gProt: Annotating Protein Interactions using Google and Gene Ontology

Rune Sætre[1], Amund Tveit[1,3], Martin T. Ranang[1], Tonje S. Steigedal[2], Liv Thommesen[2], Kamilla Stunes[2] and Astrid Lægreid[2]

[1] Department of Computer and Information Science,
[2] Department of Cancer Research and Molecular Medicine,
[3] Norwegian Centre for Patient Record Research
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway
[4] {rune.saetre,amund.tveit,martin.ranang}@idi.ntnu.no
{tonje.strommen,liv.thommesen,kamilla.stunes,astrid.laegreid}@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 cannot be done straightforward using dictionaries, so 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 combines Google and Gene Ontology for annotating protein interactions. We got promising empirical results - 57.5% terms as valid GO annotations, and 16.9% protein names in the answers provided by our system gProt. The total error-rate was 25.6% consisting mainly of overly general answers and syntactic errors, but also including semantic errors, other biological entities (than proteins and GO-terms) and false information sources.

Keywords: Biomedical Literature Data Mining, Gene Ontology, Google API

1 Introduction

With the increasing importance of accurate and up-to-date databases about proteins and genes for research, there is a need for efficient ways of updating these databases by extracting information from biomedical research literature [21, 20, 8], e.g. those indexed in MEDLINE. Examples of information resources containing such information are LocusLink, UniGene and Swiss-Prot for protein info and the Gene Ontology for semantic labels.

Due to the large and rapidly growing amounts of biomedical literature, the extraction process needs to be more *automatic* than previously. Current 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* [3], and in the biomedical domain they range

from simple automatic methods to more sophisticated, but slightly more manual, methods. Good examples are: Learning relationships between proteins/genes based on co-occurrences in MEDLINE abstracts (e.g. [9]), *manually* developed information extraction rules (e.g. [22]), information extraction (e.g. protein names) classifiers trained on *manually* annotated training corpora (e.g. [1]), and classifiers trained on *automatically* annotated training corpora [19]).

1.1 Research Hypothesis

Internet Search Engines such as Google, Yahoo and MSN Search are the world's largest readily available information sources, also in the biomedical domain. Based on promising results from recent work on using Google for semantic annotation of biomedical literature [16], we are encouraged to investigate if Google can be used to find protein interactions that match the Gene Ontology (GO). This leads to the hypothesis:

Can Internet Search engines such as Google be used to annotate protein interactions in the Gene Ontology framework?

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 last the conclusion and future work.

2 Materials

See fig. 1 for an overview of the system. As input for our experiments we used the following:

- 10 proteins that are already well-known to our biology experts.
- 37 verb-templates suggested by Martin et. al (LexiQuest) [12].

Proteins

The following proteins were used as input to the system.

Proteins used
'EGF', 'TNF', 'CCK', 'gastrin', 'CCKAR', 'CCKBR', 'p53', 'ATF1', 'CREB', 'CREM'.

In addition, each protein is also described by several other names or synonyms in the literature. E.g. gastrin is also known as 'g14', 'g17', 'g34', 'GAS', 'gast', 'gastrin precursor ', 'gastrin 14', etc. So our biologists compiled a list of roughly 10 synonyms for each protein, *giving us about 100 terms total to annotate.*

Interaction Verbs

We selected our interaction verb templates from table 1 in [12]. They had a list of 44 verbs, but we chose to use only 37 of these verbs. The reason for this is that we are focusing on simple statements like "gastrin activates ...", with the object of the verb following directly after the verb template. The following table shows the original list of verbs, with the removed ones in parenthesis.

Verb templates used
acetylates, activates, (antagonizes), associates with, (attenuates), (binding to), binds, blocks, (bonds), (complex), deactivates, decreases, degrades, dephosphorylates, dimerizes, dissociates from, downregulates, forms complex with, hydrolyses, inactivates, increases, induces, inhibits, interacts with, links, mediates, (oligomerizes), overexpresses, phosphorylates, potentiates, precipitates with, reacts with, recruits, (reduces), regulates, releases, represses, stimulates, transactivates, transduces, transforms, triggers, ubiquitinates, upregulates,

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 in the system. The first one sets up the search queries, the second runs the queries against Google, the third one tokenizes the results, the fourth parses the tokenized text, and the fifth and last module extracts all the results and presents them to the human evaluators. See figure 1.

