ACL-05/ISMB-05

Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics

Proceedings of the Workshop

24 June 2005
Detroit, Michigan, USA

Production and Manufacturing by
Omnipress Inc.
Post Office Box 7214
Madison, WI 53707-7214

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
75 Paterson Street, Suite 9
New Brunswick, NJ 08901
USA
Tel: +1-732-342-9100
Fax: +1-732-342-9339
acl@aclweb.org

Introduction

This volume contains the full papers accepted for presentation at the BioLINK 2005 meeting. This workshop represents the first joint Association for Computational Linguistics (ACL)/Intelligent Systems for Molecular Biology (ISMB) meeting. Each organization has held a workshop in this area for the past three to four years; this is the first meeting sponsored jointly by the two parent organizations. In bringing these two groups together, we have also melded two different traditions of distribution. The ISMB tradition has been focussed on invited talks and "short papers" describing works in progress. The ACL tradition has focussed on rigorously peer-reviewed full papers describing compeleted work. This workshop features works in the three categories of "full paper," "short paper," and poster submissions. Submissions in all three categories underwent an ACL-style peer review process.

Recent years have seen an interesting confluence between the worlds of bioinformatics and natural language processing. Molecular biologists, confronted with new high-throughput sources of data, have recognized that language processing can provide them with tools for handling a flood of data that is unprecedented in the history of the life sciences. The natural language processing community, in turn, has become aware of the resources that the computational bioscience community has made available, and there has been growing interest in applying natural language processing techniques to mine the biological literature to support complex applications in the biological domain, ranging from identifying relevant literature, to extraction of experimental findings for the population of biological knowledge bases, to summarization—all in order to present key facts to biologists in succinct form.

This workshop continued the interaction between these communities. We received a total of eighteen full-paper submissions, from which eight were selected for presentation at the workshop and inclusion in the ACL BioLINK workshop proceedings. An additional two of the full-paper submissions were accepted as posters. Overall, eight of the full-paper submissions were concerned with entity identification. Five of the eighteen dealt with information extraction. In addition, we received submissions on the important topic of normalizing entity mentions.

BioLINK also solicited short-paper and poster submissions. Twenty-one short-paper submissions were received, five of which were accepted for oral presentation. Four more were accepted for poster presentation. All nine of these short papers are being distributed by ISMB as part of its SIG materials. The meeting also featured a poster session.

K. Bretonnel Cohen
Lynette Hirschman
Hagit Shatkay
Christian Blaschke

Organizers:

K. Bretonnel Cohen, University of Colorado School of Medicine
Lynette Hirschman, MITRE
Hagit Shatkay, Queen's University
Christian Blaschke, *bioalma*

Program Committee:

Sophia Ananiadou, University of Salford
Lan Aronson, NLM
Breck Baldwin, Alias-i Inc.
Olivier Bodenreider, NLM
Shannon Bradshaw, University of Iowa
Bob Carpenter, Alias-i Inc.
Jeff Chang, Duke Univeristy
Aaron Cohen, Oregon Health Sciences University
Nigel Collier, National Institute of Informatics
Lynne Fox, University of Colorado Health Sciences Center
Bob Futrelle, Northeastern University
Henk Harkema, University of Sheffield
Marti Hearst, University of California at Berkeley
Larry Hunter, University of Colorado School of Medicine
Steve Johnson, Columbia University
Marc Light, University of Iowa
Hongfang Liu, University of Maryland at Baltimore County
Alex Morgan, MITRE
James Pustejovsky, Brandeis University
Thomas Rindflesch, NLM
Andrey Rzhetsky, Columbia University
Jasmin Saric, EML Research gGmbH
Lorrie Tanabe, NCBI, NLM
Jun-ichi Tsujii, University of Tokyo
Alfonso Valencia, Universidad Autonoma de Madrid
Karin Verspoor, Los Alamos National Labs
John Wilbur, NCBI, NLM
Hong Yu, Columbia University

Invited Speaker:

Judith A. Blake, Mouse Genome Informatics

Table of Contents

Conference Program

Friday, June 24, 2005

8:30–8:45 Opening Remarks

8:45–9:10 *Weakly supervised learning methods for improving the quality of gene name normalization data*
Ben Wellner

9:10–9:30 *Adaptive string similarity metrics for biomedical reference resolution*
Ben Wellner, José Castaño and James Pustejovsky

9:35–10:00 *Unsupervised gene/protein named entity normalization using automatically extracted dictionaries*
Aaron Cohen

10:00–10:30 Coffee Break

10:30–11:15 Invited Talk by Judi Blake

11:20–11:45 *A machine learning approach to acronym generation*
Yoshimasa Tsuruoka, Sophia Ananiadou and Jun'ichi Tsujii

12:00–13:00 Lunch

13:00–13:15 *Searching for high-utility text in the biomedical literature*
H. Shatkay, A. Rzhetsky and W.J. Wilbur

13:15–13:40 *MedTag: a collection of biomedical annotations*
Lawrence H. Smith, Lorraine Tanabe, Thomas Rindflesch and W. John Wilbur

13:45–14:10 *Corpus design for biomedical natural language processing*
K. Bretonnel Cohen, Lynne Fox, Philip V. Ogren and Lawrence Hunter

14:10–14:25 *A cross-domain application of natural language processing in biology*
I. Chiu and L.H. Shu

14:30–15:00 Coffee Break

15:00–15:15 *Functional annotation of genes using hierarchical text categorization*
S. Kiritchenko, S. Matwin and A.F. Famili

Friday, June 24, 2005 (continued)

15:15–15:30 *Automatic highlighting of bioscience literature*
H. Wang, S. Bradshaw and M. Light

15:30–15:55 *Using biomedical literature mining to consolidate the set of known human protein-protein interactions*
Arun Ramani, Razvan Bunescu, Raymond Mooney and Edward Marcotte

15:55–16:20 *IntEx: A syntactic role driven protein-protein interaction extractor for bio-medical text*
Syed Toufeeq Ahmed, Deepthi Chidambaram, Hasan Davulcu and Chitta Baral

16:20–16:50 Concluding discussion

16:50–17:20 Poster Boasters

17:20–18:30 Poster Session

Weakly Supervised Learning Methods for Improving the Quality of Gene Name Normalization Data

Ben Wellner
wellner@mitre.org

The MITRE Corporation
202 Burlington Rd
Bedford MA 01730

Computer Science Department
Brandeis University
Waltham MA 02454

Abstract

A pervasive problem facing many biomedical text mining applications is that of correctly associating mentions of entities in the literature with corresponding concepts in a database or ontology. Attempts to build systems for automating this process have shown promise as demonstrated by the recent BioCreAtIvE Task 1B evaluation. A significant obstacle to improved performance for this task, however, is a lack of high quality training data. In this work, we explore methods for improving the quality of (noisy) Task 1B training data using variants of weakly supervised learning methods. We present positive results demonstrating that these methods result in an improvement in training data quality as measured by improved system performance over the same system using the originally labeled data.

1 Introduction

A primary set of tasks facing biomedical text processing systems is that of categorizing, identifying and classifying entities within the literature. A key step in this process involves grouping mentions of entities together into equivalence classes that denote some underlying entity. In the biomedical domain, however, we are fortunate to have structured data resources such as databases and ontologies with entries denoting these equivalence classes. In biomedical text mining, then, this process involves associating mentions of entities with known, *existing* unique identifiers for those entities in databases or ontologies – a process referred to as *normalization*. This ability is required for text processing systems to associate descriptions of concepts in free text with a grounded, organized system of knowledge more readily amenable to machine processing.

The recent BioCreAtIvE Task 1B evaluation challenged a number of systems to identify genes associated with abstracts for three different organisms: mouse, fly and yeast. The participants were provided with a large set of noisy training data and a smaller set of higher quality development test data. They were also provided with a lexicon containing all the potential gene identifiers that might occur and a list of known, though incomplete, names and synonyms that refer to each of them.

To prepare the training data, the list of unique gene identifiers associated with each full text article was obtained from the appropriate model organism database. However, the list had to be pruned to correspond to the genes mentioned in the abstract. This was done by searching the abstract for each gene on the list or its synonyms, using exact string matching. This process has the potential to miss genes that were referred to in the abstract using a phrase that does not appear in the synonym list. Additionally, the list may be incomplete, because not all genes mentioned in the article were curated, so there are mentions of genes in an abstract that did not have a corresponding identifier on the gene list.

This paper explores a series of methods for attempting to recover some of these missing gene

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 1–8, Detroit, June 2005.

identifiers from the Task 1B training data abstracts. We start with a robust, machine learning-based baseline system: a reimplementation of the system in [1]. Briefly, this system utilizes a classifier to select or filter matches made against the synonym list with a loose matching criterion. From this baseline, we explore various methods for re-labeling the noisy training data, resulting in improved scores on the overall Task 1B development test and evaluation data. Our methods are based on weakly supervised learning techniques such as co-training [2] and self-training [3, 4] for learning with both labeled and unlabeled data.

The setting here is different than the typical setting for weakly supervised learning, however, in that we have a large amount of *noisily* labeled data, as opposed to completely *un*labeled data. The main contribution of this work is a framework for applying weakly supervised methods to this problem of re-labeling noisy training data.

Our approach is based on partitioning the training data into two sets and viewing the problem as two mutually supporting weakly supervised learning problems. Experimental results demonstrate that these methods, carefully tuned, improve performance for the gene name normalization task over those previously reported using machine learning-based techniques.

2 Background and Related Work

2.1 Gene Name Normalization and Extraction

The task of normalizing and identifying biological entities, genes in particular, has received considerable attention in the biological text mining community. The recent Task 1B from BioCreAtIvE [5] challenged systems to identify unique gene identifiers associated with paper abstracts from the literature for three organisms: mouse, fly and yeast. Task 1A from the same workshop focused on identifying (i.e. tagging) mentions of genes in biomedical journal abstracts.

2.2 NLP with Noisy and Un-labeled Training Data

Within biomedical text processing, a number of approaches for both identification and normalization of entities have attempted to make use of the many available structured biological resources to "bootstrap" systems by deriving noisy training data for the task at hand. A novel method for using noisy (or "weakly labeled") training data from biological databases to learn to identify relations in biomedical texts is presented in [6]. Noisy training data was created in [7] to identify gene name mentions in text. Similarly, [8] employed essentially the same approach using the FlyBase database to identify normalized genes within articles.

2.3 Weakly Supervised Learning

Weakly supervised learning remains an active area of research in machine learning. Such methods are very appealing: they offer a way for a learning system provided with only a small amount of labeled training data and a large amount of un-labeled data to perform better than using the labeled data alone. In certain situations (see [2]) the improvement can be substantial.

Situations with small amounts of labeled data and large amounts of unlabeled data are very common in real-world applications where labeling large quantities of data is prohibitively expensive. Weakly supervised learning approaches can be broken down into *multi*-view and *single*-view methods.

Multi-view methods [2] incrementally label unlabeled data as follows. Two classifiers are trained on the training data with different "views" of the data. The different views are realized by splitting the set of features in such a way that the features for one classifier are conditionally independent of features for the other *given the class label*. Each classifier then selects the most confidently classified instances from the unlabeled data (or some random subset thereof) and adds them to the training set. The process is repeated until all data has been labeled or some other stopping criterion is met. The intuition behind the approach is that since the two classifiers have different views of the data, a new training instance that was classified with high confidence by one classifier (and thus is "redundant" from that classifier's point of view) will serve as an informative, novel, new training instance for the other classifier and vice-versa.

Single-view methods avoid the problem of finding an appropriate feature split which is not possible or appropriate in many domains. One common approach here [4] involves learning an ensemble of

classifiers using *bagging*. With bagging, the training data is randomly sampled, with replacement, with a separate classifier trained on each sample. Un-labeled instances are then labeled if *all* of the separate classifiers agree on the label for that instance. Other approaches are based on the expectation maximization algorithm (EM) [9].

3 System Description

The baseline version of our system is essentially a reproduction of the system described in [1] with a few modifications. The great appeal of this system is that, being machine learning based, it has no organism-specific aspects hard-coded in; moving to a new organism involves only re-training (assuming there is training data) and setting one or two parameters using a held-out data set or cross-validation.

The system is given a set of abstracts (and associated gene identifiers at training time) and a lexicon. The system first proposes candidate phrases based on all possible phrases up to 8 words in length with some constraints based on part-of-speech[1]. Matches against the lexicon are then carried out by performing exact matching but ignoring case and removing punctuation from the both the lexical entries and candidate mentions. Only *maximal* matching strings were used – i.e. sub-strings of matching strings that match the same id are removed.

The resulting set of matches of candidate mentions with their matched identifiers results in a set of *instances*. These instances are then provided with a label - "yes" or "no" depending on whether the match in the abstract is correct (i.e. if the gene identifier associated with the match was annotated with the abstract). These instances are used to train a binary maximum entropy classifier that ultimately decides if a match is valid or not.

Maximum entropy classifiers model the conditional probability of a class, **y**, (in our setting, **y**="yes" or **y**="no") given some observed data, **x**. The conditional probability has the following form in the binary case (where it is equivalent to logistic regression):

$$P(y \mid x) = \frac{\exp(\sum_i \lambda_i f_i(x, y))}{Z(x)}$$ where Z(x) is the normalization function, the λ_i are real-valued model parameters and the f_i are arbitrary real-valued feature functions.

One advantage of maximum entropy classifiers is the freedom to use large numbers of statistically non-independent features. We used a number of different feature types in the classifier:

- the matching phrase
- the matched gene identifier
- the previous and subsequent two words of the phrase
- the number of words in the matching phrase
- the total number of genes that matched against the phrase
- all character prefixes and suffixes up to length 4 for words within the phrase

An example is shown below in Figure 1 below.

Abstract Excerpt:

"This new receptor, **TOR** (thymus orphan receptor)…"

Feature Class	Specific Feature
Phrase	TOR
GENEID	MGI104856
Previous-1	,
Previous-2	receptor
Subsequent-1	(
Subsequent-2	thymus
Number of Matches	2
Number of Words	1
Prefix-1	T
Prefix-2	TO
Prefix-3	TOR
Suffix-1	R
Suffix-2	OR
Suffix-3	TOR

Figure 1. An abstract excerpt with the matching phrase "TOR". The resulting features for the match are detailed in the table.

In addition to these features we created additional features constituting conjunctions of some of these "atomic" features. For example, the conjoined feature **Phrase=TOR AND GENEID=MGI104856** is "on" when both conjuncts are true of the instance.

To assign identifiers to a new abstract a set features are extracted for each matching phrase and

[1] Specifically, we excluded phrases that began with verbs prepositions, adverbs or determiners; we found this constraint did not affect recall while reducing the number of candidate mentions by more than 50%.

gene id pair just as in training (this constitutes an *instance*) and presented to the classifier for classification. As the classifier returns a *probability* for each instance, the gene id associated with the instance with highest probability is returned as a gene id associated with the abstract, except in the case where the probability is less than some threshold $T, 0 \leq T \leq 1$ in which case no gene id is returned for that phrase.

Training the model involves finding the parameters that maximize the log-likelihood of the training data. As is standard with maximum entropy models we employ a Gaussian prior over the parameters which bias them towards zero to reduce overfitting.

Our model thus has just two parameters which need to be tuned to different datasets (i.e. different organisms): the Gaussian prior and the threshold, T. Tuning the parameters can be done on a held out set (we used the Task 1B development data) or by cross validation:

4 Weakly Supervised Methods for Re-labeling Noisy Normalization Data

The primary contribution of this work is a novel method for re-labeling the noisy training instances within the Task 1B training data sets. Recall that the Task 1B training data were constructed by matching phrases in the abstract against the synonym lists for the gene ids curated for the full text article for which the abstract was written. In many cases, mentions of the gene in the abstract do not appear exactly as they do in the synonym list, which would result in a missed association of that gene id with the abstract. In other cases, the database curators simply did not curate a gene id mentioned in the abstract as it was not relevant to their particular line of interest.

Our method for re-labeling potentially mislabeled instances draws upon existing methods for *weakly supervised learning*. We describe here the generic algorithm and include specific variations below in the experimental setup.

The first step is to partition the training data into two disjoint sets, D_1 and D_2.[2] We then create two instances of the weakly supervised learning problem where in one instance, D_1 is viewed as the labeled training data and D_2 is viewed as the unlabeled data, and in the other instance their roles are reversed. Re-labeling of instances in D_1 is carried out by a classifier or ensemble of classifiers, C_2 trained on D_2. Similarly, instances in D_2 are re-labeled by C_1 trained on D_1. Those instances for which the classifier assigns high confidence (i.e. for which $P(y = "yes" | x)$ is high) but for which the existing label disagrees with the classifier are candidates for re-labeling. Figure 2 diagrams this process below.

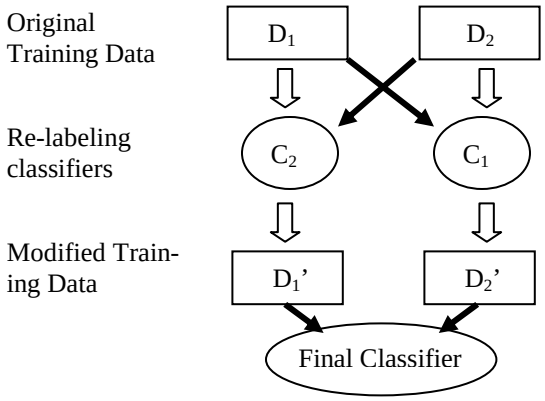


Figure 2. Diagram illustrating the method for re-labeling instances. The solid arrows indicate the training of a classifier from some set of data, while block arrows describe the data flow and re-labeling of instances.

One assumption behind this approach is that not all of the errors in the training data labels are correlated. As such, we would expect that for a particular mislabeled instance in D_1, there may be similar positive instances in D_2 that provide evidence for re-labeling the mislabeled in D_1.

Initial experiments using this approach met with failure or negligible gains in performance. We initially attributed this to too many correlated errors. Detailed error analysis revealed, however, that a significant portion of training instances being re-labeled were derived from matches against the lexicon that were not, in fact, references to genes – i.e. they were other more common English words that happened to appear in the synonym lists for which the classifier mistakenly assigned them high probability.

[2] Note that instances in D_1 and D_2 are also derived form disjoint sets of *abstracts*. This helps ensure that very similar instances are unlikely to appear in different partitions.

Our solution to this problem was to impose a constraint on instances to be re-labeled: The phrase in the abstract associated with the instance *is required to have been tagged as a gene name by a gene name tagger in addition to the instance receiving a high probability by the re-labeling classifier*. Use of a gene name tagger introduces a check against the classifier (trained on the *noisy* training data) and helps to reduce the chance of introducing false positives into the labeled data.

We trained our entity tagger, Carafe, on a the Genia corpus [10] together with the BioCreative Task 1A gene name training corpus. Not all of the entity types annotated in the Genia corpus are genes, however. Therefore we used an appropriate subset of the entity types found in the corpus. Carafe is based on Conditional Random Fields [11] (CRFs) which, for this task, employed a similar set of features to the CRF described in [12].

5 Experiments and Results

The main goal of our experiments was to demonstrate the benefits of re-labeling potentially noisy training instances in the task 1B training data. In this work we focus the weakly supervised re-labeling experiments on the *mouse* data set. In the mouse data there is a strong bias towards false negatives in the training data – i.e. many training instances have a negative label and should have a positive one. Our reasons for focusing on this data are twofold: 1) we believe this situation is likely to be more common in practice since an organism may have impoverished synonym lists or "gaps" in the curated databases and 2) the experiments and resulting analyses are made clearer by focusing on re-labeling instances in one direction only (i.e. from negative to positive).

In this section, we first describe an initial experiment comparing the baseline system (described above) using the original training data with a version trained with an augmented data set where labels changed based on a simple heuristic. We then describe our main body of experiments using various weakly supervised learning methods for re-labeling the data. Finally, we report our overall scores on the evaluation data for all three organisms using the best system configurations derived from the development test data.

