotation of Biomedical Literature using Google

Rune Sætre[1], Amund Tveit[1,3], Tonje S. Steigedal[2] and Astrid Lægreid[2]

[1] Department of Computer and Information Science,
[2] Department of Cancer Research and Molecular Medicine,
[3] Norwegian Center for Patient Record Research,
Norwegian University of Science and Technology,
NO-7491 Trondheim, Norway
[4] {rune.saetre,amund.tveit}@idi.ntnu.no
{tonje.strommen,astrid.laegreid}@medisin.ntnu.no

Abstract. With the increasing amount of biomedical literature, there is a need for automatic extraction of information to support biomedical researchers. Due to incomplete biomedical information databases, the extraction is not straightforward using dictionaries, and several approaches using contextual rules and machine learning have previously been proposed. Our work is inspired by the previous approaches, but is novel in the sense that it is using Google for semantic annotation of the biomedical words. The semantic annotation accuracy obtained - 52% on words not found in the Brown Corpus, Swiss-Prot or LocusLink (accessed using Gsearch.org) - is justifying further work in this direction.

Keywords: Biomedical Literature Data Mining, Semantic Annotation

1 Introduction

With the increasing importance of accurate and up-to-date databases for biomedical research, there is a need to extract information from biomedical research literature, e.g. those indexed in MEDLINE [34,33,15]. Examples of information databases are LocusLink, UniGene and Swiss-Prot [24,23,3].

Due to the rapidly growing amounts of biomedical literature, the information extraction process needs to be (mainly) automated. So far, the extraction approaches have provided promising results, but they are not sufficiently accurate and scalable.

Methodologically all the suggested approaches belong to the *information extraction field* [8], and in the biomedical domain they range from simple automatic methods to more sophisticated, but manual, methods. Good examples are: Learning relationships between proteins/genes based on co-occurrences in MEDLINE abstracts (e.g. [16]), *manually* developed information extraction rules

Examples of biological name entities in a textual context

1. "duodenum, a **peptone** meal in the"
2. "subtilisin plus leucine **amino-peptidase** plus prolidase followed"
3. "predictable hydrolysis of **[3H]digoxin-12alpha** occured in vitro"

(e.g. [35]), information extraction (e.g. protein names) classifiers trained on *manually* annotated training corpora (e.g. [4]), and our previous work on classifiers trained on *automatically* annotated training corpora [32]).

Semantic Annotation

An important part of information extraction is to *know* what the information is, e.g. knowing that the term "gastrin" is a protein or that "Tylenol" is a medication. Obtaining and adding this knowledge to given terms and phrases is called *semantic tagging* or *semantic annotation.*

1.1 Research Hypothesis

Our hypothesis is based on ideas from our preliminary experiments using *Google* to generate features for protein name extraction classifiers in [**?**], i.e. using the number of search hits for a word as a feature.

Fig. 1. Google is among the biggest known "information haystacks"

– Google is probably the world's largest available source of heterogeneous electronically represented information. *Can it be used for semantic tagging of textual entities in biomedical literature?* And if so, how?

The rest of this paper is organized as follows. Section 2 describes the materials used, section 3 presents our method, section 4 presents empirical results, section 5 describes related work, section 6 discusses our approach, and finally the conclusion and future work.

2 Materials

The materials used included biomedical (sample of MEDLINE abstract) and general English (Brown) textual corpora, as well as protein databases. See below for a detailed overview.

MEDLINE Abstracts - Gastrin-selection

The US National Institutes of Health (NIH) grants a free academic licence for PubMed/MEDLINE. It includes a local copy of 6.7 million abstracts, out of the 12.6 million entries that are available on their web interface. As subject for the expert validation experiments we used the collection of 12.238 gastrin-related MEDLINE abstracts that were available in September 2004.

Biomedical Information Databases

As a source for finding already known protein names we used a web search system called Gsearch, developed at Department of Cancer Research and Molecular Medicine at NTNU. It integrates common online protein databases, e.g. Swiss-Prot, LocusLink and UniGene, [24,23,3].

The Brown Corpus

The Brown repository (corpus) is an excellent resource for training a Part Of Speech (POS) tagger. It consists of 1,014,312 words of running text of edited English prose printed in the United States during the calendar year 1961. All the tokens are manually tagged using an extended Brown Corpus Tagset, containing 135 tags (Lancester-OsloBergen-tagset). The Brown corpus is included in the Python NLTK data-package, found at Sourceforge.