1. **Data Selection** N (=100) protein names are combined with M (=37) verb templates, giving a total of N x M (3700) queries to run against Google.
2. **Google** The 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 search returns more than ten results, the resultset can be expanded by ten at a time, at the cost of one of the 1000 quota-queries every time. We decided to use up to 30 results for each query in this experiment.
3. **Tokenization** The text is tokenized to split it into meaningful tokens, or "words". We use a simple WhiteSpaceTokenizer from NLTK, where every special character (like ( ) " ' - , and .) is treated as a separate token.
4. **Parsing** Each returned hit from Google contains a "snippet" with the given query phrase and approximately ten words on each side of it. We use some simple regular grammars to match the phrase and the words following it.

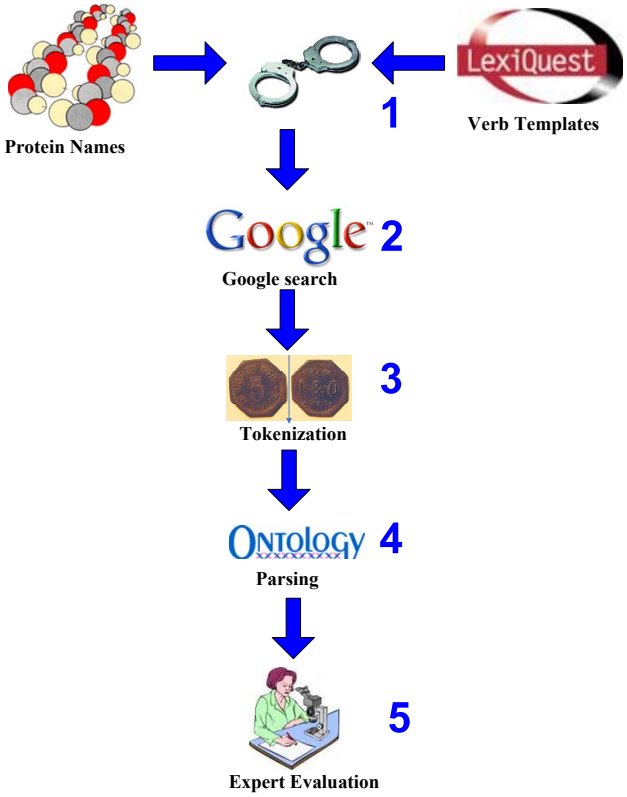

Fig. 1. Overview of Our Approach (named gProt)

If the next word is a noun it is returned. Otherwise, adjectives are skipped until a noun is encountered, or a "miss" is returned.

5. **Expert Evaluation** The results were merged so that all synonyms were treated as if the main protein name had been used in the original query. Then the results were put into groups (one group for each protein-verb pair) and sorted alphabetically within that group. These results were then presented to the biologists, who evaluated the usefulness of our results from Google.

4 Empirical results

Fig. 2 and 3 show the results. The first one shows that more than half of the extracted terms were terms that could be used to annotate the given protein according to the Gene Ontology (GO). Around one fifth of the results contained an identifiable protein name that could be stored as a protein-protein interaction. Only one quarter of the terms were deemed not useful. The different kinds of "not useful"-errors can be read out of fig. 3.

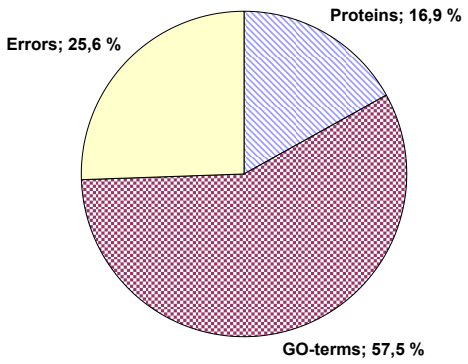

Fig. 2. Main Results

5 Related Work

Our specific approach was on using Google and Gene Ontology for annotating protein interactions. We haven't been able to find other work that does this, but the closest are Dingare et al., that uses results from Google search as a feature for a maximum entropy classifier used to detect protein and gene names [5, 6], and our previous work on semantic annotation of proteins (i.e. tagging of individual proteins, not their GO relation) [16]. Google has also been used for semantic tagging outside of the biomedical field, e.g. in Cimiano and Staab's PANKOW system [2] and in [17, 7, 10, 11, 4, 13].