5.1 Data and Methodology

We used the BioCreative Task 1B data for all our experiments. For the three data sets, there were 5000 abstracts of training data and 250, 110 and 108 abstracts of development test data for mouse, fly and yeast, respectively. The final evaluation data consisted of 250 abstracts for each organism. In the training data, the ratios of positive to negative *instances* are the following: for mouse: 40279/111967, for fly: 75677/493959 and for yeast: 25108/3856. The number of features in each trained model range from 322110 for mouse, 881398 for fly and 108948 for yeast.

Given a classifier able to rank all the test instances (in our case, the ranks derive from the probabilities output by the maximum entropy classifier), we return only the top n gene identifiers, where n is the number of correct identifiers in the development test data – this results in a balanced F-measure score. We use this metric for all experiments on the development test data as it allows better comparison between systems by factoring out the need to tune the threshold.

On the evaluation data, we do not know n. The system returns a number of identifiers based on the threshold, T. For these experiments, we set T on the development test data and choose three appropriate values for three different evaluation "submissions".

5.2 Experiment Set 1: Effect of match-based re-labeling

Our first set of experiments uses the baseline system described earlier. We compare the results of this system using the Task 1B training data "as provided" with the results obtained by re-labeling some of the negative instances provided to the classifier as positive instances. We re-labeled any instances as positive that matched a gene identifier associated with the abstract regardless of the (potentially incorrect) label associated with the identifier. The Task 1B dataset creators marked an identifier "no" if an exact lexicon match wasn't found in the abstract. As our system matching phase is a bit different (i.e. we remove punctuation and ignore case), this amounts to re-labeling the training data using this looser criterion. The results of this *match-based re-labeling* are shown in Table 1 below.

	Baseline	Re-labeled
Mouse	68.8	72.0
Fly	70.8	75.3
Yeast	89.7	90.0

Table 1 Balanced F-measure scores comparing the baseline vs. a system trained with the match-based re-labeled instances on the development test data.

5.3 Experiment Set 2: Effect of Weakly Supervised Re-labeling

In our next set of experiments we tested a number of different weakly supervised learning configurations. These different methods simply amount to different rankings of the instances to re-label (based on confidence and the gene name tags). The basic algorithm (outlined in Figure 1) remains the same in all cases. Specifically, we investigated three methods for ranking the instances to re-label: 1) naïve self-training, 2) self-training with bagging, and 3) co-training.

Naïve self-training consisted of training a single maximum entropy classifier with the full feature set on each partition and using it to re-label instances from the other partition based on confidence.

Self training with bagging followed the same idea but used bagging. For each partition, we trained 20 separate classifiers on random subsets of the training data using the full feature set. The confidence assigned to a test instance was then defined as the product of the confidences of the individual classifiers.

Co-training involved training two classifiers for each partition with feature split. We split the features into *context-based* features such as the surrounding words and the number of gene ids matching the current phrase, and *lexically-based* features that included the phrase itself, affixes, the number of tokens in the phrase, etc. We computed the aggregated confidences for each instance as the product of the confidences assigned by the resulting context-based and lexically-based classifiers.

We ran experiments for each of these three options both *with* the gene tagger and *without* the gene tagger. The systems that included the gene tagger ranked all instances derived from tagged phrases above all instances derived from phrases that were not tagged regardless of the classifier confidence.

A final experimental condition we explored was comparing batch re-labeling vs. incremental re-labeling. Batch re-labeling involved training the classifiers once and re-labeling all *k* instances using the same classifier. Incremental re-labeling consisted of iteratively re-labeling *n* instances over *k/n* epochs where the classifiers were re-trained on each epoch with the newly re-labeled training data. Interestingly, incremental re-labeling did not perform better than batch re-labeling in our experiments. All results reported here, therefore, used batch re-labeling.

After the training data was re-labeled, a single maximum entropy classifier was trained on the entire (now re-labeled) training set. This resulting classifier was then applied to the development set in the manner described in Section 3.

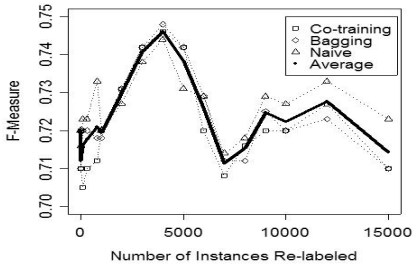
MAX	With Tagger	Without Tagger
Self-Naïve	74.4 (4000)	72.3 (5000)
Self-Bagging	74.8 (4000)	73.5 (6000)
Co-Training	74.6 (4000)	72.7 (6000)

AVG	With Tagger	Without Tagger
Self-Naïve	72.2	71.2
Self-Bagging	72.2	71.5
Co-Training	71.9	71.2

Table 2. Maximum and average balanced f-measure scores on the mouse data set for each of the six system configurations for all values of *k* – the number of instances re-labeled. The numbers in parentheses indicate for which value of *k* the maximum value was achieved.

We tested each of these six configurations for different values of *k*, where *k* is the total number of instances re-labeled[3]. Table 2 highlights the maximum and average balanced f-measure scores across all values of *k* for the different system configurations. Both the maximum and averaged scores appear noticeably higher when constraining the instances to re-label with the tagger. The three weakly supervised methods perform comparably with bagging performing slightly better.

[3] The values of *k* considered here were: 0, 10, 20, 50, 100, 200, 300, 500, 800, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 12000 and 15000.



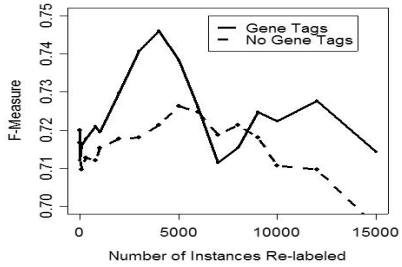


Figure 3. The top graph shows balanced F-measure scores against the number of instances re-labeled when using the tagger as a constraint. The bottom graph compares the re-labeling of instances with the gene tagger as a constraint and without.

In order to gain further insight into re-labeling instances, we have plotted the balanced F-measure performance on the development test for various values of *k*. The upper graph indicates that the three different methods correlate strongly. The bottom graph makes apparent the benefits of tagging as a constraint. It also points to the weakness of the tagger, however. At *k*=7000 and *k*=8000, the system tends to perform *worse* when using the tags as a constraint. This indicates that tagger recall errors have the potential to filter out good candidates for re-labeling.

Another observation from the graphs is that performance actually drops for small values of *k*. This would imply that many of the instances the classifiers are most confident about re-labeling are in fact spurious. To support this hypothesis, we trained the baseline system on the entire training set and computed its *calibration error* on the development test data. The calibration error measures how "realistic" the probabilities output by the classifier are. See [13] for details.

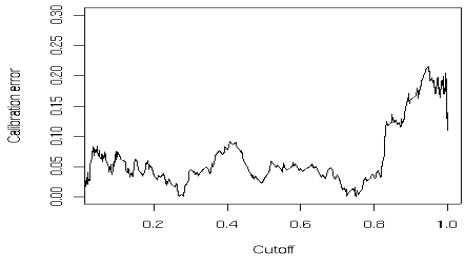


Figure 4. Classifier calibration error on the development test data.

Figure 4 illustrates the estimated calibration error at different thresholds. As can be seen, the error is greatest for high confidence values indicating that the classifier is indeed very confidently predicting an instance as positive when it is negative. Extrapolating this calibration error to the re-labling classifiers (each trained on one half of the training data) offers some explanation as to why re-labeling starts off so poorly. The error mass is exactly where we do not want it - at the highest confidence values. This also offers an explanation as to why incremental re-labeling did not help. Fortunately, introducing a gene tagger as a constraint mitigates this problem.

5.4 Experiment Set 3: Final Evaluation

We report our results using the best overall system configurations on the Task 1B evaluation data. We "submitted" 3 runs for two different *mouse* configurations and one for both *fly* and *yeast*. The highest scores over the 3 runs are reported in Table 3. *MouseWS* used the best weakly supervised method as determined on the development test data: *bagging* with k=4000. *MouseMBR*, *YeastMBR* and *FlyMBR* used match-based re-labeling described in Section 5.2. The Gaussian prior was set to 2.0 for all runs and the 3 submissions for each configuration only varied in the threshold value *T*.

	F-measure	Precision	Recall
MouseWS	0.784	0.81	0.759
MouseMBR	0.768	0.795	0.743
FlyMBR	0.767	0.767	0.767
YeastMBR	0.902	0.945	0.902

Table 3. Final evaluation results.

These results are competitive compared with the BioCreAtIvE Task 1B results where the highest F-measures for mouse, fly and yeast were 79.1, 81.5 and 92.1 with the medians at 73.8, 66.1 and 85.8, respectively. The results for mouse and fly improve upon previous best reported results with an organism invariant, automatic system [1].

6 Conclusions

The quality of training data is paramount to the success of fully automatic, organism invariant approaches to the normalization problem. In this paper we have demonstrated the utility of weakly supervised learning methods in conjunction with a gene name tagger for re-labeling noisy training data for gene name normalization. The result being higher quality data with corresponding higher performance on the BioCreAtIvE Task 1B gene name normalization task.

Future work includes applying method outlined here for correcting noisy data to other classification problems. Doing so generally requires an independent "filter" to restrict re-labeling – the equivalent of the gene tagger used here. We also have plans to improve classifier calibration. Integrating confidence estimates produced by the gene name tagger, following [14], is another avenue for investigation.

Acknowledgements

We thank Alex Morgan, Lynette Hirschman, Marc Colosimo, Jose Castano and James Pustejovsky for helpful comments and encouragement. This work was supported under MITRE Sponsored Research 51MSR123-A5.

References

1. Crim, J., R. McDonald, and F. Pereira. *Automatically Annotating Documents with Normalized Gene Lists*. in *EMBO Workshop - A critical assessment of text mining methods in molecular biology*. 2004. Granada, Spain.
2. Blum, A. and T. Mitchell. *Combining Labeled and Unlabeled Data with Co-training*. 1998. Proceedings of the Workshop on Computational Learning Theory: Morgan Kaufmann.
3. Banko, M. and E. Brill. *Scaling to very very large corpora for natural language disambiguation*. in *ACL/EACL*. 2001.
4. Ng, V. and C. Cardie. *Weakly Supervised Natural Language Learning Without Redundant Views*. in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*. 2003.
5. Hirschman, L., et al., *Overview of BioCreAtIvE task 1B: Normalized Gene Lists.* BioMed Central BioInformatics, 2005(Special Issue on BioCreAtIvE).
6. Craven, M. and J. Kumlien, *Constructing Biological Knowledge Bases by Extracting Information from Text Sources.* 1999: p. 77-86.
7. Morgan, A., et al., *Gene Name Extraction Using FlyBase Resources.* ACL Workshop on Natural Language Processing in Biomedicine, 2003.
8. Morgan, A.A., et al., *Gene name identification and normalization using a model organism database.* J Biomed Inform, 2004. **37**(6): p. 396-410.
9. Nigam, K. and R. Ghani. *Analyzing the effectiveness and applicability of co-training*. in *Information and Knowledge Management*. 2000.
10. Kim, J.-D., et al., *GENIA Corpus -- a semantically annotated corpus for bio-text mining.* Bioinformatics, 2003. **19**((Supppl 1)): p. 180-182.
11. Lafferty, J., A. McCallum, and F. Pereira. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. in *18th International Conf. on Machine Learning*. 2001. San Francisco, CA: Morgan Kaufmann.
12. McDonald, R. and F. Pereira. *Identifying Gene and Protein Mentions in Text Using Conditional Random Fields*. in *A critical assessment of text mining methods in molecular biology, BioCreative 2004*. 2004. Grenada, Spain.
13. Cohen, I. and M. Goldszmidt. *Properties and Benefits of Calibrated Classifiers*. in *EMCL/PKDD*. 2004. Pisa, Italy.
14. Culotta, A. and A. McCallum. *Confidence Estimation for Information Extraction*. in *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*. 2004. Boston, MA.

Adaptive String Similarity Metrics for Biomedical Reference Resolution

Ben Wellner[†*]
[*]The MITRE Corporation
202 Burlington Rd
Bedford MA 01730
wellner@mitre.org

José Castaño[†] **and James Pustejovsky**[†]
[†]Computer Science Department
Brandeis University
Waltham MA 02454
{jcastano,jamesp}@cs.brandeis.edu

Abstract

In this paper we present the evaluation of a set of string similarity metrics used to resolve the mapping from strings to *concepts* in the UMLS MetaThesaurus. String similarity is conceived as a single component in a full Reference Resolution System that would resolve such a mapping. Given this qualification, we obtain positive results achieving 73.6 F-measure (76.1 precision and 71.4 recall) for the task of assigning the correct UMLS concept to a given string. Our results demonstrate that adaptive string similarity methods based on Conditional Random Fields outperform standard metrics in this domain.

1 Introduction

1.1 String Similarity and Reference Resolution

String similarity/matching algorithms are used as a component in *reference resolution* algorithms. We use reference resolution in a broad sense, which includes any of the following aspects:

a. Intra-document noun phrase reference resolution.

b. Cross-document or corpus reference resolution.

c. Resolution of entities found in a corpus with databases, dictionaries or other external knowledge sources. This is also called semantic integration, e.g., (Li et al., 2005), reference grounding, e.g., (Kim and Park, 2004) or normalization, e.g., (Pustejovsky et al., 2002; Morgan et al., 2004).

The last two aspects of reference resolution are particularly important for information extraction, and the interaction of reference resolution with information extraction techniques (see for example Bagga (1998)). The extraction of a particular set of entities from a corpus requires reference resolution for the set of entities extracted (e.g., the EDT task in ACE[1]), and it is apparent that there is more variation in the cross-document naming conventions than in a single document.

The importance of edit distance algorithms has already been noticed, (Müller et al., 2002) and the importance of string similarity techniques in the biomedical domain has also been acknowledged, e.g., (Yang et al., 2004).

String similarity/matching algorithms have also been used extensively in related problems such as Name databases and similar problems in structured data, see (Li et al., 2005) and references mentioned therein.

The problem of determining whether two similar strings may denotate the same *entity* is particularly challenging in the biomedical literature. It has already been noticed (Cohen et al., 2002) that there is great variation in the naming conventions, and noun phrase constructions in the literature. It has also been noticed that bio-databases are hardly ever updated with the names in the literature (Blaschke

[1]http://www.nist.gov/speech/tests/ace/

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 9–16, Detroit, June 2005.

et al., 2003). A further complication is that the actual *mentions* found in text are more complex than just *names* - including descriptors, in particular. Finally, ambiguity (where multiple entities have the same name) is very pervasive in biomedicine.

In this paper we investigate the use of several string similarity methods to group together *string mentions* that might refer to the same *entity* or *concept*. Specifically, we consider the sub-problem of assigning an unseen mention to one of a set of existing unique entities or concepts, each with an associated set of known synonyms. As our aim here is focusing on improving string matching, we have purposely factored out the problem of ambiguity (to the extent possible) by using the UMLS MetaThesaurus as our data source, which is largly free of strings that refer to multiple entities. Thus, our work here can be viewed an important piece in a larger normalization or reference resolution system that resolves ambiguity (which includes filtering out mentions that don't refer to any entity of interest).

The experiments reported on in this paper evaluate a suite of robust string similarity techniques. Our results demonstrate considerable improvement to be gained by using *adaptive* string similarity metrics based on Conditional Random Fields customized to the domain at hand. The resulting best metric, we term SoftTFIDF-CRF, achieves 73.6 F-measure on the task of assigning a given string to the correct concept. Additionally, our experiments demonstrate a tradeoff between efficiency and recall based on q-gram indexing.

2 Background

2.1 Entity Extraction and Reference Resolution in the Biomedical Domain

Most of the work related to reference resolution in this domain has been done in the following areas: a) Intra-document Reference resolution, e.g (Castaño et al., 2002; Lin and Liang, 2004) b) Intra-document Named entity recognition (e.g Biocreative Task 1A (Blaschke et al., 2003), and others), also called classification of biological names (Torii et al., 2004) c) Intra-document alias extraction d) cross-document Acronym-expansion extraction, e.g., (Pustejovsky et al., 2001). e) Protein names resolution against database entries in SwissProt, *protein name grounding*, in the context of a relation extraction task (Kim and Park, 2004). One constraint in these approaches is that they use *several patterns* for the string matching problem. The results of the protein name grounding are 59% precision and 40% recall. The Biocreative Task 1B task challenged systems to *ground* entities found in article abstracts which contain mentions of genes in Fly, Mouse and Yeast databases. A central component in this task was resolving ambiguity as many gene names refer to multiple genes.

2.2 String Similarity and Ambiguity

In this subsection consider the string similarity issues that are present in the biology domain in particular. The task we consider is to associate a string with an existing *entity*, represented by a set of known strings. Although the issue of ambiguity is present in the examples we give, it cannot be resolved by using string similarity methods alone, but instead by methods that take into account the context in which those strings occur.

The protein name *p21* is ambiguous at least between two entities, mentioned as *p21-ras* and *p21/Waf* in the literature. A biologist can look at a set of descriptions and decide whether the strings are ambiguous or correspond to any of these two (or any other entity).

The following is an example of such a mapping, where *R* corresponds to *p21-ras, W* to *p21(Waf)* and *G* to another entity (the gene). Also it can be noticed that some of the mappings include subcases (e.g., R.1).[2]

String Form	Entity
ras-p21 protein	R
p21	R/W
p21(Waf1/Cip1)	W
cyclin-dependent kinase-I p21(Waf-1)	W
normal ras p21 protein	R
pure v-Kirsten (Ki)-ras p21	R.1
wild type p21	R/W
synthetic peptide P21	R/W.2
p21 promoter	G
transforming protein v-p21	R.3
v-p21	R.3
p21CIP1/WAF1	W
protein p21 WAF1/CIP1/Sd:1	W

Table 1: A possible mapping from strings to entities.

[2] All the examples were taken from the MEDLINE corpus.

If we want to use an external knowlege source to produce such a mapping, we can try to map it to concepts in the UMLS Methatesaurus and entries in the SwissProt database.

These two entities correspond to the concepts C0029007 (p21-Ras) and C0288472 (p21-Waf) in the UMLS Methathesaurus. There are 27 strings or *names* in the UMLS that map to C0288472 (Table 2):

oncoprotein p21	CAP20
CDK2-associated protein 20 kDa	MDA 6
Cdk2 inhibitor	WAF1 CIP1
Cdk-interacting protein	cdn1 protein
CDK-Interacting Protein 1	CDKN1A
CDKN1 protein	Cip1 protein
Cip-1 protein	mda-6 protein
Cyclin-Dependent Kinase Inhibitor 1A	p21
p21 cell cycle regulator	p21(cip1)
p21 cyclin kinase inhibitor	p21(waf1-cip1)
Pic-1 protein (cyclin)	p21-WAF1
senescent cell-derived inhibitor protein 1	protein p21
CDKN1A protein	WAF1 protein
WAF-1 Protein	

Table 2: UMLS strings corresponding to C0288472

There are 8 strings that map to concept C0029007 (Table 3).

Proto-Oncogene Protein p21(ras)	p21(c-ras)
p21 RAS Family Protein	p21 RAS Protein
Proto-Oncogene Protein ras	c-ras Protein
ras Proto-Oncogene Product p21	p21(ras)

Table 3: UMLS strings corresponding to C0029007

It can be observed that there is only one exact match: *p21* in C0288472 and Table 1. It should be noted that *p21*, is not present in the UMLS as a possible string for C0029007. There are other close matches like *p21(Waf1/Cip1)* (which seems very frequent) and *p21(waf1-cip1)*.

An expression like *The inhibitor of cyclin-dependent kinases WAF1 gene product p21* has a high similarity with *Cyclin-Dependent Kinase Inhibitor 1 A* and *The cyclin-dependent kinase-I p21(Waf-1)* partially matches *Cyclin-Dependent Kinase*

However there are other mappings which look quite difficult unless some context is given to provide additional clues (e.g., *v-p21*).

The SwissProt entries **CDN1A_FELCA**, **CDN1A_HUMAN** and **CDN1A_MOUSE** are related to *p21(Waf)*. They have the following set of common description names:

Cyclin-dependent kinase inhibitor 1, p21, CDK-interacting protein 1.[3]

There is only one entry in SwissProt related to *p21-ras*: Q9PSS8_PLAFE: with the description name *P21-ras protein* and a related gene name: *Ki-ras*.