3 Our Approach

We have taken a modular approach where every submodule can easily be replaced by other similar modules in order to improve the general performance of the system. There are five modules connected to the data gathering phase, namely data selection, tokenization, POS-tagging, Stemming and Gsearch. Then the sixth and last module does a Google search for each extracted term. See figure 2.

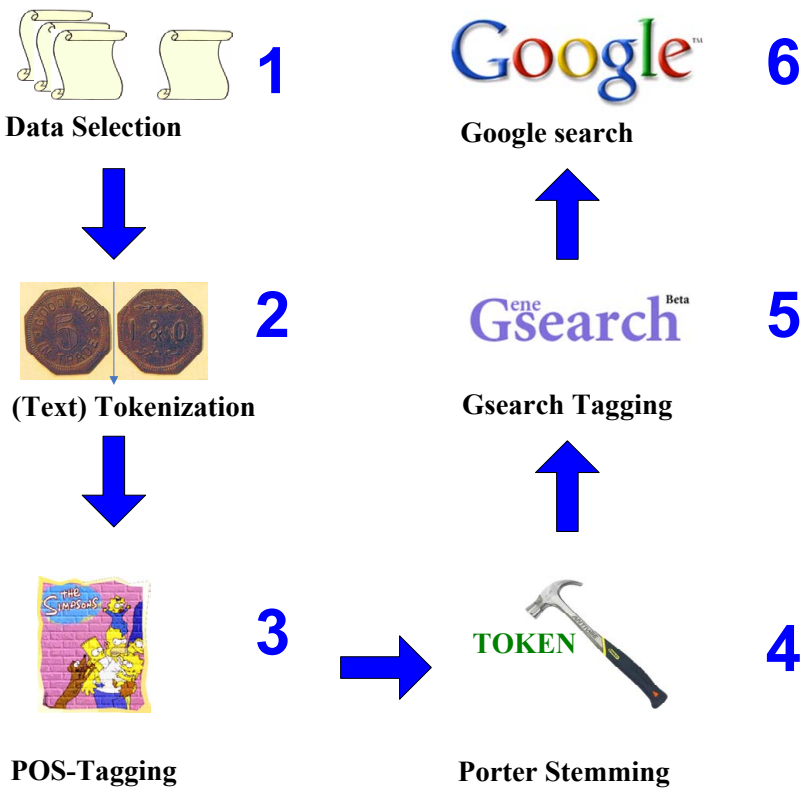

Fig. 2. Overview of Our Approach (named Alchymoogle)

1. **Data Selection** The data selection module uses PubMed Entrez online system to return a set of PubMed IDs (PMIDs) for a given protein, in our case "gastrin" (symbol GAS). The PMIDs are matched against our local copy of MEDLINE, to extract the specific abstracts.
2. **Tokenization** The text is tokenized to split it into meaningful tokens, or "words". We use the WhiteSpaceTokenizer from NLTK with some extra processing to adapt to the Brown Corpus, where every special character (like ( ) " ' - , and .) is treated as a seperate token. Words in parentheses are clustered together and tagged as a single token with the special tag *Paren*.
3. **POS tagging** Next, the text is tagged with Part-of-Speech (POS) tags using a Brill tagger trained on the Brown Corpus. This module acts as an advanced stop-word-list, excluding all the everyday common American English words from our protein search. Later, the actually given POS tags are used also as context features for the neighboring words.
4. **Porter-Stemming** We use the Porter Stemming Algorithm (also from NLTK) to remove even more everyday words from the "possibly biological term" can-

didate list. If the stem of a word can be tagged by the Brill tagger, then the word itself is given the special tag "STEM", and thereby transferred to the common word list.