A comprehensive overview of past methods for protein-related information extraction is provided in [18].

6 Discussion

In the following section we discuss our approach step-by-step. (The steps as presented in fig. 1.)

1. **Data Selection** The results were inspected by cancer researchers, so the focus was naturally on proteins with a role in cancer development, and more specifically cancer in the stomach. One such protein is gastrin, used as a running example in this article. In the experiment we used ten such protein names with around ten synonyms for each. The large number of synonyms used for each original protein name gave us a valuable increase in the recall of expected facts from Google.
2. **Google** Since we decided to download up to 3 (times 10) results for each query, we had to do around 11.000 queries. This took almost two weeks, because of Google's restraint to only run 1000 queries per day. If we want to

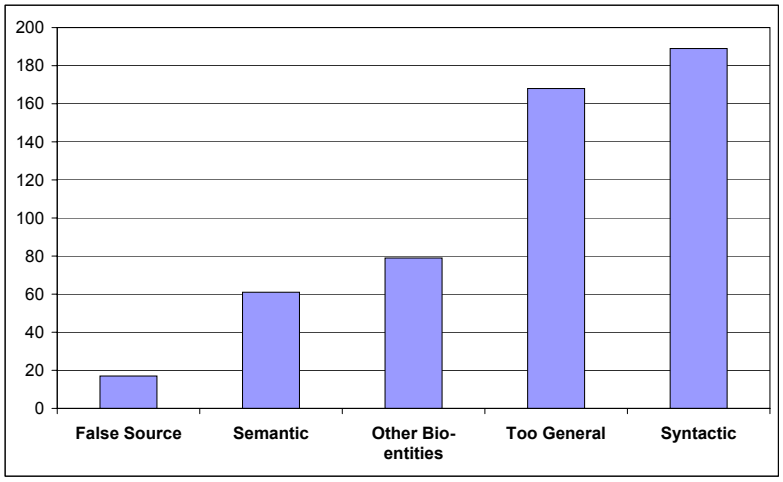

Fig. 3. Breakdown of Errors

scale up this method in the future, we would probably have to pay Google to let us do more queries per day, or consider using the recently announced Yahoo API that allows 5.000 queries per day. The number of returned GO-processes was over 50%, which is very promising for automatic annotation, considering that no information has been used in the process to match GO-terms more often than e.g. protein names.

3. **Tokenization** Most of the "errors" are syntactic errors, and many of the syntactic errors occur because of bad tokenization, mainly because a lot of the returned words are just parts of multi-word-tokens. Also, many of the words are not nouns at all, so they are not suitable class names in the first place. In the future more work should be done in the tokenization phase. The WhiteSpaceTokenizer was used because it is easy and fast, but with some sort of NP-clustering and parentheses handling, almost half of the errors could be removed. One example of NP clustering is protein names, such as "g-protein coupled receptor (GPCR)".
   How to deal with parentheses? 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. And the worst problem is that they are quite often "unbalanced", either because of typing errors, "1) 2) 3)"-style numbering, or smileys.
4. **Parsing** We used a really simple grammar to extract the interacting terms from what Google returned. It can be summed up as: After the template, keep reading words until a "stop-word" is encountered. As "stop-words" we used some common prepositions, in addition to full-stop punctuation (.,;?!). There is obviously room for a lot of improvements here, e.g. using more advanced Natural Language Understanding techniques.

5. **Expert Evaluation** The evaluation was quite simple, just focusing on deciding whether this way of using Google to do information extraction is worth pursuing or not. Since the tokenization and grammar modules aren't perfect yet, the biologist also had access to the complete snippets (and the corresponding homepage) in their evaluation work. It is now obvious to us that we should keep developing this system, since almost three out of four results were relevant, and many of them also novel, information.