It should be noted that SwissProt classifies, as different entities, the proteins that refer to different organisms. The UMLS MetaThesaurus, on the other hand, does not make this distinction. Neither is this distinction always present in the literature.

3 Methods for Computing String Similarity

A central component in the process of normalization or reference resolution is computing string similarity between two strings. Methods for measuring string similarity can generally be broken down into character-based and token-based approaches.

Character-based approaches typically consist of the edit-distance metric and variants thereof. Edit distance considers the number of edit operations (addition, substitution and deletion) required to transform a string s_1 into another string s_2. The Levenstein distance assigns unit cost to all edit operations. Other variations allow arbitrary costs or special costs for starting and continuing a "gap" (i.e., a long sequence of adds or deletes).

Token-based approaches include the Jaccard similarity metric and the TF/IDF metric. The methods consider the (possibly weighted) overlap between the tokens of two strings. Hybrid token and character-based are best represented by SoftTFIDF, which includes not only exact token matches but also close matches (using edit-distance, for example). Another approach is to perform the Jaccard similarity (or TF/IDF) between the q-grams of the two strings instead of the tokens. See Cohen et al. (2003) for a detailed overview and comparison of some of these methods on different data sets.

[3] There are two more description names for the human and mouse entries. The SwissProt database has also associated Gene names to those entries which are related to some of the possible names that we find in the literature. Those gene names are: *CDKN1A, CAP20, CDKN1, CIP1, MDA6, PIC1, SDI1, WAF1, Cdkn1a, Cip1, Waf1*. It can be seen that those names are incorporated in the UMLS as protein names.

Recent work has also focused on automatic methods for adapting these string similarity measures to specific data sets using machine learning. Such approaches include using classifiers to weight various fields for matching database records (Cohen and Richman, 2001). (Belenko and Mooney, 2003) presents a generative, Hidden Markov Model for string similarity.

4 An Adaptive String Similarity Model

Conditional Random Fields (CRF) are a recent, increasingly popular approach to sequence labeling problems. Informally, a CRF bears resemblance to a Hidden Markov Model (HMM) in which, for each input position in a sequence, there is an observed variable and a corresponding hidden variable. Like HMMs, CRFs are able to model (Markov) dependencies between the hidden (predicted) variables. However, because CRFs are conditional, discriminatively trained models, they can incorporate arbitrary overlapping (non-independent) features over the entire input space — just like a discriminative classifier.

CRFs are log-linear models that compute the probability of a state sequence, $\vec{s} = (s_1, s_2, ..., s_T)$, given an observed sequence, $\vec{o} = (o_1, o_2, ..., o_T)$ as:

$$P(\vec{s}|\vec{o}) = \frac{1}{Z_{\vec{o}}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(s_{t-1}, s_t, \vec{o}, t)\right)$$

where the f_k are arbitrary feature functions, the λ_k are the model parameters and $Z_{\vec{o}}$ is a normalization function.

Training a CRF amounts to finding the λ_k that maximize the conditional log-likelihood of the data.

Given a trained CRF, the inference problem involves finding the most likely state sequence given a sequence of observations. This is done using a slightly modified version of the Viterbi algorithm (See Lafferty et al. (2001) more for details on CRFs).

4.1 CRFs for String Similarity

CRFs can be used to measure string similarity by viewing the observed sequence, $\vec{o}$, and the state sequence, $\vec{s}$, as sequences of characters. In practice we are presented with two strings, q_1, and q_2 of possibly *differing* lengths. A necessary first step is to align the two strings by applying the Levenstein distance procedure as described earlier. This produces a series of edit operations where each operation has one of three possible forms: 1) $\epsilon \rightarrow o_j$ (addition), 2) $s_i \rightarrow o_j$ (substitution) and 3) $s_i \rightarrow \epsilon$ (deletion). The observed and hidden sequences are then derived by reading off the terms on the right and left-hand sides of the operations, respectively. Thus, the possible state values include all the characters in our domain plus the special null character, ϵ.

Feature Description	Variables
State uni-gram	(s_i)
State bi-gram	(s_{i-1}, s_i)
Obs. uni-gram; state uni-gram	(o_i, s_i)
Obs. bi-gram; state uni-gram	(o_{i-1}, o_i, s_i)
Obs. is *punctuation* and state uni-gram	(o_i, s_i)
Obs. is a *number* and state uni-gram	(o_i, s_i)

Table 4: Features used for string similarity

We employ a set of relatively simple features in our string similarity model described in Table 4. One motivation for keeping the set of features simple was to determine the utility of string similarity CRFs without spending effort designing domain-specific features; this is a primary motivation for taking a machine learning approach in the first place. Additionally, we have found that more specific, discriminating features (e.g., observation tri-grams with state bi-grams) tend to reduce the performance of the CRF on this domain - in some cases considerably.

4.2 Practical Considerations

We discuss a few practical concerns with using CRFs for string similarity.

The first issue is how to scale CRFs to this task. The inference complexity for CRFs is $O(s^2t)$ where s is the size of the vocabulary of states and t is the number of input positions. In our setting, the number of state variable values is very large - one for each character in our alphabet (which is on the order of 40 or more including digits and punctuation). Moreover, we typically have very large training sets largely due to the fact that $\binom{n}{2}$ training pairs are derivable from an equivalence class of size n.

Given this situation, standard training for CRFs becomes unwieldy, since it involves performing inference over the entire data set repeatedly (typically a few hundred iterations are required to converge).

As such, we resort to an approximation: Voted Perceptron training (Collins, 2002). Voted Perceptron training does not involve maximizing log-likelihood, but instead updates parameters via stochastic gradient descent with a small number of passes over the data.

Another consideration that arises is given a pair of strings, which one should be considered the "observed" sequence and which one the "hidden" sequence.

Another consideration that arises is given a pair of strings, which string should be considered the "observed" sequence and which the "hidden" sequence?[4] We have taken to always selecting the longest string as the "observed" string, as it appears most natural, though that decision is somewhat arbitrary.

A last observation is that the probability assigned to a pair of strings by the model will be reduced geometrically for longer string pairs (since the probability is computed as a product of t terms, where t is the length of the sequence). We have taken to normalizing the probabilities by the length of the sequence roughly following the approach of (Belenko and Mooney, 2003).

A final point here is that it is possible to use Viterbi decoding to find the n-best hidden strings given *only* the observed string. This provides a mechanism to generate domain-specific string alterations for a given string ranked by their probability. The advantage of this approach is that such alterations can be used to expand a synonym list; exact matching can then be used greatly increasing efficiency. Work is ongoing in this area.

5 Matching Procedure

Our matching procedure in this paper is set in the context of finding the concept or entity (each with some existing set of known strings) that a given string, s, is referring to. In many settings, such as the BioCreative Task 1B task mentioned above, it is necessary to match large numbers of strings against the lexicon - potentially every possible phrase in a large number of documents. As such, very fast matching times (typically on the order of milliseconds) are required.

Our method can be broken down into two steps. We first select a reasonable candidate set of strings (associated with a concept or lexical entry), $S = s_1, s_2, ..., s_n$, reasonably similar to the given string s using an efficient method. We then use one of a number of string similarity metrics on all the pairs: $\langle s, s_1 \rangle, \langle s, s_2 \rangle, ...\langle s, s_n \rangle$

The set of candidate strings, $s_1, s_2, ..., s_n$ is determined by the *q-gram match ratio*, which we define as:

$$qRatio(s, s_i) = 1 - \frac{|ng(s) \cap ng(s_i)|}{|ng(s) \cup ng(s_i)|}$$

where $ng(x) = \{y|$ such that y is a q-gram of $x\}$. This set is retrieved very quickly by creating a q-gram index: a mapping between each q-gram and the strings (entries) in which it occurs. At query time, the given string is broken into q-grams and the sets corresonding to each q-gram are retrieved from the index. A straightforward computation finds those entries that have a certain number of q-grams in common with the query string s from which the ratio can be readily computed.

Depending on the setting, three options are possible given the returned set of candidates for a string s:

1. Consider s and s_i equivalent where s_i is the most similar string
2. Consider s and s_i equivalent where s_i is the most similar string and $sim(s, s_i) \geq T$, for some threshold T
3. Consider s and s_i equivalent for *all* s_i where $sim(s, s_i) \geq T$, for some threshold T

In the experiments in this paper, we use the first criterion since for a given string, we know that it should be assigned to exactly one concept (see below).

6 Experiments and Results

6.1 Data and Experimental Setup

We used the UMLS MetaThesaurus for all our experiments for three reasons: 1) the UMLS represents a wide-range of important biomedical concepts

[4] Note that a standard use for models such as this is to find the most likely hidden sequence given *only* the observed sequence. In our setting here we are provided the hidden sequence and wish to compute it's (log-)probability given the observed sequence.

for many applications and 2) the size of the UMLS (compared with BioCreative Task 1B, for example) promotes statistically significant results as well as sufficient training data 3) the problem of ambiguity (multiple concepts with the same name) is largely absent in the UMLS.

The UMLS is a taxonomy of medical and clinical concepts consisting of 1,938,701 lexical entries (phrase strings) where each entry belongs to one (or, in very rarely, more than one) of 887,688 concepts. We prepared the data by first selecting only those lexical entries belonging to a concept containing 12 or more entries. This resulted in a total of 129,463 entries belonging to 7,993 concepts. We then divided this data into a training set of 95,167 entries and test set of 34,296 entries where roughly 70% of the entries for each concept were placed in the training set and 30% in the test set. Thus, the training set and test set both contained some string entries for each of the 7,993 concepts. While restricting the number of entries to 12 or more was somewhat arbitrary, this allowed for at least 7 (70% of 12) entries in the training data for each concept, providing sufficient training data.

The task was to assign the correct concept identifier to each of the lexical entries in the test set. This was carried out by finding the most similar string entry in the training data and returning the concept identifier associated with that entry. Since each test instance must be assigned to exactly one concept, our system simply ranked the candidate strings $s_1, s_2 ..., s_n$ based on the string similarity metric used. We compared the results for different maximum q-gram match ratios. Recall that the q-gram match mechanism is essentially a filter; higher values correspond to larger candidate pools of strings considered by the string similarity metrics.

We used six different string similarity metrics that were applied to the same set of candidate results returned by the q-gram matching procedure for each test string. These were **TFIDF**, **Levenstein**, **q-gram-Best**, **CRF**, **SoftTFIDF-Lev** and **SoftTFIDF-CRF**. **TFIDF** and **Levenstein** were described earlier. The **q-gram-Best** metric simply selects the match with the lowest q-gram match ratio returned by the q-gram match procedure described above[5]. The **SoftTFIDF-Lev** model is the SoftTFIDF metric described earlier where the secondary metric for similarity between pairs of tokens is the Levenstein distance.

	Precision	Recall	F-measure
SoftTFIDF-CRF(0.5)	0.761	0.714	0.736
SoftTFIDF-Lev(0.5)	0.742	0.697	0.718
CRF(0.6)	0.729	0.705	0.717
q-gram Best(0.475)	0.714	0.658	0.685
Levenstein(0.4)	0.710	0.622	0.663
TFIDF(0.325)	0.730	0.576	0.644

Table 5: Maximum F-measure attained for each string similarity metric, with corresponding precision and recall values. The numbers in parentheses indicate the q-gram match value for which the highest F-measure was attained.

The **CRF** metric is the CRF string similarity model applied to the entire strings. This model was trained on pairs of strings that belonged to the same concept in the training data, resulting in 130,504 string pair training instances. The **SoftTFIDF-CRF** metric is the SoftTFIDF method where the secondary metric is the CRF string similarity model. This CRF model was trained on pairs of tokens (not entire phrases). We derived pairs of tokens by finding the most similar pairs of tokens (similarity was determined here by Levenstein distance) between strings belonging to the same concept in the training data. This resulted in 336,930 string pairs as training instances.

6.2 Results

We computed the precision, recall and F-measure for each of the string similarity metrics across different q-gram match ratios shown in Fig. 1. Both a precision *and* recall error is introduced when the top-returned concept id is incorrect; just a recall error occurs when no concept id is returned at all - i.e. when the q-gram match procedure returns the empty set of candidate strings. This is more likely to occur when for lower q values and explains the poor recall in those cases. In addition, we computed the *mean reciprocal rank* of each of the methods. This is computed using the ranked, ordered list of the concepts returned by each method. This scoring method as-

[5]This is essentially the Jaccard similarity metric over q-grams instead of tokens

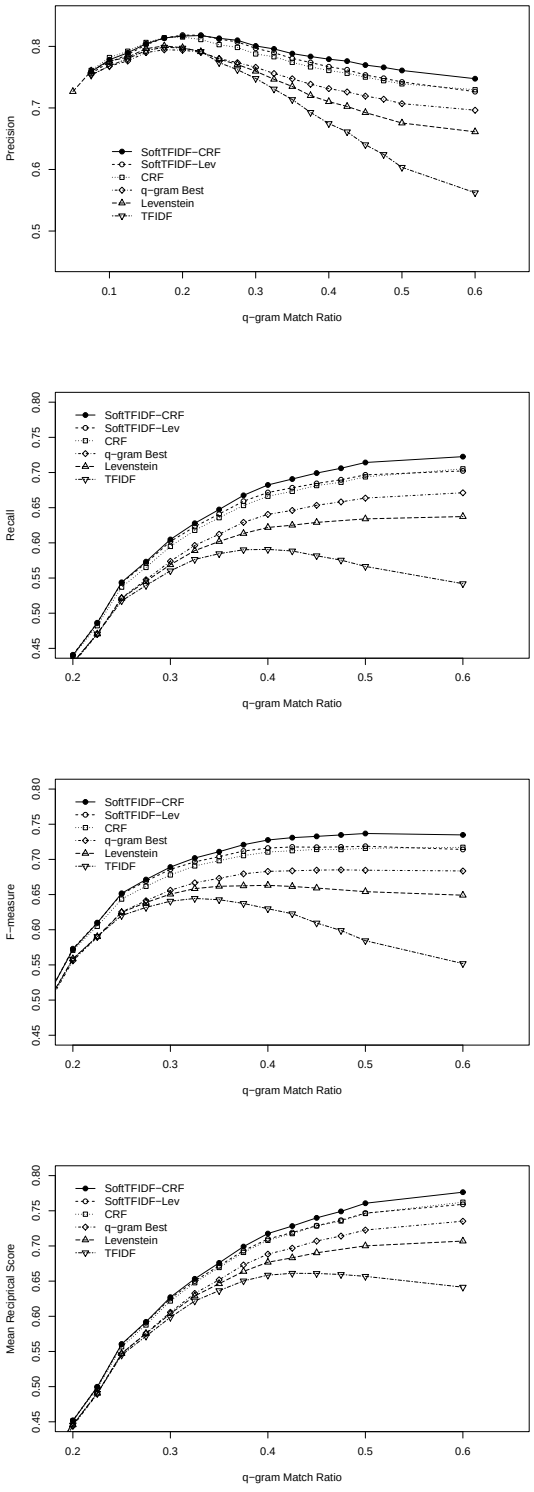


Figure 1: Precision, Recall, F-measure and Mean Reciprocal Rank comparisions for each string similarity metric across different q-gram match ratios.

signs a score of $1/R$ for each test instance where R is the position in the ranked list at which the correct concept is found. For example, by returning the correct concept as the 4th element in the ranked list, a method is awarded $1/4 = 0.25$. The *mean reciprocal rank* is just the average score over all the test elements.

As can be seen, the **SoftTFIDF-CRF** string-similarity metric out-performs all the other methods on this data set. This approach is robust to both word order variations and character-level differences, the latter with the benefit of being adapted to the domain. Word order is clearly a critical factor in this domain[6] though the **CRF** metric, entirely character-based, does surprisingly well - much better than the Levenstein distance. The **q-gram-Best** metric, being able to handle word order variations and character-level differences, performs fairly.

The graphs illustrate a tradeoff between efficiency and accuracy (recall). Lower q-gram match ratios return fewer candidates with correspondingly fewer pairwise string similarities to compute. Precision actually peaks with a q-gram match ratio of around 0.2. Recall tapers off even up to high q-gram levels for all metrics, indicating that nearly 30% of the test instances are probably too difficult for any string similarity metric. Error analysis indicates that these cases tend to be entries involving synonymous "nicknames". Acquiring such synonyms requires other machinery, e.g., (Yu and Agichtein, 2003).

7 Conclusions

We have explored a set of string similarity metrics in the biological domain in the service of reference resolution. String similarity is only one parameter to be considered in this task. We presented encouraging results for assigning strings to UMLS concepts based solely on string similarity metrics — demonstrating that adaptive string similarity metrics show significant promise for biomedical text processing. Further progress will require a system that 1) utilizes context of occurrence of respective strings for handling ambiguity and 2) further improves recall

[6] Inspection of the data indicates that the purely character-based methods are more robust than one might think. There are at least 8 strings to match against for a concept and it is likely that at least one of them will have similar word order to the test string.

through expanded synonyms.

Future work should also consider the *dependent* nature (via transitivity) of reference resolution. Comparing a test string against *all* (current) members of an equivalence class and considering multiple, similar test instances simultaneously (McCallum and Wellner, 2003) are two directions to pursue in this vein.

8 Acknowledgements

We thank Dave Harris, Alex Morgan, Lynette Hirschman and Marc Colosimo for useful discussions and comments. This work was supported in part by MITRE Sponsored Research 51MSR123-A5.

References

A. Bagga. 1998. *Coreference, cross-document coreference, and information extraction methodologies*. Ph.D. thesis, Duke University. Supervisor-Alan W. Biermann.

M. Belenko and R. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Datamining*, pages 39–48, Washington D.C.

C. Blaschke, L. Hirschman, A. Yeh, and A. Valencia. 2003. Critical assessment of information extraction systems in biology. *Comparative and Functional Genomics*, pages 674–677.

J. Castaño, J. Zhang, and J. Pustejovsky. 2002. Anaphora resolution in biomedical literature. In *International Symposium on Reference Resolution*, Alicante, Spain.

William Cohen and Jacob Richman. 2001. Learning to match and cluster entity names. In *ACM SIGIR-2001 Workshop on Mathematical/Formal Methods in Information Retrieval*, New Orleans, LA, September.

K. Bretonnel Cohen, Andrew Dolbey, George Acquaah-Mensah, and Lawrence Hunter. 2002. Contrast and variability in gene names. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 14–20, Philadelphia, July. Association for Computational Linguistics.

W. Cohen, P. Ravikumar, and S. Fienburg. 2003. A comparison of string metrics for matching names and records. In *KDD Workshop on Data Cleaning and Object Consolidation*.

Michael Collins. 2002. Discriminative training methods for hidden markove models: Theory and experiments with perceptron algorithms. In *EMNLP 2002*.

Yu H, Hatzivassiloglou V, Friedman C, Rzhetsky A, and Wilbur W. 2002. Automatic extraction of gene and protein synonyms from medline and journal articles. In *Proc AMIA Symposium*, pages 919–23.

Jung-Jae Kim and Jong C. Park. 2004. Bioar: Anaphora resolution for relating protein names to proteome database entries. In Sanda Harabagiu and David Farwell, editors, *ACL 2004: Workshop on Reference Resolution and its Applications*, pages 79–86, Barcelona, Spain, July. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

X. Li, P. Morie, and D. Roth. 2005. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*.

Y. Lin and T. Liang. 2004. Pronominal and sortal anaphora resolution for biomedical literature. In *Proceedings of ROCLING XVI: Conference on Computational Linguistics and Speech Processing*, Taipei, Taiwan.

Andrew McCallum and Ben Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, pages 79–86, Acapulco, Mexico, August.

A. Morgan, L. Hirschman, M. Colosimo, A. Yeh, and J. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, (6):396–410.

Christoph Müller, Stefan Rapp, and Michael Strube. 2002. Applying co-training to reference resolution. In *ACL*, pages 352–359.

J. Pustejovsky, J. Castaño, B. Cochran, M. Kotecki, and M. Morrell. 2001. Automatic extraction of acronym-meaning pairs from medline databases. In *Proceedings of Medinfo, London*.

J. Pustejovsky, J. Castaño, J. Zhang, R. Sauri, and W. Luo. 2002. Medstract: creating large-scale information servers from biomedical texts. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 85–92, Philadelphia, July. Association for Computational Linguistics.