5. **Gsearch** Identifies and removes already known entities from the search, but after the lookup in Gsearch, there are still some unknown words that are not yet stored in our dictionaries or databases, so in order to do any reasoning about these words it is important to know which class they belong to. Therefore, in the next phase they are subjected to some advanced Google-searching, in order to determine this.
6. **Google Class Selection** We have a network of 275 nouns, arranged in a semantic network on the form "X is a kind of Y". These nouns represent the classes that we want to annotate each word with. The input to this phase is a list of hitherto unknown words. From each Word a query on the form in the example below is formed (query syntax: *Word is (an|a)"*)
   Then these queries are fed to the PyGoogle module which allows 1000 queries to be run against the Google search engine every day with a personal password key. In order to maximize the use of this quota, the results of every query are cached locally, so that each given query will be executed only once. If a solution to the classification problem is not present among the first 10 results returned, the resultset can be expanded by 10 at a time, at the cost of one of the thousand quota-queries every time.
   Each returned hit from Google contains a "snippet" with the given query phrase and approximately 10 words on each side of it. We use some simple regular grammars to match the phrase and the words following it. If the next word is a noun it is returned. Otherwise, adjectives are skipped until a noun is encountered, or a "miss" is returned.

4 Empirical results

The table below shows the calculated classification scores for the expert evaluation phase. The first column shows *correct* predictions (True Positives and Negatives), the second column shows *incorrect* predictions (False Positives and Negatives), the third column gives Precision and Recall, the fourth gives the standard (balanced) F-Score number, and the last column presents the overall classification accuracy (correct classifications vs. incorrect ones).

Table 1. Semantic classification of *untagged* words

Classifier	TP/TN	FP/FN	Prec/Rec	F-score	CA
Alchymoogle	24/80	31/65	43.6/27.0	33.3	52.0

5 Related Work

Our specific approach was on using Google for *direct* semantic annotation (searching for is-a relations) of tokens (words) in biomedical corpora. We haven't been able to find other work that does this, but Dingare et al. is on using the number of Google hits as input features for a maximum entropy classifier used to detect protein and gene names [10,11]. Our work differs since we use Google to *directly determine* the semantic class of a word (searching for is-a relationships and parsing text (filtering adjectives) after *(a/an)* in *"Word is (a|an)*, as opposed to Dingare et al.'s *indirect* use of Google search as a feature for the information extraction classifier. A second difference between the approaches is that we search for explicit semantic annotation (e.g. "word is a protein") as opposed to their search for hints (e.g. "word protein"). The third important difference is that our approach does *automatic* annotation of corpuses, whereas they require pre-tagged (manually created) corpuses in their approach.

Other related works include extracting protein names from biomedical literature and some on semantic tagging using the web. Under, a brief overview of related work is given.

Work describing approaches for semantic annotation using the Web can be found in [27,12,18,19,9,22].

Semantic Annotation of Biomedical Literature

Other approaches for (semantic) annotation (mainly for protein and gene names) of biomedical literature include:

- Rule-based discovery of names (e.g. of proteins and genes), [13,29,36,35]
- Methods for discovering relationships of proteins and genes, [2,16].
- Classifier approaches (machine learning) with textual context as features, [4,5,6,14,1,20,30,21,17]
- Other approaches include generating probabilistic rules for detecting variants of biomedical terms, [31]

A comprehensive overview of such methods is provided in [28].

The paper by Cimiano and Staab [7] shows that a system (PANKOW) similar to ours works, and can be taken as a proof that automatic extraction using Google is a useful approach. Our systems differ in that we have 275 different semantic tags, while they only use 59 concepts in their ontology. They also have a table explaining how the number of concepts in a system influences the recall and precision in several other semantic annotation systems.

6 Discussion

In the following section we discuss our approach step-by-step. (The steps as presented in fig. 2.)