7 Conclusion and Future Work

This paper presents a novel approach - gProt - using Google to find semantic (GO-) annotations for specific proteins.

We got empirically promising results - 57.5% semantic annotation classes, and 16.9% protein names in the answers provided by gProt. This means that 74.4% of the results are useful. This encourages further work, possibly in combination with other approaches (e.g. rule based information extraction methods), in order to improve the overall accuracy. In the similar task of protein name identification, recently presented precision scores ranges from 70 to 75% [1]. Hopefully, more advanced methods will greatly reduce the number of errors (useless information), which is currently at 25.6%. Disambiguation is another issue that needs to be further investigated, because sometimes different search-results are really just one single identity, because of synonyms and acronyms for example. 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.
- Search for other verb templates using Google. E.g. Which templates give the best results, and what about negations ("does not activate ...")?
- 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. using *all* the returned results from Google.
- Combine this approach with more advanced natural language parsing techniques in order to improve the accuracy, [14, 15].
- In order to find multiword tokens, one could extend the search query *(" X activates ")* 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.
- In this experiment very crude Part Of Speech (POS) tagging is done, so our results can be seen as a baseline for this kind of experiment. In the future

we want to improve the results, for example by utilizing better grammars, and more advanced natural language understanding techniques.

Acknowledgements

We would like to thank Waclaw Kusnierczyk and Tore Amble for continuous support.

References

1. 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.
2. Philipp Cimiano and Steffen Staab. Learning by Googling. *SIGKDD Explorations Newsletter*, 6(2):24–34, December 2004.
3. J. Cowie and W. Lehnert. Information Extraction. *Communications of the ACM*, 39(1):80–91, January 1996.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. Vinay Kakade and Madhura Sharangpani. Improving the Precision of Web Search for Medical Domain using Automatic Query Expansion. Online, 2004.
11. 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.
12. Eric P. G. Martin, Eric G. Bremer, Marie-Claude Guerin, Catherine DeSesa, and Olivier Jouve. Analysis of Protein/Protein Interactions Through Biomedical Literature: Text Mining of Abstracts vs. Text Mining of Full Text Articles. In *Proceedings of the Knowledge Exploration in Life Science Informatics (KELSI2004)*

Symposium, volume 3303 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 96–108. Springer-Verlag Heidelberg, 2004.
13. 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.
14. Rune Sætre. GeneTUC, A Biolinguistic Project. (Master Project) Norwegian University of Science and Technology, Norway, June 2002.
15. 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.
16. Rune Sætre, Amund Tveit, Tonje Strœmmen Steigedal, and Astrid Lægreid. Semantic Annotation of Biomedical Literature using Google. In Dr. Marina Gavrilova, Dr. Youngsong Mun, Dr. David Taniar, Dr. Osvaldo Gervasi, Dr. Kenneth Tan, and Dr. Vipin Kumar, editors, *Proceedings of the International Workshop on Data Mining and Bioinformatics (DMBIO2005)*, Lecture Notes in Computer Science (LNCS) (Forthcoming), Singapore, May 2005. Springer-Verlag Heidelberg.
17. Urvi Shah, Tim Finin, and Anupam Joshi. Information Retrieval on the Semantic Web. In *Proceedings of CIKM 2002*, pages 461–468. ACM Press, 2002.
18. Hagit Shatkay and Ronen Feldman. Mining the Biomedical Literature in the Genomic Era: An Overview. *Journal of Computational Biology*, 10(6):821–855, 2003.
19. Amund Tveit, Rune Sætre, Tonje S. Steigedal, and Astrid Lægreid. ProtChew: Automatic Extraction of Protein Names from Biomedical Literature. In *Proceedings of the International Workshop on Biomedical Data Engineering (BMDE 2005, in conjunction with ICDE 2005)*, Tokyo, Japan, April 2005. IEEE Press (Forthcoming).
20. Limsoon Wong. A Protein Interaction Extraction System. In *Proceedings of the Pacific Symposium on Biocomputing 2001*, pages 520–530, 2001.
21. Limsoon Wong. Gaps in Text-based Knowledge Discovery for Biology. *Drug Discovery Today*, 7(17):897–898, September 2002.
22. 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.