M. Torii, S. Kamboj, and K. Vijay-Shanker. 2004. Using name-internal and contextual features to classify biological terms. *Journal of Biomedical Informatics*, pages 498–511.

X. Yang, G. Zhou, J. Su, and C. L. Tan. 2004. Improving noun phrase coreference resolution by matching strings. In *Proceedings of 1st Internation Joint Conference of Natural Language Processing*, pages 326–333.

H. Yu and E. Agichtein. 2003. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, pages 340–349.

Unsupervised gene/protein named entity normalization using automatically extracted dictionaries

Aaron M. Cohen
Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University, Portland, OR, USA
cohenaa@ohsu.edu

Abstract

Gene and protein named-entity recognition (NER) and normalization is often treated as a two-step process. While the first step, NER, has received considerable attention over the last few years, normalization has received much less attention. We have built a dictionary based gene and protein NER and normalization system that requires no supervised training and no human intervention to build the dictionaries from online genomics resources. We have tested our system on the Genia corpus and the BioCreative Task 1B mouse and yeast corpora and achieved a level of performance comparable to state-of-the-art systems that require supervised learning and manual dictionary creation. Our technique should also work for organisms following similar naming conventions as mouse, such as human. Further evaluation and improvement of gene/protein NER and normalization systems is somewhat hampered by the lack of larger test collections and collections for additional organisms, such as human.

1 Introduction

In the genomics era, the field of biomedical research finds itself in the ironic situation of generating new information more rapidly than ever before, while at the same time individual researchers are having more difficulty getting the specific information they need. This hampers their productivity and efficiency. Text mining has been proposed as a means to assist researchers in handling the current expansion of the biomedical knowledge base (Hirschman *et al.*, 2002). Fundamental tasks in text mining are named entity recognition (NER) and normalization. NER is the identification of text terms referring to items of interest, and normalization is the mapping of these terms to the unique concept to which they refer. Once the concepts of interest are identified, text mining can proceed to extract facts and other relationships of interest that involve these recognized entities. With the current research focus on genomics, identifying genes and proteins in biomedical text has become a fundamental problem in biomedical text mining research (Cohen and Hersh, 2005). The goal of our work here is to explore the potential of using curated genomics databases for dictionary-based NER and normalization. These databases contain a large number of the names, symbols, and synonyms and would likely enable recognition of a wide range of genes on a wide range of literature without corpus-specific training.

Gene and protein NER and normalization can be viewed as a two-step process. The first step, NER, identifies the strings within a sample of text that refer to genes and proteins. The second step, normalization, determines the specific genes and proteins referred to by the text strings.

Many investigators have examined the initial step of gene and protein NER. One of the most successful rules-based approaches to gene and protein NER in biomedical texts has been the AbGene system (Tanabe and Wilbur, 2002), which has been used by several other researchers. After training on hand-tagged sentences from biomedical text, it applies a Brill-style tagger (Brill, 1992) and manually generated post-processing rules. AbGene achieves a precision of 85.7% at a recall of 66.7% (F1 = 75%). Another successful system is GAPSCORE (Chang *et al.*, 2004). It assigns a numeric score to each word in a sentence based on appearance, morphology, and context of the word and then applies a classifier trained on these features. After training on the Yapex corpus (Franzen *et al.*, 2002), the system achieved a precision of 81.5% at a recall of 83.3% for partial matches.

For many applications of text mining, the second step, normalization is as important as the first step. Many biomedical concepts, including genes and proteins, have large numbers of synonymous terms (Yu and Agichtein, 2003, Tuason *et al.*, 2004). Without normalization, different terms for the same concept are treated as distinct items, which can distort statistical and other analysis. Normalization can aggregate references a given gene or protein and can therefore increase the sample size for concepts with common synonyms. However, normalization of gene and protein references has not received as much attention as the NER step.

One recent conference, the BioCreative Critical Assessment for Information Extraction in Biology (Krallinger, 2004), had a challenge task that addressed gene and protein normalization. The task was to identify the specific genes mentioned in a set of abstracts given that the organism of interest was mouse, fly, or yeast. Training and test collections of about 250 abstracts were manually prepared and made available to the participants along with synonym lists. Seven groups participated in this challenge task (Hirschman *et al.*, 2004), with the best F-measures ranging from 92.1% on

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 17–24, Detroit, June 2005.

yeast to 79.1% on mouse. The overall best performing system used a combination of hand built dictionaries, approximate string matching, and parameter tuning based on the training data, and performed match disambiguation using a collection of biomedical abbreviations combined with approximate string match scoring and preferring concepts with a high count of occurring terms (Hanisch *et al.*, 2004).

One thing that almost all of these systems have in common is that they need to be trained on a text corpus and/or use manually built dictionaries based on the training corpus. Since the training corpus may be a small sample of the total relevant biomedical literature, it is uncertain how the performance of these systems will change over time or when applied to other sources of biomedical text. Also, since new genes and proteins are being described all the time, it is unclear how these systems will handle genes discovered after system training is complete. This is may especially be a problem for normalization.

Dictionary-based approaches to gene and protein NER and normalization that require no training have several advantages over orthographic, lexical, and contextual based approaches. Currently there are few test collections for gene and protein normalization, and they are relatively small (Hirschman *et al.*, 2004). Unsupervised systems therefore may perform more uniformly over different data sets and over time for the near future. Since they are not dependent upon training to discover local orthographic or lexigraphic clues, they can recognize long multi-word names as easily as short forms. Dictionary-based approaches can also normalize gene and protein names, reducing many synonyms and phrases representing the same concept to a single identifier for that gene or protein.

In addition, dictionary-based approaches can make use of the huge amount of information in curated genomics databases. Currently, there is an enormous amount of manual curation activity related to gene and protein function. Several genomics databases contain large amounts of curated gene and protein name symbols as well as full names. Groups such as the Human Genome Organisation (HUGO), Mouse Genome Institute (MGI), UniProt, and the National Center for Biotechnology Information (NCBI) collect and organize information on gene and proteins, much of it from the biomedical literature, including gene names, symbols, and synonyms. Dictionary-based approaches provide a way to make use of this information for gene and protein NER and normalization. As the databases are updated by the curating organization, a NER system based on these databases can automatically incorporate additional new names and symbols. These approaches can also be very fast. Much of the computation can be performed during the construction of the dictionary. This can leave the actual searching for dictionary terms a simple and rapid process.

Tsuruoka and Tsujii recently studied the use of dictionary-based approaches for protein name recognition (Tsuruoka and Tsujii, 2004), although they did not evaluate the normalization performance. They applied a probabilistic term variant generator to expand the dictionary, and a Bayesian contextual filter with a sub-sentence window size to classify the terms in the GENIA corpus as likely to represent protein names. Overall they obtained a precision of 71.1%, at a recall of 62.3% and an F-measure of 66.6%. Tsuruoka and Tsujii did not make use of curated database information, and instead split the GENIA corpus into training and test data sets of 1800 and 200 abstracts respectively, and extracted the tagged protein names from the training set to use as a dictionary. These results compare well to, being a bit below, other non-dictionary based methods applied to the GENIA corpus (Lee *et al.*, 2004, Zhou *et al.*, 2004).

In this work we attempt to answer several questions pertaining to dictionary-based gene/protein NER:

- What curated databases provide the best collection of names and symbols?
- Can simple rules generate sufficient orthographic variants?
- Can common English word lists be used to decrease false positives?
- What is the overall normalization performance of an unsupervised dictionary-based approach?

2 Methods

A dictionary-based NER system starts out with a list, potentially very large, of text strings, called *terms*, which represent concepts of interest. In our system, the terms are organized by *concept*, in this case a unique identifier for the gene or protein. All terms for a given concept are kept together. The combination of terms indexed by concept is similar to a traditional thesaurus, and when used for NER and normalization is usually called a *dictionary*. When a term is found in a sample of text, it is a simple process to map the term to the unique gene or protein that it represents. There are several unique identifiers in use by the gene curation organizations, we chose to use the official symbol as a default, but it is easy to use other database identifiers as needed.

2.1 Building the dictionary

Building the initial dictionary is an essential first step in dictionary-based NER. The dictionaries we used in this study were built automatically from five databases

available for download: MGI, Saccharomyces, UniProt (the curated SwissProt portion only), LocusLink, and the Entrez Gene database. For each of these databases, the official symbol, unique identifiers, name, symbol, synonym, and alias fields were extracted. Symbols, synonyms, and aliases corresponding to the same official symbol were combined into a single list. At this stage in dictionary generation, any leading or trailing white space characters are removed. The original capitalization of each term is kept. This will be important in a later step

Like several other investigators (Tanabe and Wilbur, 2002, Chang *et al.*, 2004), we do not discriminate between the names of genes and the proteins that they code for. For many text mining purposes, recognizing a mention of a gene or the coded protein has been treated as equivalent (Cohen and Hersh, 2005). Therefore, combining terms corresponding to the same official symbol is justified, even if one database is composed of genes and the other proteins.

2.2 Generating orthographic variants

Our previous work on gene and protein name synonyms (Cohen *et al.*, 2005) led us to make the observation that many name synonyms are simple orthographic variants of each other, and that most of these variants can be generated with a few simple rules. The next step in dictionary generation is to generate variant terms for each term extracted from the downloaded databases.

Our system uses seven simple rules to generate variants:

(1) If the original term includes internal spaces, these can be replaced by hyphens (e.g., “IL 10” to “IL-10”).

(2) If the original term includes internal hyphens, these can be replaced by spaces (e.g., “mmac-1” to “mmac 1”).

(3) If the original term includes internal spaces or hyphens, these can be removed (e.g., “nf-kappa b” to “nfkappab”).

(4) If the original term ends in a letter followed by single digit, or a letter followed by single digit and then a single letter, a hyphen can be added before the digit (e.g., “NFXL1” to “NFXL-1”).

(5) If the original term ends in a digit, followed by the single letter ‘a’ or ‘b’, we can add a hyphen before the ‘a’ or ‘b’ and also expand ‘a’ to ‘alpha’ and ‘b’ to ‘beta’ (e.g., “epm2b” to “epm2-beta”).

(6) If the original term ends in ‘-1’ or a ‘-2’, replace this ending with the Roman numeral equivalent, ‘-i’ or ‘-ii’ respectively.

(7) For yeast only, if the original term consists of one space-delimited token, append a “p” (see (Cherry, 1995)).

These rules are applied iteratively until no new terms are generated.

2.3 Separating common English words

The next step aids in discriminating mentions of gene and protein names from common English words. The dictionary now contains a large number of terms extracted from the databases along with generated variants. At this point the dictionary is split into two parts. Terms that case-insensitively match a list of common English words are put into the one dictionary, and other terms are put into a separate dictionary.

In practice, this creates a small dictionary of terms easy to confuse with common English words (the *confusion* dictionary) and a much larger dictionary of terms that are not confused with English words (the *main* dictionary). When searching text for gene and protein names, the terms in the smaller dictionary will be handled differently than the terms in the larger dictionary.

For the work presented here, a file of 74,550 common English words was used to filter the terms. This file is available as part of the Moby lexical resource, and is available at (Ward, 2000).

2.4 Screening out the most common English words

Some English words are so common that when they occur they are rarely references to gene and protein names. Our approach includes a list of about 300 English words that is used as a “stop” list. In our system these words are never recognized as gene or protein terms, even if those terms appear in one of the curated databases.

We obtained our list of the 300 most common words in the English language (Carroll *et al.*, 1971). To this list we added a few terms that are commonly found in the biomedical literature that should not be confused with specific gene names. These include “gene”, “genes”, “protein”, “proteins”, ”locus”, ”site”, “alpha”, “beta”, and “as a”.

Terms appearing in this most common word list are removed from both of the dictionaries. The final product of the four preceding steps are two dictionaries, a main dictionary and a confusion dictionary, each which map terms to the unique identifier for the gene/protein symbol corresponding to that term.

2.5 Searching the text

With the two dictionaries complete it is straightforward to search input text for mentions of gene and protein

names. While the algorithm can handle practically any size input text, in practice the input will usually be individual sentences or abstracts, and this is the input size to which we have tuned our system.

For speed and accuracy, we first search the input text for the terms within the dictionary, and if a term is found, we then check to ensure that the matching text is bounded by characters that are acceptable delimiters for gene and protein names. In our system this includes white space characters as well as these characters: .,/\\(){}[]=;?*!". Note that our approach does not prohibit these characters from appearing within the name, only that the matching sequence of characters is bounded by these delimiters. Also, the approach does not require tokenization of the input string. We consider this more flexible than delimiter-based tokenization, which would not allow delimiters to appear within the terms.

This method of searching and checking delimiters is applied for every term in both the main and confusion dictionaries with one essential difference. Case-insensitive search is performed on the terms in the main dictionary. Strict case-sensitive search is performed on terms in the confusion dictionary. This requires terms in the confusion dictionary to exactly match the capitalization of the input text. The observation here is that a string like "dark" appearing in biomedical text is most often being used as a normal English word, while a string like "DARK", is likely being used as a gene name.

Finally, the algorithm examines all matching terms on the input text. Overlaps are resolved with a combination of criteria based on comparing the confidence and length of each recognized entity. In the current implementation, the confidence of the dictionary-based NER is always 1.0, so in practice the system resolves overlap by keeping the entity recognized by the longest overlapping term and discarding any shorter overlapping entities.

2.6 Disambiguation

It has been shown that a large number of gene and protein terms refer to more than one actual concept with over 5% of terms being ambiguous intra-species and 85% being ambiguous with gene names for other organisms (Tuason *et al.*, 2004, Chen *et al.*, 2005). For normalization, occurrences of these ambiguous terms need to be resolved to the correct concept. This is called *disambiguation*.

Various disambiguation approaches have been proposed, including the method of Hanisch previously described, as well as simply ignoring ambiguous terms. Ignoring all ambiguous terms can be wasteful, since context may allow disambiguation to a unique concept. This can helpful for increasing the sample size for further text mining. For example NER and normalization can be performed on abstracts, and further processing (e.g., co-occurrence detection) performed at the sentence level. Our approach to disambiguation makes two assumptions about the biomedical literature. First, ambiguous terms are often synonyms for other, non-ambiguous terms within the same text sample, and second, authors usually explicitly provide sufficient context for readers to resolve ambiguous terms.

For each ambiguous term, we collect the potential normalized concepts. If any of those concepts appears in the text sample using an unambiguous term for that concept, we assign the ambiguous term to the concept with the unambiguous term. If there is more than one concept with an unambiguous term (this occurs infrequently), we select one of these concepts at random. We ignore terms that cannot be resolved in this manner. Notice that this is a general dictionary disambiguation algorithm and does not require any information specific to genes and proteins.

2.7 Optimization

One of the benefits of the dictionary-based approach is that it is simple and amenable to code optimization. In our case we were able to gain almost a thousand-fold speed improvement over brute force searching against every term in the database. We accomplished this using an approach based on indexing the term prefixes, taking each unique sequence of *n* initial term characters as the index for all terms with that initial sequence. In our system we chose an *n* of 6 as a good balance between performance and memory requirements.

Searching for gene and protein terms then becomes an efficient matter of only searching for the terms that correspond to 6 character sequences (prefixed by a delimiter) that actually exist in the input text. This greatly reduces the number of searching operations necessary. While other more complex optimization algorithms are possible, such as organizing the terms character-by-character into an n-way tree, or completely grouping the terms into a complete prefix tree, our approach is simple, very fast, and has modest memory needs.

3 Evaluation

We based our evaluation on two test corpora that have been previous used to evaluate gene and protein NER and normalization. We used the GENIA corpus, version 3.02 (Kim *et al.*, 2003), to evaluate the utility of each online database as a source of terms for gene and protein NER, and we used the BioCreative Task1B

mouse and yeast collections to evaluate the performance of our system for normalized gene and protein identification.

The GENIA corpus is a key resource in biomedical text mining, and has been used by many investigators (e.g., (Collier and Takeuchi, 2004, Lee *et al.*, 2004, Tsuruoka and Tsujii, 2004)). However, some system-dependent decisions still need to be made in order to use it as a gold standard for gene and protein NER. First, GENIA marks genes separately from proteins. While the "protein_molecule" attribute appears to be used in a manner that tightly and specifically delimits mentions of proteins, other attributes such as the "DNA_domain_or_region" attribute and the "protein_family_or_group" attribute are used more loosely. "DNA_domain_or_region" can be used to mark a specific gene (e.g., "IL-2 gene", "peri kappa-B site"), sometimes including words such as "gene" and "site". At other times the attribute marks a non-specific gene concept (e.g., "viral gene"). Similar observations are true about the "protein_family_or_group" attributes (e.g., "CD28", "transcription factor"). Clearly when evaluating dictionary-based (possibly as opposed to corpus trained) gene/protein NER, many of the concepts marked with the "DNA_domain_or_region", "protein_family_or_group" and other similar attributes should be treated as correct for the purposes of precision. However, the large number of more generic concepts that these attributes mark should not be included in the calculation of recall.

Because of these issues, here we have used a hybrid technique in order to produce the most meaningful results in choosing a database for wide coverage of gene and protein names and symbols. Entities marked with the "protein_molecule" attribute are included for computation of both precision and recall. The text marked with the DNA and protein family attributes are only used for the computation of precision. This method is different from that applied by others using the GENIA corpus for both training and testing and therefore our NER results here are not directly comparable to prior work using GENIA.

In the first set of experiments we are primarily concerned with evaluating the richness of each database and combination of databases as a source of names for gene and protein NER. Therefore, we use the weak match criteria of Chang et al., to evaluate performance (Chang *et al.*, 2004). The weak match criteria treats any overlap of identified text with the gold standard as a positive.

In the second set of experiments we use the BioCreative mouse and yeast test collections to evaluate the performance of our unsupervised dictionary-based method of gene and protein NER and normalization. For mouse, the more challenging organism, we evaluate the effect of each system feature separately and in combination. We also evaluate the effect of using just the organism-specific database to populate the dictionary, along with the organism-specific database in combination with the richest database determined in the first set of experiments. Table 1 shows information on the databases that were used to generate the dictionaries and the fields taken from each database.

4 Results

Table 2 presents the results of applying our dictionary-based NER to the GENIA 3.02 corpus using the three multi-organism databases individually. The Entrez Gene database performs the best, having both the highest F-measure of 75.5% at a precision of 73.5% and a recall of 77.6%. The LocusLink database is next, and not significantly different in performance (LocusLink is being phased out and replaced with Entrez Gene as of March 2005). The UniProt database performs much worse overall. This is surprising, performing well on precision at 78.5%, but having recall of 59.1%, poorer than we expected for a multi-species database.

Table 1. Databases used to create protein/gene NER dictionaries. Fields marked with an asterisk were used as the unique identifier.

Database & Organism	Fields used	Dictionary Size
Entrez multi-organism	SYMBOL*, SYNONYMS, DESCRIPTION	59 Mbytes
LocusLink multi-organism	PRODUCT, OFFICIAL_SYMBOL*, PREFERRED_SYMBOL, OFFICIAL_GENE_NAME, PREFERRED_GENE_NAME, PREFERRED_PRODUCT, ALIAS_SYMBOL, ALIAS_PROT	14 Mbytes
MGI mouse only	MGI MARKER ACCESSION ID*, MGI GENE TERM, STATUS	7 Mbytes
UniProt multi-organism	Name*, Synonyms, OrderedLocusNames, ORFNames	5 MBytes
Saccharomyces yeast only	Locus, ORF, SGID*, alias, standard name, feature name	1.5 MBytes

Table 2. Results of creating dictionary from a single database for NER of GENIA genes and proteins.

Dictionary	Precision	Recall	F-measure
Entrez	0.735	0.776	0.755
LocusLink	0.723	0.773	0.747
UniProt	0.785	0.474	0.591

Table 3. Results of creating dictionary from a combination of two databases for NER of GENIA genes and proteins.

Dictionaries	Precision	Recall	F-measure
Entrez	0.735	0.776	0.755
Entrez+UniProt	0.707	0.792	0.747
Entrez+LocusLink	0.734	0.780	0.756

Table 4. Results of using dictionary created from databases for NER and normalization for mouse.