1. **Data selection** Since the results were inspected by cancer researchers the focus was naturally on proteins with a role in cancer development, and more specifically cancer in the stomach. One such protein is gastrin, and even though a search in the online PubMed Database returned more than eighteen thousand abstract IDs, only twelve thousand of these were found in our local academic copy of MEDLINE. Therefore only 12.238 abstracts were used as input to the tokenizer. Another important question is if the gastrin collection is representative for MEDLINE in general or for the "molecular biology" part of MEDLINE in particular.
2. **Tokenization into "words"** The tokenization algorithm is important in the sense that it dictates which "words" you have to deal with later in the pipeline. Our choice of using the Brown Corpus for training the Unigram and Brill taggers also influences our choice of tokenizing algorithm. For example, in the Brown Corpus all punctuation characters like comma, full stop, hyphen and so on are written with whitespace both before and after them. This turns them into separate tokens, disconnected from each other and from the other tokens. How to deal with parentheses is another question. Sometimes they are important parts of a protein name (often part of "formulae" describing the protein), and other times they are just used to state that the words within them aren't that important. We decided to keep the contents of parentheses as a single token, but this kind of parentheses clustering is a hard problem, especially if the parentheses aren't well balanced (like smiley and "1), 2), 3)" style paragraph numbering). Parentheses in MEDLINE are usually well balanced, but still some mistokenization was introduced at this point. Other tokens that require special attention are the multi-word-tokens. They can sometimes be composed using dash, bracket etc. as glue, but are at other times single words separated with whitespaces, even though they should really be one single token. One example is protein names, such as g-protein coupled receptor (GPCR).
3. **a) Brown Corpus and tagging** To train the Unigram and Brill taggers, an already tagged text is needed as a training set. We used the Brown Corpus, an American English corpus made from texts from 1961. They are rather old, and might not be as representative of "MEDLINE English" as we want. There is also the challenge of how quote symbols and apostrophes are used for protein names in MEDLINE abstracts, e.g. as a marker for the five-prime or three-prime end of a DNA formula. Also, there are only one million words in the corpus, so not all lowercase and capital letter combinations of every word are present.

   b) POS tagging with Brill algorithm and the Brown Corpus The Brill tagger doesn't tag perfectly, so maybe classifier-based taggers such as SVM could perform better. The performance of the Brill tagger could be better if we used a higher-ordered tagger than the unigram tagger as input to Brill, but the memory need for n-gram taggers are $O(m^n)$, where m is the number of words in the dictionary. So with million word training- and test sets, even the use of just a bi-gram tagger gets quite expensive in terms

of memory and time-use. Tagging itself may also introduce ambiguous tags (e.g. superman is a protein, but it may be tagged as a noun/name earlier in the pipeline, because that's the most common sense mentioned in the Brown Corpus).

4. **Porter-stemming** turns out to work poorly on protein and biological names, since they are often rooted in Latin or have acronyms as their name or symbol. E.g. the symbol for gastrin is GAS, and the porter stem of GAS becomes GA, which is wrong, and too ambiguous.
5. **Gsearch** The indexing algorithm of Gsearch also contains some stemming of the search terms, leading to some "strange" results when removing well-known proteins from the unknown words list. It should be extended with a larger selection of databases and dictionaries covering biological terms, so that protein names like "peptone" could also be found in the database. In other words there are "precision and recall" issues also at this stage, but our program should be able to solve "half of this problem" automatically. The worst problem is actually how to handle names with "strange characters" like ([]) in them, since these characters are usually not taken into account during the index-building in systems like Gsearch (or Google).
6. **Google Search** The precision of (positive) classification and the total classification accuracy is close to 50%, which is really good considering that no context information has been used in the classification process. By using context information in the way that is done in [**?**] it should be possible to increase the classification accuracy further. We had a lower recall than expected ($24/89 = 27.0\%$), mainly because a lot of our unknown words are parts of a multi-word-tokens, and can only be sensibly classified using the context which contains the rest of the multi-word-unit. Also, many of the words are not nouns, so they are not suitable class names in the first place, but still expert biologists often think of them in a concrete way. One example of this is "extracardiac", which were tagged as a place (outside the heart), even though nobody would actually write "extracardiac is a place outside the heart". (Except, I just did! And that really illustrates the problem of freedom, when dealing with Natural Language Understanding.)
   We did another test using 1500 semantic classes, instead of the 275 strictly molecular biology related classes. Then we got more hits among the 200 words, so this may be a method to increase the coverage of our system. It is of course much harder to manually evaluate these results, and there is also the danger of lowering the precision this way.

Acknowledgements

We would like to thank Waclaw Kusnierczyk for proposing additional biomedical information databases for inclusion in future work, and Tore Amble for continuous support. We would also like to thank Martin Thorsen Ranang for proposing improvements for future work. And finally a thanks to the Gsearch developers Jo Kristian Bergum, Hallgeir Bergum and Frode Jünge.

7 Conclusion and Future Work

This paper presents a novel approach - Alchymoogle - using Google for semantic annotation of entities (words) in biomedical literature.