Dictionary	Precision	Recall	F-measure
Entrez/MGI	0.775	0.726	0.750
MGI	0.710	0.535	0.610

Having found that Entrez Gene was the single best online database for dictionary creation, we tried combining it with the other databases. As can be seen from Table 3, this did not result in any meaningful performance improvement.

For the remainder of our experiments we used the BioCreative mouse and yeast test collections and gold standard files to evaluate the performance of our system for gene/protein NER and normalization. The gold standard required the unique identifiers to be MGI or SGD accession numbers. To accomplish this, we performed a join between the Entrez database and the MGI (or Saccharomyces) database using a mapping identifier between the MGI (or SGI) database entries and the Entrez Gene ids while extracting dictionary terms.

Table 4 shows the results of using the joined Entrez/MGI dictionary for mouse NER and normalization compared to using the dictionary created from the MGI database alone. Using the MGI database alone has much worse recall than using the dictionary created with a combination of Entrez and MGI databases, with recall falling almost 20%. Restricting the dictionary to the MGI database also results in a 6.5% decrease in precision.

Table 5 shows the results of individually removing each of the three main dictionary pre-processing features and the disambiguation algorithm and evaluating the NER and normalization performance for mouse. All four of these variations perform worse than our full system. Variant generation made the smallest difference, giving an F-measure improvement of 2.0%. Ambiguity resolution improves the F-measure 2.8%. The 300 most common word stop list contributed an improvement of 6.8%. Lastly, separation into case-sensitive and case-insensitive dictionaries made the largest improvement of 15.6%. Removing all of the pre-processing features at once and using the combined Entrez/MGI database as a "raw" term list performs very badly, with good recall but a precision of only 30.1%.

Table 6 compares the results of our system to the participants of BioCreative Task 1B for the mouse and yeast corpora. On both mouse and yeast, our system performs above the median F-measure. On mouse the difference in F-measure between our system and the top scoring system is less than 5%. On the yeast corpus, our approach has among the highest precision, with recall slightly below the median, and F-measure about 3% below the highest scoring system.

While ambiguity resolution resulted in a modest improvement, we wanted to get an idea of the magnitude of the ambiguity within our automatically created dictionaries. Table 7 shows the number and percentage of ambiguous terms and genes with at least one ambiguous term in the dictionaries that we created using Entrez in combination with the MGI database, as well as MGI alone.

The system runs very rapidly. On a 1.7GHz Pentium 4m laptop with 512M RAM, the 18,000 sentences in the GENIA corpus were processed in about 30 seconds. The 250 abstracts in the BioCreative corpora were processed in less than 5 seconds.

5 Discussion

The Entrez Gene database was identified as the best general-purpose source of gene and protein terms for use in a dictionary-based NER and normalization. Including data from other databases did not improve NER performance. It appears that the producers of Entrez Gene are doing an excellent job in finding and curating this information from the available sources. One of the most common difficulties cited in recognizing gene and protein names is that the vocabulary of terms is continuously expanding (Hirschman *et al.*, 2002). Online databases, such as Entrez Gene provide a curated central repository for these terms, making the task of keeping gene/protein NER and normalization systems up to date on new genes and proteins somewhat easier.

All three of our dictionary pre-processing enhancements improved performance, as did the ambiguity resolution algorithm. Surprisingly, variant generation made the smallest difference in F-measure. This may be due to the tendency for genes to be mentioned multiple times within an abstract, or that authors are keeping to the forms collected in the genomics databases, or that the database curators are doing a good job in keeping up with the terms used by authors. The BioCreative test collection scores normalization at the level of an entire abstract. It is possible that variant generation might have made a larger difference if the test collection was scored at a sentence level. On the other hand, it may be that the

Entrez database itself contains sufficient variants. In either case, the small improvement gained from variant generation suggests that computationally expensive approximate string matching techniques may not be worth the effort.

The next largest improvement was made by ambiguity resolution. Precision increased almost 8%, while recall dropped only about 2%. While an F-measure improvement of 2.8% is small, this figure is highly dependent upon the make up of the test corpus.

Certainly, as seen in Table 7, there are a large proportion of mouse genes with ambiguous terms in our dictionary. How often these ambiguous terms actually appear in the literature is an open question. Additional and larger test collections may be necessary to accurately measure the overall importance of ambiguity resolution.

Table 5. NER and normalization performance results when removing dictionary pre-processing features and ambiguity resolution for mouse.

System	Precision	Recall	F-measure	Difference
full system	0.775	0.726	0.750	-
- case	0.493	0.746	0.594	-15.6%
- stop	0.643	0.726	0.682	-6.8%
- variant	0.771	0.693	0.730	-2.0%
- ambiguity	0.697	0.748	0.722	-2.8%
- all	0.301	0.713	0.423	-32.7%

Table 6. Comparison with results from BioCreative on mouse and yeast corpora.

Organism	System	Precision	Recall	F-measure
Mouse	biocreative-highest	0.765	0.819	0.791
	cohen	0.775	0.726	0.750
	biocreative-median	0.765	0.730	0.738
	biocreative-lowest	0.418	0.898	0.571
Yeast	biocreative-highest	0.950	0.894	0.921
	cohen	0.950	0.837	0.890
	biocreative-median	0.940	0.848	0.858
	biocreative-lowest	0.661	0.902	0.763

Table 7. Term ambiguity measurements for mouse genes.

	Entrez/MGI	MGI
# All Distinct Genes	57185	57180
# All Distinct Terms	336353	250435
# Ambiguous Terms	6585 (1.96%)	2104 (0.84%)
# Genes w/ Ambiguous Terms	8036 (14.05%)	2619 (4.58%)

The stop-list made the next largest improvement in F-measure, 6.8%. Use of the stop list improved precision greatly and did not change recall. Case-sensitivity using the common word file made the largest improvement of 15.6%. While making a large, almost 30% difference in precision, case sensitivity decreased recall by only 2%.

Overall, all three of the dictionary pre-processing methods we applied worked well, as did ambiguity resolution. Each method resulted in improvement in either precision or recall, and did not greatly degrade the other measure. Together the three techniques gave an F-measure improvement of over 30% as compared to using a plain unprocessed dictionary.

Chen et al. investigated the ambiguity of official gene names within and across organisms and found the level of shared official names within an organism to be low (~0.02%) but the level of ambiguity when considering all terms associated with a gene to be higher, about 5% (Chen *et al.*, 2005). Our results are similar, with about 5% of genes in the MGI database having terms also associated with other genes. This rises to 14% when combined with the information in the Entrez database. As previously noted, inter-organism ambiguity is much higher. Further work is needed to determine the extent of the problem present within in the actual literature.

We did not apply our method to fly, the other organism in the BioCreative Task 1B test collection. We were unable to find direct mappings between identifiers in the fly database and Entrez Gene. Moreover, the fly corpus would present special problems for our method. Unlike for mouse and yeast, the fly genome contains many genes that have the same names as common English words, and the use of these words as gene names are not commonly delineated using capitalization as they are with mouse. For fly at least, methods such as ours are at a disadvantage compared to trained systems.

However, the literature of one of the most important and interesting genomes (at least to us), human, does appear to follow the practice of differentiating common English words from gene and protein names by uppercase or initial capitalization similar to the mouse literature (Chen *et al.*, 2005). Therefore we expect that our unsupervised approach will be useful for human genomics literature as well.

Unfortunately at the present time we are unable to test this hypothesis. We are unaware of any human gene NER and normalization test collection. While there are several test collections widely available for NER alone (Franzen *et al.*, 2002, Kim *et al.*, 2003, Hu *et al.*, 2004), the same cannot be said for the essential normalization step. More and larger collections, covering additional organisms such as human and rat, are necessary to measure and motivate progress in gene and protein NER and normalization.

6 Conclusions and Future Work

These results demonstrate that an unsupervised dictionary-based approach to gene and protein NER and normalization can be effective. The dictionaries can be created automatically without human intervention or review. Dictionary-based systems such as ours can be set up to automatically update themselves by downloading the database files on the Internet and pre-processing the files into updated dictionaries. This could be done on a nightly basis if necessary, since the entire dictionary creation process only takes a few minutes. One general database, combined with an organism-specific database for each species, is sufficient.

Our work is distinguished from other dictionary-based work such as Tsurukoka and Tsujii, and Hanisch et al. in several ways. Unlike both of these prior investigators, we use on-line curated information as our primary source of terms, instead of deriving them from a training set, and have shown both which databases to use and how to process them into effective sources of terms for NER. Our textual variants are generated by simple rules determined by domain knowledge instead of machine learning on training data. Lastly, the disambiguation algorithm presented here is unique and has been shown to have a positive impact on performance.

The system is as accurate as other more complex approaches. It does not require training, and so may be less sensitive to specific characteristics of a given text corpus. It may also be applied to organisms for which there do not exist sufficient training and test collections. In addition, the system is very fast. This may enable some text mining tasks to be done for users in real time, rather than the batch processing mode that is currently most common in biomedical text mining research.

Dictionary-based approaches are likely to remain an essential part of gene and protein normalization, even if the NER step is handled by other methods. Further work is necessary to determine the best manner to combine automatically created dictionaries with trained NER systems. It may be the case that different approaches work best for different organisms, depending upon the specific naming conventions of scientists working on that species.

References

Brill, E. (1992) A simple rule-based part of spech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*.

Carroll, J. B., Davies, P. and Richman, B. (1971) The American heritage word frequency book. Houghton Mifflin, Boston,.

Chang, J. T., Schutze, H. and Altman, R. B. (2004) GAPSCORE: finding gene and protein names one word at a time. *Bioinformatics*, **20**, 216-25.

Chen, L., Liu, H. and Friedman, C. (2005) Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, **21**, 248-56.

Cherry, J. M. (1995) Genetic nomenclature guide. Saccharomyces cerevisiae. *Trends Genet*, 11-2.

Cohen, A. M. and Hersh, W. (2005) A Survey of Current Work in Biomedical Text Mining. *Briefings in Bioinformatics*, **6**, 57-71.

Cohen, A. M., Hersh, W. R., Dubay, C. and Spackman, K. (2005) Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts. *BMC Bioinformatics*, **6**,

Collier, N. and Takeuchi, K. (2004) Comparison of character-level and part of speech features for name recognition in biomedical texts. *J Biomed Inform*, **37**, 423-35.

Franzen, K., Eriksson, G., Olsson, F., Asker, L., Liden, P. and Coster, J. (2002) Protein names and how to find them. *Int J Med Inf*, **67**, 49-61.

Hanisch, D., Fundel, K., Mevissen, H. T., Zimmer, R. and Fluck, J. (2004) ProMiner: Organism-specific protein name detection using approximate string matching. In *BioCreative: Critical Assessment for Information Extraction in Biology*.

Hirschman, L., Morgan, A. A. and Yeh, A. S. (2002) Rutabaga by any other name: extracting biological names. *J Biomed Inform*, **35**, 247-59.

Hirschman, L., Colosimo, M., Morgan, A., Columbe, J. and Yeh, A. (2004) Task 1B: Gene List Task BioCreAtIve Workshop. In *BioCreative: Critical Assessment for Information Extraction in Biology*.

Hu, Z. Z., Mani, I., Hermoso, V., Liu, H. and Wu, C. H. (2004) iProLINK: an integrated protein resource for literature mining. *Comput Biol Chem*, **28**, 409-16.

Kim, J. D., Ohta, T., Tateisi, Y. and Tsujii, J. (2003) GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, **19**, i180-i182.

Krallinger, M. (2004) BioCreAtIvE - Critical Assessment of Information Extraction systems in Biology. http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html

Lee, K. J., Hwang, Y. S., Kim, S. and Rim, H. C. (2004) Biomedical named entity recognition using two-phase model based on SVMs. *J Biomed Inform*, **37**, 436-47.

Tanabe, L. and Wilbur, W. J. (2002) Tagging gene and protein names in biomedical text. *Bioinformatics*, **18**, 1124-32.

Tsuruoka, Y. and Tsujii, J. (2004) Improving the performance of dictionary-based approaches in protein name recognition. *J Biomed Inform*, **37**, 461-70.

Tuason, O., Chen, L., Liu, H., Blake, J. A. and Friedman, C. (2004) Biological nomenclatures: a source of lexical knowledge and ambiguity. *Pac Symp Biocomput*, 238-49.

Ward, G. (2000) Grady Ward's Moby. http://www.dcs.shef.ac.uk/research/ilash/Moby/mwords.html

Yu, H. and Agichtein, E. (2003) Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, **19**, i340-i349.

Zhou, G., Zhang, J., Su, J., Shen, D. and Tan, C. (2004) Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, **20**, 1178-90.

A Machine Learning Approach to Acronym Generation

Yoshimasa Tsuruoka[†‡]
[†]CREST
Japan Science and Technology Agency
Japan

Sophia Ananiadou
School of Computing
Salford University
United Kingdom

Jun'ichi Tsujii[‡†]
[‡]Department of Computer Science
The University of Tokyo
Japan

tsuruoka@is.s.u-tokyo.ac.jp
S.Ananiadou@salford.ac.uk
tsujii@is.s.u-tokyo.ac.jp

Abstract

This paper presents a machine learning approach to acronym generation. We formalize the generation process as a sequence labeling problem on the letters in the definition (expanded form) so that a variety of Markov modeling approaches can be applied to this task. To construct the data for training and testing, we extracted acronym-definition pairs from MEDLINE abstracts and manually annotated each pair with positional information about the letters in the acronym. We have built an MEMM-based tagger using this training data set and evaluated the performance of acronym generation. Experimental results show that our machine learning method gives significantly better performance than that achieved by the standard heuristic rule for acronym generation and enables us to obtain multiple candidate acronyms together with their likelihoods represented in probability values.

1 Introduction

Technical terms and named-entities play important roles in knowledge integration and information retrieval in the biomedical domain. However, spelling variations make it difficult to identify the terms conveying the same concept because they are written in different manners. Acronyms constitute a major part of spelling variations (Nenadic et al., 2002), so proper management of acronyms leads to improved performance of the information systems in this domain.

As for the methods for recognizing acronym-definition pairs from running text, there are many studies reporting high performance (e.g. over 96% accuracy and 82% recall) (Yoshida et al., 2000; Nenadic et al., 2002; Schwartz and Hearst, 2003; Zahariev, 2003; Adar, 2004). However, another aspect that we have to consider for efficient acronym management is to generate acronyms from the given definition (expanded form).

One obvious application of acronym generation is to expand the keywords in information retrieval. As reported in (Wren et al., 2005), for example, you can retrieve only 25% of the documents concerning the concept of "JNK" by using the keyword "c-jun N-terminal kinase". In more than 33% of the documents the concept is written with its acronym "JNK". To alleviate this problem, some research efforts have been devoted to constructing a database containing a large number of acronym-definition pairs from running text of biomedical documents (Adar, 2004).

However, the major problem of this database-building approach is that building the database offering complete coverage is nearly impossible because not all the biomedical documents are publicly available. Although most of the abstracts of biomedical papers are publicly available on MEDLINE, there is still a large number of full-papers which are not available.

In this paper, we propose an alternative approach

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 25–31, Detroit, June 2005.

to providing acronyms from their definitions so that we can obtain acronyms without consulting acronym-definition databases.

One of the simplest way to generate acronyms from definitions would be to choose the letters at the beginning of each word and capitalize them. However, there are a lot of exceptions in the acronyms appearing in biomedical documents. The followings are some real examples of the definition-acronym pairs that cannot be created with the simple heuristic method.

RNA polymerase (RNAP)
antithrombin (AT)
melanoma cell adhesion molecule (Mel-CAM)
the xenoestrogen 4-tert-octylphenol (t-OP)

In this paper we present a machine learning approach to automatic generation of acronyms in order to capture a variety of mechanisms of acronym generation. We formalize this problem as a sequence labeling task such as part-of-speech tagging, chunking and other natural language tagging tasks so that common Markov modeling approaches can be applied to this task.

2 Acronym Generation as a Sequence Labeling Problem

Given the definition (expanded form), the mechanism of acronym generation can be regarded as the task of selecting the appropriate action on each letter in the definition.

Figure 1 illustrates an example, where the definition is "Duck interferon gamma" and the generated acronym is "DuIFN-gamma". The generation proceeds as follows:

> The acronym generator outputs the first two letters unchanged and skips the following three letters. Then the generator capitalizes 'i' and skip the following four letters...

By assuming that an acronym is made up of alphanumeric letters, spaces and hyphens, the actions being taken by the generator are classified into the following five classes.

- SKIP

  The generator skips the letter.

- UPPER

  If the target letter is uppercase, the generator outputs the same letter. If the target letter is lowercase, the generator coverts the letter into the corresponding upper letter.

- LOWER

  If the target letter is lowercase, the generator outputs the same letter. If the target letter is uppercase, the generator coverts the letter into the corresponding lowercase letter.

- SPACE

  The generator convert the letter into a space.

- HYPHEN

  The generator convert the letter into a hyphen.

From the probabilistic modeling point of view, this task is to find the sequence of actions $t_1...t_n$ that maximizes the following probability given the observation $o = o_1...o_n$

$$P(t_1...t_n|o). \tag{1}$$

Observations are the letters in the definition and various types of features derived from them. We decompose the probability in a left-to-right manner.

$$P(t_1...t_n|o) = \prod_{i=1}^{n} p(t_i|t_1...t_{i-1}o). \tag{2}$$

By making a first-order markov assumption, the equation becomes

$$P(t_1...t_n|o) = \prod_{i=1}^{n} p(t_i|t_{i-1}o). \tag{3}$$

If we have the training data containing a large number of definition-acronym pairs where the definition is annotated with the labels for actions, we can estimate the parameters of this probabilistic model and the best action sequence can be efficiently computed by using a Viterbi decoding algorithm.

In this paper we adopt a maximum entropy model (Berger et al., 1996) to estimate the local probabilities $p(t_i|t_{i-1}o)$ since it can incorporate diverse types of features with reasonable computational cost. This modeling, as a whole, is called Maximum Entropy Markov Modeling (MEMM).

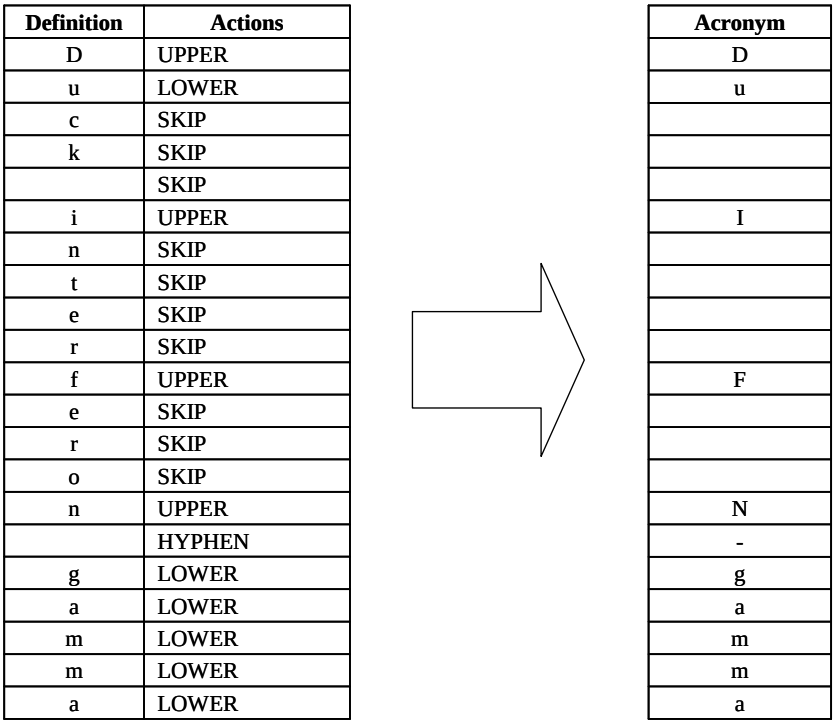


Figure 1: Acronym generation as a sequence labeling problem. The definition is "Duck interferon gamma" and the acronym is "DuIFN-gamma". Each letter in the acronym is generated from a letter in the definition following the action for the letter.

Regularization is important in maximum entropy modeling to avoid overfitting to the training data. For this purpose, we use the maximum entropy modeling with inequality constraints (Kazama and Tsujii, 2003). The model gives equally good performance as the maximum entropy modeling with Gaussian priors (Chen and Rosenfeld, 1999), and the size of the resulting model is much smaller than that of Gaussian priors because most of the parameters become zero. This characteristic enables us to easily handle the model data and carry out quick decoding, which is convenient when we repetitively perform experiments. This modeling has one parameter to tune, which is called *width factor*. We set this parameter to be 1.0 throughout the experiments.

3 The Data for Training and Testing

Since there is no training data available for the machine learning task described in the previous section, we manually created the data. First, we extracted definition-acronym pairs from MEDLINE abstracts using the acronym acquisition method proposed by (Schwartz and Hearst, 2003). The abstracts used for constructing the data were randomly selected from the abstracts published in the year of 2001. Duplicated pairs were removed from the set.

In acquiring the pairs from the documents, we focused only on the pairs that appear in the form of

... *expanded_form* (*acronym*) ...

We then manually removed misrecognized pairs and annotated each pair with positional information. The positional information tells which letter in the definition should correspond to a letter in the acronym. Table 1 lists a portion of the data. For example, the positional information in the first pair indicates that the first letter 'i' in the definition corresponds to 'I' in the acronym, and the 12th letter 'm' corresponds to 'M'.

With this positional information, we can create the training data for the sequence labeling task because there is one-to-one correspondence between the sequence labels and the data with positional information. In other words, we can determine the ap-

Definition	Acronym	Positional Information
intestinal metaplasia	IM	1, 12
lactate dehydrogenase	LDH	1, 9, 11
cytokeratin	CK	1, 5
cytokeratins	CKs	1, 5, 12
Epstein-Barr virus	EBV	1, 9, 14
30-base pairs	bp	4, 9
in-situ hybridization	ISH	1, 4, 9
:	:	:

Table 1: Curated data containing definitions, their acronyms and the positional information.

propriate action for each letter in the definition by comparing the letter with the corresponding letter in the acronym.

4 Features

Maximum entropy modeling allows us to incorporate diverse types of features. In this paper we use the following types of features in local classification. As an example, consider the situation where we are going to determine the action at the letter 'f' in the definition "Duck interferon gamma".

- Letter unigram (UNI)

  The unigrams of the neighboring letters. (i.e. "uni_{-1} r", "uni_0 f", and "uni_{+1} e")

- Letter bigram (BI)

  The bigrams of the neighboring letters. (i.e. "bi_{-2} er", "bi_{-1} rf", "bi_0 fe", and "bi_{+1} er")

- Letter trigram (TRI)

  The trigrams of the neighboring letters. (i.e. "tri_{-2} ter", "tri_{-1} erf", "tri_0 rfe", "tri_{+1} fer", and "tri_{+2} ero")

- Action history (HIS)

  The preceding action (i.e. SKIP)

- Orthographic features (ORT)

  Whether the target letter is uppercase or not (i.e. false)

- Definition Length (LEN)

  The number of the words in the definition (i.e. "len=3")

- Letter sequence (SEQ)

  1. The sequence of the letters ranging from the beginning of the word to the target letter. (i.e. "seq_{left} interf")
  2. The sequence of the letters ranging from the target letter to the end of the word. (i.e. "seq_{right} feron")
  3. The word containing the target letter. (i.e. "seq_{word} interferon")

- Distance (DIS)

  1. The distance between the target letter and the beginning of the word. (i.e. "dis_{left} 6")
  2. The distance between the target letter and the tail of the word. (i.e. "dis_{right} 5")

Rank	Probability	String
1	**0.779**	**TBI**
2	0.062	TUBI
3	0.028	TB
4	0.019	TbI
5	0.015	TB-I
6	0.009	tBI
7	0.008	TI
8	0.007	TBi
9	0.002	TUB
10	0.002	TUbI
ANSWER		TBI

Table 2: Generated acronyms for "traumatic brain injury".

5 Experiments

To evaluate the performance of the acronym generation method presented in the previous section, we ran five-fold cross validation experiments using the manually curated data set. The data set consists of 1,901 definition-acronym pairs.

For comparison, we also tested the performance of the popular heuristics for acronym generation in which we choose the letters at the beginning of each word in the definition and capitalize them.

Rank	Probability	String
1	0.423	ORF1
2	0.096	OR1
3	**0.085**	**ORF-1**
4	0.070	RF1
5	0.047	OrF1
6	0.036	OF1
7	0.025	ORf1
8	0.019	OR-1
9	0.016	R1
10	0.014	RF-1
ANSWER		ORF-1

Table 3: Generated acronyms for "open reading frame 1".

Rank	Probability	String
1	0.163	RNA-P
2	0.147	RP
3	0.118	RNP
4	**0.110**	**RNAP**
5	0.064	RA-P
6	0.051	R-P
7	0.043	RAP
8	0.041	RN-P
9	0.034	RNA-PM
10	0.030	RPM
ANSWER		RNAP

Table 4: Generated acronyms for "RNA polymerase".

5.1 Generated Acronyms

Tables 2 to 5 show some examples of generated acronyms together with their probabilities. They are sorted with their probabilities and the top ten acronyms are shown. The correct acronym given in the training data is described in the bottom row in each table.

In Table 2, the definition is "traumatic brain injury" and the correct acronym is "TBI". This is the simplest case in acronym generation, where the first letter of each word in the definition is to be capitalized. Our acronym generator gives a high probability to the correct acronym and it is ranked at the top.

Table 3 shows a slightly more complex case, where the generator needs to convert the space be-

Rank	Probability	String
1	0.405	MCPP
2	0.149	MCP
3	0.056	MCP
4	0.031	MPP
5	0.028	McPP
6	0.024	MchPP
7	0.020	MC
8	0.011	MP
9	**0.011**	**mCPP**
10	0.010	MCRPP
ANSWER		mCPP

Table 5: Generated acronyms for "meta-chlorophenylpiperazine".

Rank	Probability	String
1	0.811	TV
2	0.034	TSV
3	0.030	TCV
4	0.021	Tv
5	0.019	TVs
6	0.013	T-V
7	0.008	TOV
8	0.004	TSCV
9	0.002	T-v
10	**0.001**	**TOSV**
ANSWER		TOSV

Table 6: Generated acronyms for "Toscana virus".

Rank	Coverage (%)
1	55.2
2	65.8
3	70.4
4	73.2
5	75.4
6	76.7
7	78.3
8	79.8
9	81.1
10	82.2
BASELINE	47.3

Table 7: Coverage achieved with the Top N Candidates.

tween ‘F’ and ‘1’ into a hyphen. The correct answer is located at the third rank.

The definition in Table 4 is “RNA polymerase” and the correct acronym is “RNAP”, so the generator needs to the first three letters unchanged. The correct answer is located at the fourth rank, and the probability given the correct answer does not have a large gap with the top-ranked acronym.

Table 5 shows a more difficult case, where you need to output the first letter in lowercase and choose appropriate letters from the string having no delimiters (e.g. spaces and hyphens). Our acronym generator outputs the correct acronym at the nine-th rank but the probability given this acronym is very low compared to that given to the top-ranked string.

Table 6 shows a similar case. The probability given to the correct acronym is very low.

5.2 Coverage

Table 7 shows how much percentage of the correct acronyms are covered if we take top N candidates from the outputs of the acronym generator. The bottom line (BASELINE) shows the coverage achieved by generating one acronym using the standard heuristic rule for acronym generation. Note that the coverage achieved with a single candidate (Rank 1) is better that of BASELINE.

If we take top five candidates, we can have a coverage of 75.4%, which is considerably better than that achieved by the heuristic rule. This suggests that the acronym generator could be used to significantly improve the performance of the systems for information retrieval and information integration.

5.3 Features

To evaluate how much individual types of features affect the generation performance, we ran experiments using different feature types. Table 8 shows the results. Overall, the results show that various types of features have been successfully incorporated in the MEMM modeling and individual types of features contribute to improving performance.

The performance achieved with only unigram features is almost the same as that achieved by the heuristic rule. Note that the features on the previous state improve the performance, which suggests that our selection of the states in the Markov modeling is a reasonable choice for this task.

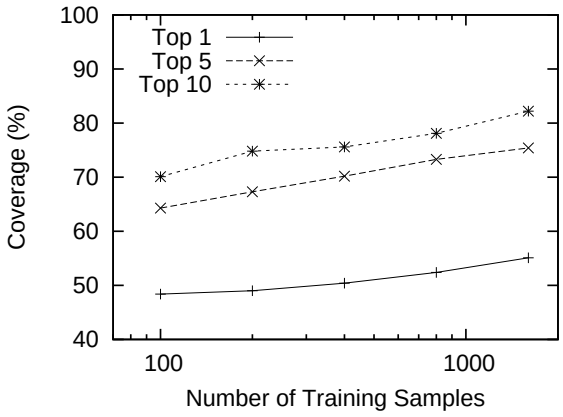


Figure 2: Learning curve.

5.4 Learning Curve

Figure 2 shows a learning curve of our acronym generator, which shows the relationship between the number of the training samples and the performance of the system. The graph clearly indicates that the performance consistently improves as the training data increases and still continues to improve even when the size of the training data reaches the maximum. This suggests that we can achieve improved performance by increasing the annotated data for training.

6 Conclusion

We presented a machine learning approach to acronym generation. In this approach, we regarded the generation process as a sequence labeling problem, and we manually created the data for training and testing.

Experimental results using 1901 definition-acronym pairs, we achieved a coverage of 55.1%, which is significantly bettern than that achieved by the standard heuristic rule for acronym generation. The algorithm also enables us to have other acronym candidates together with the probabilities representing their likelihood.

6.1 Future work

In this paper we did not consider the generation mechanisms where the letters in the acronym appear in a different order in the definition. Since about 3% of acronyms reportedly involve this types of generation mechanism (Schwartz and Hearst, 2003), we

Feature Templates	Top 1 Coverage (%)	Top 5 Coverage (%)	Top 10 Coverage (%)
UNI	48.2	66.2	74.2
UNI, BI	50.1	71.2	78.3
UNI, BI, TRI	50.4	72.3	80.1
UNI, BI, TRI, HIS	50.6	73.6	81.2
UNI, BI, TRI, HIS, ORT	51.0	73.9	80.9
UNI, BI, TRI, HIS, ORT, LEN	53.9	74.6	81.3
UNI, BI, TRI, HIS, ORT, LEN, DIS	54.4	75.0	81.8
UNI, BI, TRI, HIS, ORT, LEN, DIS, SEQ	55.1	75.4	82.2

Table 8: Performance with Different Feature Sets.

might further improve performance by considering such permutation of letters.

As the learning curve (Fig 2) suggested, one obvious way to improve the performance is to increase the training data. The size of the training data used in the experiments is fairly small compared to those in other sequence tagging tasks such POS tagging and chunking. We plan to increase the size of the training data with a semi-automatic way that could reduce the human effort for annotation.

References

Eytan Adar. 2004. Sarad: A simple and robust abbreviation dictionary. *Bioinformatics*, 20(4):527–533.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. *Technical Report CMUCS -99-108, Carnegie Mellon University*.

Jun'ichi Kazama and Jun'ichi Tsujii. 2003. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of EMNLP 2003*.

Goran Nenadic, Irena Spasic, and Sophia Ananiadou. 2002. Automatic acronym acquisition and term variation management within domain-specific texts. In *Proceedings of the LREC-3*, pages 2155–2162.

Ariel Schwartz and Marti Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical texts,. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*.

Jonathan D. Wren, Jeffrey T. Chang, James Pustejovsky, Eytan Adar, Harold R. Garner, and Russ B. Altman. 2005. Biomedical term mapping databases. *Nucleic Acid Research*, 33.

M. Yoshida, K. Fukuda, and T. Takagi. 2000. Pnad-css: a workbench for constructing a protein name abbreviation dictionary. *Bioinformatics*, 16(2):169–175.

Manuel Zahariev. 2003. An efficient methodology for acronym-expansion matching. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*.

MedTag: A Collection of Biomedical Annotations

L.H. Smith[†], **L. Tanabe**[†], **T. Rindflesch**[‡], **W.J. Wilbur**[†]
[†]National Center for Biotechnology Information
[‡]Lister Hill National Center for Biomedical Communications
NLM, NIH, 8600 Rockville Pike, Bethesda, MD 20894
{lsmith,tanabe,wilbur}@ncbi.nlm.nih.gov
rindflesch@nlm.nih.gov

Abstract

We present a database of annotated biomedical text corpora merged into a portable data structure with uniform conventions. MedTag combines three corpora, MedPost, ABGene and GENETAG, within a common relational database data model. The GENETAG corpus has been modified to reflect new definitions of genes and proteins. The MedPost corpus has been updated to include 1,000 additional sentences from the clinical medicine domain. All data have been updated with original MEDLINE text excerpts, PubMed identifiers, and tokenization independence to facilitate data accuracy, consistency and usability.

The data are available in flat files along with software to facilitate loading the data into a relational SQL database from ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedTag/medtag.tar.gz.

1 Introduction

Annotated text corpora are used in modern computational linguistics research and development to fine-tune computer algorithms for analyzing and classifying texts and textual components. Two important factors for useful text corpora are 1) accuracy and consistency of the annotations, and 2) usability of the data. We have recently updated the text corpora we use in our research with respect to these criteria.

Three different corpora were combined. The ABGene corpus consists of over 4 000 sentences annotated with gene and protein named entities. It was originally used to train the ABGene tagger to recognize gene/protein names in MEDLINE records, and recall and precision rates in the lower 70 percentile range were achieved (Tanabe and Wilbur, 2002). The MedPost corpus consists of 6 700 sentences, and is annotated with parts of speech, and gerund arguments. The MedPost tagger was trained on 3 700 of these sentences and achieved an accuracy of 97.4% on the remaining sentences (Smith et. al., 2004). The GENETAG corpus for gene/protein named entity identification, consists of 20 000 sentences and was used in the BioCreative 2004 Workshop (Yeh et. al., 2005; Tanabe et. al., 2005) (only 15 000 sentences are currently released, the remaining 5 000 are being retained for possible use in a future workshop). Training on a portion of the data, the top performing systems achieved recall and precision rates in the lower 80 percentile range. Because of the scarcity of good annotated data in the realm of biomedicine, and because good performance has been obtained using this data, we feel there is utility in presenting it to a wider audience.

All of the MedTag corpora are based on MEDLINE abstracts. However, they were queried at different times, and used different (but similar) algorithms to perform tokenization and sentence segmentation. The original annotations were assigned to tokens, or sequences of tokens, and extensively reviewed by the authors at different times for the different research projects.

The main goals in combining and updating these

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 32–37, Detroit, June 2005.

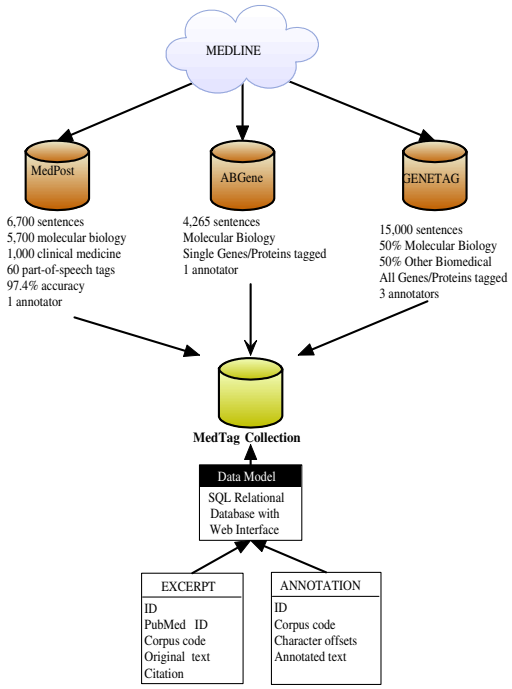


Figure 1: Component corpora, common data model and main record types of the MedTag collection.

Corpus	sentences	tokens	most frequent tag
GENETAG-05	15,000	418,246	insulin_GENE(112)
MedPost	6,700	181,626	the_DD(8,507)
AbGene	4,265	123,208	cyclin_GENE(165)

MedPost

Adj	Adv	Aux	Noun	Punct	Verb
14,648	4,553	56,262	60,732	21,806	23,625

GENETAG-05

GENE	ALTGENE
24,562	19,216

ABGene

GENE	ALTGENE
8,185	0

Table 1: MedTag Corpora. GENE = gene and protein names, ALTGENE = acceptable alternatives for gene and protein names. MedPost tagset contains 60 parts of speech which have been binned here for brevity.

corpora into a single corpus were to

1. update the text for all corpora to that currently found in MEDLINE, storing a correct citation and the original, untokenized text for each excerpt
2. eliminate tokenization dependence
3. put all text and annotations into a common database format
4. provide programs to convert from the new corpus format to the data formats used in previous research

2 Merging the Corpora

We describe what was done to merge the original corpora, locating original sources and modifying the text where needed. An overview is given in Figure 1. Some basic statistics are given in Table 1.

2.1 Identifying Source Data

The original data of the three corpora were assembled and the text was used to search MEDLINE to find the closest match. An exact or near exact match was found for all but a few excerpts. For only a few excerpts, the MEDLINE record from which the excerpt was originally taken had been removed or modified and an alternative sentence was selected. Thus, each excerpt in the database is taken from a MEDLINE record as it existed at one time in 2004. In order to preserve the reference for future work, the PubMed ID and citation data were also retrieved and stored with each excerpt. Each excerpt in the current database roughly corresponds to a sentence, although the procedure that extracted the sentence is not specified.

2.2 Eliminating Tokenization Dependence

In the original ABGene and GENETAG corpora, the gene and protein phrases were specified by the tokens contained in the phrase, and this introduced a dependence on the tokenization algorithm. This created problems for researchers who wished to use a different tokenization. To overcome this dependence, we developed an alternative way of specify-

ing phrases. Given the original text of an excerpt, the number of non-whitespace characters to the start of the phrase does not depend on the tokenization. Therefore, all annotations now refer to the first and last character of the phrase that is annotated. For example the protein *serum LH* in the excerpt

> There was no correlation between *serum LH* and chronological or bone age in this age group, which suggests that the correlation found is not due to age-related parallel phenomena.

is specified as characters 28 to 34 (the first character is 0).

2.3 Data Model

There are two main record types in the database, EXCERPT and ANNOTATION. Each EXCERPT record stores an identifier and the original corpus code (abgene, medpost, and genetag) as well as subcorpus codes that were defined in the original corpora. The original text, as it was obtained from MEDLINE, is also stored, and a human readable citation to the article containing the reference.

Each ANNOTATION record contains a reference to the excerpt (by identifier and corpus), the character offset of the first and last characters of the phrase being annotated (only non-whitespace characters are counted, starting with 0), and the corresponding annotation. The annotated text is stored for convenience, though it can be obtained from the corresponding excerpt record by counting non-whitespace characters.

The data is provided as an ASCII file in a standard format that can be read and loaded into a relational database. Each record in the file begins with a line of the form >>*table_name* where *table_name* is the name of the table for that record. Following the table name is a series of lines with the form *field: value* where *field* is the name of the field and *value* is the value stored in that field.