We got empirically promising results - 52% semantic annotation accuracy ((TP+TN)/N, TP=24,TN=80,N=200) in the answers provided by Alchymoogle compared to expert classification performed by a molecular biologist. This encourages further work possibly in combination with other approaches (e.g. rule- and classification based information extraction methods), in order to improve the overall accuracy (both with respect to precision and recall). Disambiguation is another issue that needs to be further investigated. Other opportunities for future work include:

- Improve tokenization. Just splitting on whitespace and punctuation characters is *not* good enough. In biomedical texts non-alphabetic characters such as brackets and dashes need to be handled better.
- Improve stemming. The Porter algorithm for English language gives mediocre results on biomedical terms (e.g. protein names).
- Do spell-checking before a query is sent to Google, e.g. allowing minor variations of words (using the Levenshtein Distance).
- Search for other semantic tags using Google, e.g. "is a kind of" and "resembles", as well as negations ("is not a").
- Investigate whether the Google ranking is correlated with the accuracy of the proposed semantic tag. Are highly ranked pages better sources than lower ranked ones?
- Test our approach on larger datasets, e.g. *all* available MEDLINE abstracts.
- Combine this approach with more advanced natural language parsing techniques in order to improve the accuracy, [25,26].
- In order to find multiword tokens, one could extend the search query *(" X is (an|a) ")* to also include neighboring words of X, and then see how this affects the number of hits returned by Google. If there is no reduction in the number of hits, this means that the words are "always" printed together and are likely constituents in a multiword token. If you have only one actual hit to begin with, the certainty of the previous statement is of course very weak, but with increasing number of hits, the confidence is also growing.

References

1. Steffen Bickel, Ulf Brefeld, Lukas Faulstich, Jrg Hakenberg, Ulf Leser, Condrad Plake, , and Tobias Scheffer. A Support Vector Machine classifier for gene name recognition. In *Proceedings of the EMBO Workshop: A Critical Assessment of Text Mining Methods in Molecular Biology*, March 2004.
2. C. Blaschke, MA. Andrade, C. Ouzounis, and A. Valencia. Automatic Extraction of biological information from scientific text: Protein-protein interactions. In *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, pages 60–67. AAAI, 1999.

3. B. Boeckmann, A. Bairoch, R. Apweiler, MC. Blatter, A. Estreicher, E. Gasteiger, MJ Martin, K Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 31(1):365–370, January 2003.
4. Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. Comparative Experiments on Learning Information Extractors for Proteins and their Interactions. *Journal Artificial Intelligence in Medicine: Special Issue on Summarization and Information Extraction from Medical Documents (Forthcoming)*, 2004.
5. Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Raymond J. Mooney, Yuk Wah Wong, Edward M. Marcotte, and Arun Kumar Ramani. Learning to Extract Proteins and their Interactions from Medline Abstracts. In *Proceedings of the ICML-2003 Workshop on Machine Learning in Bioinformatics*, pages 46–53, August 2003.
6. Razvan Bunescu, Ruifang Ge, Raymond J. Mooney, Edward Marcotte, and Arun Kumar Ramani. Extracting Gene and Protein Names from Biomedical Abstracts. Unpublished Technical Note, Machine Learning Research Group, University of Texas at Austin, USA, March 2002.
7. Philipp Cimiano and Steffen Staab. Learning by Googling. *SIGKDD Explorations Newsletter*, 6(2):24–34, December 2004.
8. J. Cowie and W. Lehnert. Information Extraction. *Communications of the ACM*, 39(1):80–91, January 1996.
9. Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. SemTag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the Twelfth International World Wide Web Conference, WWW2003*, pages 178–186. ACM, 2003.
10. Shipra Dingare, Jenny Finkel, Christopher Manning, Malvina Nissim, and Beatrice Alex. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text. In *Proceedings of the BioCreative Workshop*, March 2004.
11. Shipra Dingare, Jenny Finkel, Christopher Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text. Submitted to BMC Bioinformatics, 2004.
12. Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. Submitted to Artificial Intelligence, 2004.
13. K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi. Toward information extraction: identifying protein names from biological papers. In *Proceedings of Pacific Symposium on Biocomputing*, pages 707–718, 1998.
14. Filip Ginter, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. New Techniques for Disambiguation in Natural Language and Their Application to Biological Texts. *Journal of Machine Learning Research*, 5:605–621, June 2004.
15. Jun ichi Tsuji and Limsoon Wong. Natural Language Processing and Information Extraction in Biology. In *Proceedings of the Pacific Symposium on Biocomputing 2001*, pages 372–373, 2001.
16. Tor-Kristian Jenssen, Astrid Lægreid, Jan Komorowski, and Eivind Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28, May 2001.
17. Sittichai Jiampojamarn. Biological term extraction using classification methods. Presentation at Dalhousie Natural Language Processing Meeting, June 2004.