Scripts are provided for loading the data into a relational database, such as *mysql* or *ORACLE*. SQL queries can then be applied to retrieve excerpts and annotations satisfying any desired condition. For example, here is an SQL query to retrieve excerpts from the MedPost corpus containing the token *p53* and *signaling* or *signalling*

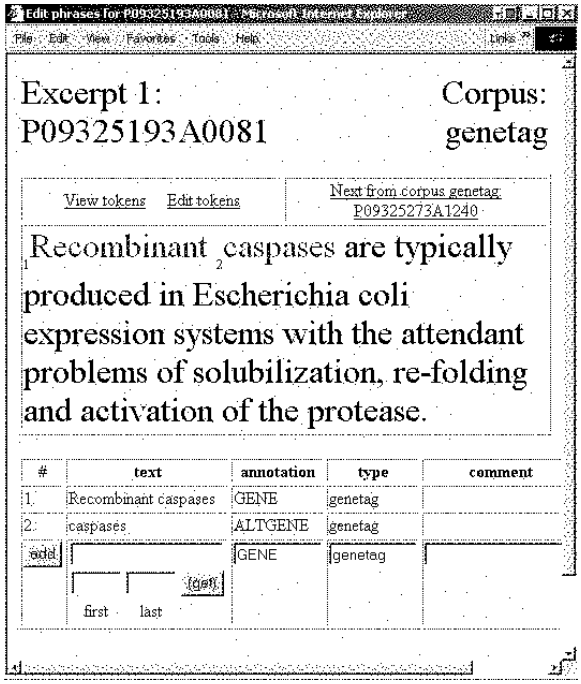


Figure 2: A screen capture of the annotator's interface and the GENETAG-05 annotations for a sentence.

```
select text from excerpt
where text like '%p53%'
and text rlike 'signa[l]*ing';
```

2.4 Web Interface

A web-based corpus editor was used to enter and review annotations. The code is being made available, as is, and requires that the data are loaded into a mysql database that can be accessed by a web server. The interface supports two annotation types: MedPost tags and arbitrary phrase annotations. MedPost tags are selectable from a pull-down menu of preprogrammed likely tags. For entering phrase annotations, the user highlights the desired phrase, and pressing the enter key computes and saves the first and last character offsets. The user can then enter the annotation code and an optional comment before saving it in the database. A screen dump of the phrase annotations for a sentence in the genetag corpus is shown in figure 2.

The data from the database was dumped to the flat file format for this release. We have also included some files to accommodate previous users of the corpora. A perl program, `alt_eval.perl` is in-

cluded that replaces the GENETAG evaluation program using non-whitespace character numbers instead of token numbers. Copies of the ABGene and MedPost corpora, in the original formats, are also included.

3 Updates of Component Corpora

3.1 MedPost Update

The MedPost corpus (Smith et. al., 2004) originally contained 5 700 tokenized sentences. An additional 1 000 annotated sentences have been added for this release. Each sentence in the MedPost corpus is fully tokenized, that is, divided into non-overlapping annotated portions, and each token is annotated with one of 60 part of speech tags (see Table 1). Minor corrections to the annotations have been made since the original release.

Since most of the original corpus, and all of the sentences used for training the MedPost tagger, were in the area of molecular biology, we added an additional 1 000 sentences selected from random MEDLINE abstracts on the subject of clinical medicine. As a preliminary result, the trained MedPost tagger achieves approximately 96.9% accuracy, which is comparable to the 97.4% accuracy achieved on the subset of 1 000 sentences selected randomly from all of MEDLINE. An example of a sentence from the clinical medicine collection is

> $\text{Evidence}_{\text{NN}}$ is_{VBZ} now_{RR} $\text{available}_{\text{JJ}}$ to_{TO} show_{VVI} a_{DD} $\text{beneficial}_{\text{JJ}}$ $\text{effect}_{\text{NN}}$ of_{II} $\text{bezafibrate}_{\text{NN}}$ on_{II} $\text{retarding}_{\text{VVGN}}$ $\text{atherosclerotic}_{\text{JJ}}$ $\text{processes}_{\text{NNS}}$ and_{CC} in_{II} $\text{reducing}_{\text{VVGN}}$ risk_{NN} of_{II} $\text{coronary}_{\text{JJ}}$ heart_{NN} $\text{disease}_{\text{NN}}$.

In addition to the token-level annotations, all of the gerunds in the MedPost corpus (these are tagged *VVGN*) were also examined and it was noted whether the gerund had an explicit subject, direct object, or adjective complement. This annotation is stored with an annotation of type *gerund.* To illustrate, the two gerunds in the previous example, *retarding* and *reducing* both have direct objects (*retarding processes* and *reducing risk*), and the gerund tag is entered as "o". The gerund annotations have been used to improve a noun phrase bracketer able to recognize gerundive phrases.

3.2 GENETAG Update

GENETAG is a corpus of MEDLINE sentences that have been annotated with gene and protein names. The closest related work is the GENIA corpus (Kim et. al., 2003). GENIA provides detailed coverage of a large number of semantic entities related to a specific subset of human molecular biology, whereas GENETAG provides gene and protein name annotations only, for a wide range of organisms and biomedical contexts (molecular biology, genetics, biochemistry, clinical medicine, etc.)

We are including a new version of GENETAG, GENETAG-05, as part of the MedTag system. GENETAG-05 differs from GENETAG in four ways: 1) the definition of a gene/protein entity has been modified, 2) significant annotation errors in GENETAG have been corrected, 3) the concept of a non-specific entity has been refined, and 4) character-based indices have been introduced to reduce tokenization problems. We believe that these changes result in a more accurate and robust corpus.

GENETAG-05 maintains a wide definition of a gene/protein entity including genes, proteins, domains, sites, sequences, and elements, but excluding plasmids and vectors. The specificity constraint requires that a gene/protein name must be included in the tagged entity. This constraint has been applied more consistently in GENETAG-05. Additionally, plain sequences like *ATTGGCCTTTAAC* are no longer tagged, embedded names are tagged (*ras*-mediated), and significantly more terms have been judged to violate the specificity constraint (*growth factor, proteases, protein kinase, ribonuclease, snoRNA, rRNA, tissue factor, tumor antigen, complement, hormone receptors, nuclear factors, etc.*).

The original GENETAG corpus contains some entities that were erroneously tagged as gene/proteins. Many of these errors have been corrected in the updated corpus. Examples include *camp-responsive elements, mu element, VDRE, melanin, dentin, myelin, auxin, BARBIE box, carotenoids,* and *cellulose.* Error analysis resulted in the updated annotation conventions given in Table 1.

Enzymes are a special class of proteins that catalyze biochemical reactions. Enzyme names have varying degrees of specificity, so the line drawn for

tagging purposes is based on online resources[1] as well as background knowledge. In general, tagged enzymes refer to more specific entities than untagged enzymes (*tyrosine kinase* vs. *protein kinase*, *ATPase* vs. *protease*). Enzymes that can refer to either DNA or RNA are tagged if the reference is specified (*DNA endonuclease* vs. *endonuclease*). Enzymes that do not require DNA/RNA distinction are tagged (*lipase* vs. *ligase*, *cyclooxygenase* vs. *methylase*). Non-specific enzymes are tagged if they clearly refer to a gene or protein, as in (1).

1) The structural gene for *hydrogenase* encodes a protein product of molecular mass 45820 Da.

Semantic constraints in GENETAG-05 are the same as those for GENETAG. To illustrate, the name in (2) requires *rabies* because *RIG* implies that the gene mentioned in this sentence refers to the *rabies immunoglobulin*, and not just any *immunoglobulin*. In (3), the word *receptor* is necessary to differentiate *IGG receptor* from *IGG*, a crucial biological distinction. In (4), the number *1* is needed to accurately describe a specific type of *tumor necrosis factor*, although *tumor necrosis factor* alone might be adequate in a different context.

2) rabies immunoglobulin (RIG)
3) IGG receptor
4) Tumor necrosis factor 1

Application of the semantic constraint can result in apparent inconsistencies in the corpus (*immunoglobulin* is sufficient on its own in some sentences in the corpus, but is insufficient in (2)). However, we believe it is important that the tagged entity retain its true meaning in the *sentence context*.

4 Recommended Uses

We have found the component corpora of MedTag to be useful for the following functions:

1) Training and evaluating part-of-speech taggers
2) Training and evaluating gene/protein named entity taggers
3) Developing and evaluating a noun phrase bracketer for PubMed phrase indexing
4) Statistical analysis of grammatical usage in medical text
5) Feature generation for machine learning

The MedPost tagger was recently ported to Java and is currently being employed in MetaMap, a program that maps natural language text into the UMLS (Aronson,A.R., 2001).

5 Conclusion

We have merged three biomedical corpora into a collection of annotations called MedTag. MedTag uses a common relational database format along with a web interface to facilitate annotation consistency. We have identified the MEDLINE excerpts for each sentence and eliminated tokenization dependence, increasing the usability of the data. In GENETAG-05, we have clarified many grey areas for annotation, providing better guidelines for tagging these cases. For users of previous versions of the component corpora, we have included programs to convert from the new standardized format to the formats used in the older versions.

References

Aronson, A. R. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc. AMIA Symp., 1721.

Kim, J.-D., Ohta, T., Tateisi, Y. and Tsujii, J. 2003. GENIA corpus: a semantically annotated corpus for biotextmining. Bioinformatics, 19: 180 - 182.

Tanabe, L and Wilbur, WJ. 2002. Tagging gene and protein names in biomedical text. Bioinformatics, 18, 1124-1132.

Tanabe L, Xie N, Thom, LH, Matten W, Wilbur, WJ: GENETAG: a tagged gene corpus for gene/protein named entity recognition. BMC Bioinformatics 2005.

Smith, L, Rindflesch, T, and Wilbur, WJ. 2004. MedPost: a part of speech tagger for biomedical text. Bioinformatics, 20(13) 2320-2321.

Yeh A, Hirschman L, Morgan A, Colosimo M: BioCreAtIvE task 1A: gene mention finding evaluation. BMC Bioinformatics 2005.

[1] http://cancerweb.ncl.ac.uk/omd/copyleft.html
http://www.onelook.com/

Entity Type	Problem	GENETAG-05 Convention	Positive Examples	Negative Examples
Protein Families	Some are named after structural motifs.	Do not tag structures alone, but tag structurally related gene and protein families.	*Zinc finger protein, bZIP transcription factor, homeobox gene, TATA binding protein*	*Zinc finger, helix-turn-helix motif, leucine zipper, homeobox, TATA box*
Domains	Name can refer to 1) the amino acid content of a sequence (*PEST*), 2) the protein that binds the sequence (*TFIIIA DNA binding domain*), 3) a homologous gene (*SH2 - Src homology domain 2*), 4) the first proteins in which the domain was discovered (*LIM, PDZ*), or 5) structural entities (*POZ, zinc finger domain*).	Tag only if the domain refers to a gene or protein. Immuno-globulin regions are tagged. (*VH* refers to the *Immuno-globulin heavy chain V region*).	*BTB domain, LIM domain, HECT domain, VH domain, SH2 domain, TFIIIA DNA binding domain, Krüppel-associated box (KRAB) domains, NF-IL6 beta leucine zipper domain*	*PEST domain, SR domain, zinc finger domain, b-Zip domain, POZ domain, GATA domain, RS domain, GAR domain*
Boxes, Response Elements and Sites	Name can refer to 1) the sequence or site itself (*TAAG*), 2) a non-protein that binds to it (*Glucocorticoid Response Element*), 3) a protein that binds to it (*Sp1*), or 4) to homologous genes (*VL30*).	Tag only if the sequence or site refers to a gene or protein.	*VL30 element, Zta response elements, activating protein 1 (AP-1) site, Ets binding site, SP1 site, AP-2 box*	*GRE, TRE, cyclic AMP response element (CRE), TAAG sites, TGn motif, TAR element, UP element*
Hormones	Some are peptide hormones.	Tag only peptide hormones.	*Insulin, Glucagon, growth hormone*	*Estrogen, Progesterone, thyroid hormone*
"and" constructs	Some conjuncts require the entire construct.	Unless both conjuncts can stand alone, tag them together.	*TCR alpha and beta, D-lactate and D-glycerate dehydrogenase*	*TCR alpha, beta, D-lactate, D-glycerate dehydrogenase*
Viral Sequences	Promoters, enhancers, repeats are distinguished by organism.	Tag only if the organism is present.	*Viral LTR, HIV long terminal repeat, SV40 promoter*	*LTR, long terminal repeat*
Sequences	Some sequences lack gene or protein names.	Tag only if a gene name is included.	*NF kappa B enhancer (TGGAAATTCC)*	*TCTTAT, TTGGGG repeats*
Embedded Names	Some names are embedded in non-gene text.	Tag only the gene part.	*P-47*-deficient, *ras*-transformed	*P-47-deficient, ras-transformed*
Transposons, Satellites	Often repetitive sequences.	Tag if specific.	*L1 element, TN44, copia retrotransposon*	*non-LTR retrotransposon*
Antibodies	Often use organism or disease name.	Tag if specific.	*anti-SF group rickettsiae (SFGR)*	*antinuclear antibody*
Alternative Transcripts	Names differ from primary transcript.	Tag if primary transcript named.	*I kappa B gamma,VEGF20*	*Exon 2, IIA*

Table 2: Some problematic gene/protein annotations and conventions followed in GENETAG-05.

Corpus design for biomedical natural language processing

K. Bretonnel Cohen
Center for Computational Pharmacology
U. of Colorado School of Medicine
Aurora, Colorado
kevin.cohen@gmail.com

Lynne Fox
Denison Library
U. of Colorado Health Sciences Center
Denver, Colorado
lynne.fox@uchsc.edu

Philip V. Ogren
Center for Computational Pharmacology
U. of Colorado School of Medicine
Aurora, Colorado
philip.ogren@uchsc.edu

Lawrence Hunter
Center for Computational Pharmacology
U. of Colorado School of Medicine
Aurora, Colorado
larry.hunter@uchsc.edu

Abstract

This paper classifies six publicly available biomedical corpora according to various corpus design features and characteristics. We then present usage data for the six corpora. We show that corpora that are carefully annotated with respect to structural and linguistic characteristics and that are distributed in standard formats are more widely used than corpora that are not. These findings have implications for the design of the next generation of biomedical corpora.

1 Introduction

A small number of data sets for evaluating the performance of biomedical language processing (BLP) systems on a small number of task types have been made publicly available by their creators (Blaschke et al. 1999[1], Craven and Kumlein 1999[2], Pustejovsky et al. 2002[3], Franzén et al. 2002[4], Collier et al. 1999[5], Tanabe et al. 2005[6]). From a biological perspective, a number of these corpora (PDG, GENIA, Medstract, Yapex, GENETAG) are exceptionally well curated. From the perspective of system evaluation, a number of these corpora (Wisconsin, GENETAG) are very well designed, with large numbers of both positive and negative examples for system training and testing. Despite the positive attributes of all of these corpora, they vary widely in their external usage rates: some of them have been found very useful in the natural language processing community outside of the labs that created them, as evinced by their high rates of usage in system construction and evaluation in the years since they have been released. In contrast, others have seen little or no use in the community at large. These data sets provide us with an opportunity to evaluate the consequences of a variety of approaches to biomedical corpus construction. We examine these corpora with respect to a number of design features and other characteristics, and look for features that characterize widely used—and infrequently used—corpora. Our findings have implications for how the next generation of biomedical corpora should be constructed, and for how the existing corpora can be modified to make them more widely useful.

2 Materials and methods

Table 1 lists the publicly available biomedical corpora of which we are aware. We omit discussion here of the corpus currently in production by the University of Pennsylvania and the Children's Hospital of Philadelphia (Kulick et al. 2004), since it is not yet available in finished form. We also omit text collections from our discussion. By *text collection* we mean textual data sets that may include metadata about documents, but do not contain mark-up of the document contents. So, the OHSUMED text collec-

[1] We refer to this corpus as the Protein Design Group (PDG) corpus.

[2] We refer to this as the University of Wisconsin corpus.

[3] The Medstract corpus.

[4] The Yapex corpus.

[5] The GENIA corpus.

[6] Originally the BioCreative Task 1A data set, now known as the GENETAG corpus.

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 38–45, Detroit, June 2005.

Table 1: Name, date, genre, and size for the six corpora. Size is in words.

Name	date	genre	size
PDG	1999	Sentences	10,291
Wisconsin	1999	Sentences	1,529,731
GENIA	1999	Abstracts	432,560
MEDSTRACT	2001	Abstracts	49,138
Yapex	2002	Abstracts	45,143
GENETAG	2004	Sentences	342,574

Table 2: Low- and high-level tasks to which the six corpora are applicable. SS is sentence segmentation, T is tokenization, and POS is part-of-speech tagging. EI is entity identification, IE is information extraction, A is acronym/abbreviation definition, and C is coreference resolution.

Name	SS	T	POS	EI	IE	A	C
PDG				•	•		
Wisconsin				•	•		
GENIA	•	•	•	•			
Medstract				•		•	•
Yapex				•			
GENETAG				•			

tion (Hersh et al. 1994) and the TREC Genomics track data sets (Hersh and Bhupatiraju 2003, Hersh et al. 2004) are excluded from this work, although their utility in information retrieval is clear.

Table 1 lists the corpora, and for each corpus, gives its release date (or the year of the corresponding publication), the genre of the contents of the corpus, and the size of the corpus[7].

The left-hand side of Table 2 lists the data sets and, for each one, indicates the lower-level general language processing problems that it could be applied to, either as a source of training data or for evaluating systems that perform these tasks. We considered here sentence segmentation, word tokenization, and part-of-speech (POS) tagging.

The right-hand side of Table 2 shows the higher-level tasks to which the various corpora can be applied. We considered here entity identification, information (relation) extraction, abbreviation/acronym definition, and coreference resolution. (Information retrieval is approached via text collections, versus corpora.) These tasks are directly related to the types of semantic annotation present in each corpus. The three EI-only corpora (GENIA, Yapex, GENETAG) are annotated with semantic classes of relevance to the molecular biology domain. In the case of the Yapex and GENETAG corpora, this annotation uses a single semantic class, roughly equivalent to the gene or gene product. In the case of the GENIA corpus, the annotation reflects a more sophisticated, if not widely used, ontology. The Medstract corpus uses multiple semantic classes, including *gene*, *protein*, *cell type*, and *molecular process*. In all of these cases, the semantic annotation was carefully curated, and in one (GENETAG) it includes alternative analyses. Two of the corpora (PDG, Wisconsin) are indicated in Table 2 as being applicable to both entity identification and information extraction tasks. From a biological perspective, the PDG corpus has exceptionally well-curated positive examples. From a linguistic perspective, it is almost unannotated. For each sentence, the entities are listed, but their locations in the text are not indicated, making them applicable to some definitions of the entity identification task but not others. The Wisconsin corpus contains both positive and negative examples. For each example, entities are listed in a normalized form, but without clear pointers to their locations in the text, making this corpus similarly difficult to apply to many definitions of the entity identification task.

The Medstract corpus is unique among these in being annotated with coreferential equivalence sets, and also with acronym expansions.

All six corpora draw on the same subject matter domain—molecular biology—but they vary widely with respect to their level of semantic restriction within that relatively broad category. One (GENIA) is restricted to the subdomain of human blood cell transcription factors. Another (Yapex) combines data from this domain with abstracts on protein binding in humans. The GENETAG corpus is considerably broader in topic, with all of PubMed/MEDLINE serving as a potential data

[7] Sizes are given in words. Published descriptions of the corpora don't generally give size in words, so this data is based on our own counts. See the web site at http://compbio.uchsc.edu/corpora for details on how we did the count for each corpus.

Table 3: External usage rates. The *systems* column gives the count of the number of systems that actually used the dataset, as opposed to publications that cited the paper but did not use the data itself. *Age* is in years as of 2005.

Name	age	systems
GENIA	6	21
GENETAG	1	8
Yapex	3	6
Medstract	4	3
Wisconsin	6	1
PDG	6	0

source. The Medstract corpus contains biomedical material not apparently related to molecular biology. The PDG corpus is drawn from a very narrow subdomain on protein-protein interactions. The Wisconsin corpus is composed of data from three separate sub-domains: protein-protein interactions, subcellular localization of proteins, and gene/disease associations.

Table 3 shows the number of systems *built outside of the lab that created the corpus* that used each of the data sets described in Tables 1 and 2. The counts in this table reflect work that actually used the datasets, versus work that cites the publication that describes the data set but doesn't actually use the data set. We assembled the data for these counts by consulting with the creators of the data sets and by doing our own literature searches[8]. If a system is described in multiple publications, we count it only once, so the number of systems is slightly smaller than the number of publications.

3 Results

Even without examining the external usage data, two points are immediately evident from Tables 1 and 2:

- Only one of the currently publicly available corpora (GENIA) is suitable for evaluating performance on basic preprocessing tasks.
- These corpora include only a very limited range of genres: only abstracts and roughly sentence-sized inputs are represented.

Examination of Table 3 makes another point immediately clear. The currently publicly available corpora fall into two groups: ones that have had a number of external applications (GENIA, GENETAG, and Yapex), and ones that have not (Medstract, Wisconsin, and PDG). We now consider a number of design features and other characteristics of these corpora that might explain these groupings[9].