18. Vinay Kakade and Madhura Sharangpani. Improving the Precision of Web Search for Medical Domain using Automatic Query Expansion. Online, 2004.
19. Udo Kruschwitz. Automatically Acquired Domain Knowledge for ad hoc Search: Evaluation Results. In *Proceedings of the 2003 Intl. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE'03)*. IEEE, 2003.
20. Sougata Mukherjea, L. Venkata Subramaniam, Gaurav Chanda, Sriram Sankararaman, Ravi Kothari, Vishal Batra, Deo Bhardwaj, and Biplav Srivastava. Enhancing a biomedical information extraction system with dictionary mining and context disambiguation. *IBM Journal of Research and Development*, 48(5/6):693–701, September/November 2004.
21. M. Narayanaswamy, KE Ravikumar, and K Vijay-Shanker. A biological named entity recognizer. In *Proceedings of the Pacific Symposium on Biocomputing 2003*, pages 427–438, 2003.
22. David Parry. A fuzzy ontology for medical document retrieval. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32*, pages 121–126. ACM Press, 2004.
23. JU. Pontius, L. Wagner, and GD. Schuler. *The NCBI Handbook*, chapter UniGene: a unified view of the transcriptome. National Center for Biotechnology Information, 2003.
24. KD Pruitt and DR Maglott. RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Research*, 29(1):137–140, January 2001.
25. Rune Sætre. GeneTUC, A Biolinguistic Project. (Master Project) Norwegian University of Science and Technology, Norway, June 2002.
26. Rune Sætre. Natural Language Processing of Gene Information. Master's thesis, Norwegian University of Science and Technology, Norway and CIS/LMU Munchen, Germany, April 2003.
27. Urvi Shah, Tim Finin, and Anupam Joshi. Information Retrieval on the Semantic Web. In *Proceedings of CIKM 2002*, pages 461–468. ACM Press, 2002.
28. Hagit Shatkay and Ronen Feldman. Mining the Biomedical Literature in the Genomic Era: An Overview. *Journal of Computational Biology*, 10(6):821–855, 2003.
29. Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.
30. Manabu Torii and K. Vijay-Shanker. Using Unlabeled MEDLINE Abstracts for Biological Named Entity Classification. In *Proceedings of the 13th Conference on Genome Informatics*, pages 567–568, 2002.
31. Yoshimasa Tsuruoka and Jun'ichi Tsuji. Probabilistic Term Variant Generator for Biomedical Terms. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 167–173. ACM, July/August 2003.
32. Amund Tveit, Rune Sætre, Tonje S. Steigedal, and Astrid Lægreid. ProtChew: Automatic Extraction of Protein Names from . In *Proceedings of the International Workshop on Biomedical Data Engineering (BMDE 2005, in conjunction with ICDE 2005)*, Tokyo, Japan, April 2005. IEEE Press (Forthcoming).
33. Limsoon Wong. A Protein Interaction Extraction System. In *Proceedings of the Pacific Symposium on Biocomputing 2001*, pages 520–530, 2001.
34. Limsoon Wong. Gaps in Text-based Knowledge Discovery for Biology. *Drug Discovery Today*, 7(17):897–898, September 2002.
35. Hong Yu, Vasileios Hatzivassiloglou, Carol Friedman, Andrey Rzhetsky, and W. John Wilbur. Automatic Extraction of Gene and Protein Synonyms from

MEDLINE and Journal Articles. In *Proceedings of the AMIA Symposium 2002*, pages 919–923, 2002.

36. Hong Yu, Vasileios Hatzivassiloglou, Andrey Rzhetsky, and W. John Wilbur. Automatically identifying gene/protein terms in MEDLINE abstracts. *Journal of Biomedical Informatics*, 35(5/6):322–330, October 2002.