3.1 Effect of age

We considered the very obvious hypothesis that it might be length of time that a corpus has been available that determines the amount of use to which it has been put. (Note that we use the terms "hypothesis" and "effect" in a non-statistical sense, and there is no significance-testing in the work reported here.) Tables 1 and 3 show clearly that this is not the case. The age of the PDG, Wisconsin, and GENIA data is the same, but the usage rates are considerably different—the GENIA corpus has been much more widely used. The GENETAG corpus is the newest, but has a relatively high usage rate. Usage of a corpus is determined by factors other than the length of time that it has been available.

3.2 Effect of size

We considered the hypothesis that size might be the determinant of the amount of use to which a corpus is put—perhaps smaller corpora simply do not provide enough data to be helpful in the development and validation of learning-based systems. We can

[8] In the cases of the two corpora for which we found only zero or one external usage, this search was repeated by an experienced medical librarian, and included reviewing 67 abstracts or full papers that cite Blaschke et al. (1999) and 37 that cite Craven and Kumlein (1999).

[9] Three points should be kept in mind with respect to this data. First, although the sample includes all of the corpora that we are aware of, it is small. Second, there is a variety of potential confounds related to sociological factors which we are aware of, but do not know how to quantify. One of these is the effect of association between a corpus and a shared task. This would tend to increase the usage of the corpus, and could explain the usage rates of GENIA and GENETAG, although not that of Yapex. Another is the effect of association between a corpus and an influential scientist. This might tend to increase the usage of the corpus, and could explain the usage rate of GENIA, although not that of GENETAG. Finally, there may be interactions between any of these factors, or as a reviewer pointed out, there may be a separate explanation for the usage rate of each corpus in this study. Nevertheless, the analysis of the quantifiable factors presented above clearly provides useful information about the design of successful corpora.

reject this hypothesis: the Yapex corpus is one of the smallest (a fraction of the size of the largest, and only roughly a tenth of the size of GENIA), but has achieved fairly wide usage. The Wisconsin corpus is the largest, but has a very low usage rate.

3.3 Effect of structural and linguistic annotation

We expected a priori that the corpus with the most extensive structural and linguistic annotation would have the highest usage rate. (In this context, by *structural annotation* we mean tokenization and sentence segmentation, and by *linguistic annotation* we mean POS tagging and shallow parsing.) There isn't a clear-cut answer to this.

The GENIA corpus is the only one with curated structural and POS annotation, and it has the highest usage rate. This is consistent with our initial hypothesis.

On the other hand, the Wisconsin corpus could be considered the most "deeply" linguistically annotated, since it has both POS annotation and—unique among the various corpora—shallow parsing. It nevertheless has a very low usage rate. However, the comparison is not clearcut, since both the POS tagging and the shallow parsing are fully automatic and not manually corrected. (Additionally, the shallow parsing and the tokenization on which it is based are somewhat idiosyncratic.) It *is* clear that the Yapex corpus has relatively high usage despite the fact that it is, from a linguistic perspective, very lightly annotated (it is marked up for entities only, and nothing else). To our surprise, structural and linguistic annotation do not appear to uniquely determine usage rate.

3.4 Effect of format

Annotation format has a large effect on usage. It bears repeating that these six corpora are distributed in six different formats—even the presumably simple task of populating the *Size* column in Table 1 required writing six scripts to parse the various data files. The two lowest-usage corpora are annotated in remarkably unique formats. In contrast, the three more widely used corpora are distributed in relatively more common formats. Two of them (GENIA and Yapex) are distributed in XML, and one of them (GENIA) offers a choice for POS tagging information between XML and the whitespace-separated, one-token-followed-by-tags-per-line format that is common to a number of POS taggers and parsers. The third (GENETAG) is distributed in the widely used slash-attached format (e.g. *sense/NN*).

3.5 Effect of semantic annotation

The data in Table 2 and Table 3 are consistent with the hypothesis that semantic annotation predicts usage. The claim would be that corpora that are built specifically for entity identification purposes are more widely used than corpora of other types, presumably due to a combination of the importance of the entity identification task as a prerequisite to a number of other important applications (e.g. information extraction and retrieval) and the fact that it is still an unsolved problem. There may be some truth to this, but we doubt that this is the full story: there are large differences in the usage rates of the three EI corpora, suggesting that semantic annotation is not the only relevant design feature. If this analysis is in fact correct, then certainly we should see a reduction in the use of all three of these corpora once the EI problem is solved, unless their semantic annotations are extended in new directions.

3.6 Effect of semantic domain

Both the advantages and the disadvantages of restricted domains as targets for language processing systems are well known, and they seem to balance out here. The scope of the domain does not affect usage: both the low-use and higher-use groups of corpora contain at least one highly restricted domain (GENIA in the high-use group, and PDG in the low-use group) and one broader domain (GENETAG in the high-use group, and Wisconsin in the lower-use group).

4 Discussion

The data presented in this paper show clearly that external usage rates vary widely for publicly available biomedical corpora. This variability is not related to the biological relevance of the corpora—the PDG and Wisconsin corpora are clearly of high biological relevance as evinced by the number of systems that have tackled the information extraction tasks that they are meant to support. Additionally, from a biological perspective, the quality of the data in the

PDG corpus is exceptionally high. Rather, our data suggest that basic issues of distribution format and of structural and linguistic annotation seem to be the strongest predictors of how widely used a biomedical corpus will be. This means that as builders of of data sources for BLP, we can benefit from the extensive experience of the corpus linguistics world. Based on that experience, and on the data that we have presented in this paper, we offer a number of suggestions for the design of the next generation of biomedical corpora.

We also suggest that the considerable investments already made in the construction of the less-frequently-used corpora can be protected by modifying those corpora in accordance with these suggestions.

Leech (1993) and McEnery and Wilson (2001), coming from the perspective of corpus linguistics, identify a number of definitional issues and design maxims for corpus construction. Some of these are quite relevant to the current state of biomedical corpus construction. We frame the remainder of our discussion in terms of these issues and maxims.

4.1 Level of annotation

From a definitional point of view, annotation is one of the distinguishing points of a corpus, as opposed to a text collection. Perhaps the most salient characteristic of the currently publicly available corpora is that from a linguistic or language processing perspective, with the exception of GENIA and GENETAG, they are barely annotated at all. For example, although POS tagging has possibly been the sine qua non of the usable corpus since the earliest days of the modern corpus linguistic age, five of the six corpora listed in Table 2 either have no POS tagging or have only automatically generated, uncorrected POS tags. The GENIA corpus, with its carefully curated annotation of sentence segmentation, tokenization, and part-of-speech tagging, should serve as a model for future biomedical corpora in this respect. It is remarkable that with just these levels of annotation (in addition to its semantic mark-up), the GENIA corpus has been applied to a wide range of task types other than the one that it was originally designed for. Eight papers from COLING 2004 (Kim et al. 2004) used it for evaluating entity identification tasks. Yang et al. (2002) adapted a subset of the corpus for use in developing and testing a coreference resolution system. Rinaldi et al. (2004) used it to develop and test a question-answering system. Locally, it has been used in teaching computational corpus linguistics for the past two years. We do not claim that it has not required extension for some of these tasks—our claim is that it is its annotation on these structural and linguistic levels, in combination with its format, that has made these extensions practical.

4.1.1 Formatting choices and formatting standardization

A basic desideratum for a corpus is *recoverability:* it should be possible to map from the annotation to the raw text. A related principle is that it should be easy for the corpus user to extract all annotation information from the corpus, e.g. for external storage and processing: "in other words, the annotated corpus should allow the maximum flexibility for manipulation by the user" (McEnery and Wilson, p. 33). The extent to which these principles are met is a function of the annotation format. The currently available corpora are distributed in a variety of one-off formats. Working with any one of them requires learning a new format, and typically writing code to access it. At a minimum, none of the non-XML corpora meet the recoverability criterion. None[10] of these corpora are distributed in a standoff annotation format. *Standoff annotation* is the strategy of storing annotation and raw text separately (Leech 1993). Table 4 contrasts the two. Non-standoff annotation at least obscures—more frequently, destroys—important aspects of the structure of the text itself, such as which textual items are and are not immediately adjacent. Using standoff annotation, there is no information loss whatsoever. Furthermore, in the standoff annotation strategy, the original input text is immediately available in its raw form. In contrast, in the non-standoff annotation strategy, the original must be retrieved independently or recovered from the annotation (if it is recoverable at all). The standoff annotation strategy was relatively new at the time that most of the corpora in Table 1 were designed, but by now has become easy to implement, in part

[10]The semantic annotation of the GENETAG corpus is in a standoff format, but neither the tokenization nor the POS tagging is.

Table 4: Contrasting standoff and non-standoff annotation

Raw text
MLK2 has a role in vesicle formation
Non-standoff annotation
MLK2/NN has/VBZ a/DT role/NN in/IN vesicle/NN formation/NN
Standoff annotation
<POS="NN" start=0 end=3> <POS="VBZ" start=5 end=7> <POS="DT" start=9 end=9> <POS="NN" start=11 end=14> <POS="IN" start=16 end=17> <POS="NN" start=19 end=25> <POS="NN" start=27 end=35>

due to the availability of tools such as the University of Pennsylvania's WordFreak (Morton and LaCivita 2003).

Crucially, this annotation should be based on character offsets, avoiding a priori assumptions about tokenization. See Smith et al. (2005) for an approach to refactoring a corpus to use character offsets.

4.1.2 Guidelines

The maxim of *documentation* suggests that annotation guidelines should be published. Further, basic data on who did the annotations and on their level of agreement should be available. The currently available datasets mostly lack assessments of inter-annotator agreement, utilize a small or unspecified number of annotators, and do not provide published annotation guidelines. (We note the Yang et al. (2002) coreference annotation guidelines, which are excellent, but the corresponding corpus is not publicly available.) This situation can be remedied by editors, who should insist on publication of all of these. The GENETAG corpus is notable for the detailed documentation of its annotation guidelines. We suspect that the level of detail of these guidelines contributed greatly to the success of some rule-based approaches to the EI task in the BioCreative competition, which utilized an early version of this corpus.

4.1.3 Balance and representativeness

Corpus linguists generally strive for a well-structured stratified sample of language, seeking to "balance" in their data the representation of text types, different sorts of authors, and so on. Within the semantic domain of molecular biology texts, an important dimension on which to balance is the genre or text type.

As is evident from Table 1, the extant datasets draw on a very small subset of the types of genres that are relevant to BLP: we have not done a good job yet of observing the principle of balance or representativeness. The range of genres that exist in the research (as opposed to clinical) domain alone includes abstracts, full-text articles, GeneRIFs, definitions, and books. We suggest that all of these should be included in future corpus development efforts.

Some of these genres have been shown to have distinguishing characteristics that are relevant to BLP. Abstracts and isolated sentences from them are inadequate, and also unsuited to the opportunities that are now available to us for text data mining with the recent announcement of the NIH's new policy on availability of full-text articles (NIH 2005). This policy will result in the public availability of a large and constantly growing archive of current, full-text publications. Abstracts and sentences are inadequate in that experience has shown that significant amounts of data are not found in abstracts at all, but are present only in the full texts of articles, sometimes not even in the body of the text itself, but rather in tables and figure captions (Shatkay and Feldman 2003). They are not suited to the upcoming opportunities in that it is not clear that practicing on abstracts will let us build the necessary skills for dealing with the flood of full-text articles that PubMedCentral is poised to deliver to us. Furthermore, there are other types of data—GeneRIFs and domain-specific dictionary definitions, for instance—that are fruitful sources of biological knowledge, and which may actually be easier to process automatically than abstracts. Space does not permit justifying the importance of all of these genres, but we discuss the rationale for including full text at some length due to the recent NIH announcement and due to the large body of evidence that can currently be brought to bear on the issue. A growing body of recent research makes

two points clear: full-text articles are different from abstracts, and full-text articles must be tapped if we are to build high-recall text data mining systems.

Corney et al. (2004) looked directly at the effectiveness of information extraction from full-text articles versus abstracts. They found that recall from full-text articles was more than double that from abstracts. Analyzing the relative contributions of the abstracts and the full articles, they found that more than half of the interactions that they were able to extract were found in the full text and were absent in the abstract.

Tanabe and Wilbur (2002) looked at the performance on full-text articles of an entity identification system that had originally been developed and tested using abstracts. They found different false positive rates in the Methods sections compared to other sections of full-text articles. This suggests that full-text articles, unlike abstracts, will require parsing of document structure. They also noted a range of problems related to the wider range of characters (including, e.g., superscripts and Greek letters) that occurs in full-text articles, as opposed to abstracts.

Schuemie et al. (2004) examined a set of 3902 full-text articles from *Nature Genetics* and BioMed Central, along with their abstracts. They found that about twice as many MeSH concepts were mentioned in the full-text articles as in the abstracts. They also found that full texts contained a larger number of unique gene names than did abstracts, with an average of 2.35 unique gene names in the full-text articles, but an average of only 0.61 unique gene names in the abstracts.

It seems clear that for biomedical text data mining systems to reach anything like their full potential, they will need to be able to handle full-text inputs. However, as Table 1 shows, no publicly available corpus contains full-text articles. This is a deficiency that should be remedied.

5 Conclusion

5.1 Best practices in biomedical corpus construction

We have discussed the importance of recoverability, publication of guidelines, balance and representativeness, and linguistic annotation. Corpus maintenance is also important. Bada et al. (2004) point out the role that an organized and responsive maintenance plan has played in the success of the Gene Ontology. It seems likely that the continued development and maintenance reflected in the three major releases of GENIA (Ohta et al. 2002, Kim et al. 2003) have contributed to its improved quality and continued use over the years.

5.2 A testable prediction

We have interpreted the data on the characteristics and usage rates of the various datasets discussed in this paper as suggesting that datasets that are developed in accordance with basic principles of corpus linguistics are more useful, and therefore more used, than datasets that are not.

A current project at the University of Pennsylvania and the Children's Hospital of Philadelphia (Kulick et al. 2004) is producing a corpus that follows many of these basic principles. We predict that this corpus will see wide use by groups other than the one that created it.

5.3 The next step: grounded references

The logical next step for BLP corpus construction efforts is the production of corpora in which entities and concepts are grounded with respect to external models of the world (Morgan et al. 2004).

The BioCreative Task 1B data set construction effort provides a proof-of-concept of the plausibility of building BLP corpora that are grounded with respect to external models of the world, and in particular, biological databases. These will be crucial in taking us beyond the stage of extracting information about text strings, and towards mining knowledge about known, biologically relevant entities.

6 Acknowledgements

This work was supported by NIH grant R01-LM008111. The authors gratefully acknowledge helpful discussions with Lynette Hirschman, Alex Morgan, and Kristofer Franzén, and thank Sonia Leach and Todd A. Gibson for LaTeXassistance. Christian Blaschke, Mark Craven, Lorraine Tanabe, and again Kristofer Franzén provided helpful data. We thank all of the corpus builders for their generosity in sharing their valuable resources.

References

Bada, Michael; Robert Stevens; et al. 2004. A short study on the success of the Gene Ontology. *Journal of web semantics* 1(2):235-240.

Blaschke, Christian; Miguel A. Andrade; Christos Ouzounis; and Alfonso Valencia. 1999. Automatic extraction of biological information from scientific text: protein-protein interactions. *ISMB-99*, pp. 60-67. AAAI Press.

Collier, Nigel, Hyun Seok Park, Norihiro Ogata, Yuka Tateisi, Chikashi Nobata, Takeshi Sekimizu, Hisao Imai and Jun'ichi Tsujii. 1999. The GENIA project: corpus-based knowledge acquisition and information extraction from genome research papers. *EACL 1999*.

Corney, David P.A.; Bernard F. Buxton; William B. Langdon; and David T. Jones. 2004. BioRAT: extracting biological information from full-length papers. *Bioinformatics* 20(17):3206-3213.

Craven, Mark; and Johan Kumlein. 1999. Constructing biological knowledge bases by extracting information from text sources. *ISMB-99*, pp. 77-86. AAAI Press.

Franzén, Kristofer; Gunnar Eriksson; Fredrik Olsson; Lars Asker Per Lidén; and Joakim Cöster. 2002. Protein names and how to find them. *International Journal of Medical Informatics*, 67(1-3), pp. 49-61.

Hersh, William; Chris Buckley; TJ Leone; and David Hickam. 1994. OHSUMED: an interactive retrieval evaluation and new large test collection for research. *SIGIR94*, pp. 192-201.

Hersh, William; and Ravi Teja Bhupatiraju. 2003. TREC genomics track overview. *TREC 2003*, pp. 14-23.

Hersh et al. 2004. TREC 2004 genomics track overview. *TREC Notebook*.

Kim, Jin-Dong; Tomoko Ohta; Yuka Tateisi; and Jun'ichi Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics* 19(Suppl. 1):i180-i182.

Kim, Jin-Dong; Tomoko Ohta; Yoshimasa Tsuruoka; and Yuka Tateisi. 2004. Introduction to the bio-entity recognition task at JNLPBA. *Proc. international joint workshop on natural language processing in biomedicine and its applications*, pp. 70-75.

Kulick, Seth; Ann Bies; Mark Liberman; Mark Mandel; Ryan McDonald; Martha Palmer; Andrew Schein; and Lyle Ungar. 2004. Integrated annotation for biomedical information extraction. *BioLink 2004*, pp. 61-68.

Leech, G. 1993. Corpus annotation schemes. *Literary and linguistic computing* 8(4):275-281.

McEnery, Tony; and Andrew Wilson. 2001. *Corpus linguistics*, 2nd edition. Edinburgh University Press.

Morgan, Alexander A.; Lynette Hirschman; Marc Colosimo; Alexander S. Yeh; and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *JBMI* 37:396-410.

Morton, Thomas; and Jeremy LaCivita. 2003. WordFreak: an open tool for linguistic annotation. *HLT/NAACL 2003: demonstrations*, pp. 17-18.

NIH (National Institutes of Health). 2005. http://www.nih.gov/news/pr/feb2005/od-03.htm

Ohta, Tomoko; Yuka Tateisi; and Jin-Dong Kim. 2002. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. *HLT 2002*, pp. 73-77.

Pustejovsky, James; José Castaño; R. Saurí; A. Rumshisky; J. Zhang; and W. Luo. 2002. Medstract: creating large-scale information servers for biomedical libraries. *Proc. workshop on natural language processing in the biomedical domain*, pp. 85-92. Association for Computational Linguistics.

Rinaldi, Fabio; James Dowdall; Gerold Schneider; and Andreas Persidis. 2004. Answering questions in the genomics domain. *Proc. ACL 2004 workshop on question answering in restricted domains*, pp. 46-53.

Schuemie, M.J.; M. Weeber; B.J. Schijvenaars; E.M. van Mulligen; C.C. van der Eijk; R. Jelier; B. Mons; and J.A. Kors. 2004. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics* 20(16):2597-2604.

Shatkay, Hagit; and Ronen Feldman. 2003. Mining the biomedical literature in the genomic era: an overview. *Journal of computational biology* 10(6):821-855.

Smith, Lawrence H.; Lorraine Tanabe; Thomas Rindflesch; and W. John Wilbur. 2005. MedTag: a collection of biomedical annotations. *BioLINK 2005*, this volume.

Tanabe, Lorraine; and L. John Wilbur. 2002. Tagging gene and protein names in full text articles. *Proc. ACL workshop on natural language processing in the biomedical domain*, pp. 9-13.

Tanabe, Lorraine; Natalie Xie; Lynne H. Thom; Wayne Matten; and W. John Wilbur. 2005. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics* 6(Suppl. 1):S3.

Yang, Xiaofeng; Guodong Zhou; Jian Su; and Chew Lim Tan. Improving noun phrase coreference resolution by matching strings. 2002. *IJCNLP04*, pp. 326-333.

Using Biomedical Literature Mining to Consolidate the Set of Known Human Protein–Protein Interactions

Arun Ramani, Edward Marcotte
Institute for Cellular and Molecular Biology
University of Texas at Austin
1 University Station A4800
Austin, TX 78712
arun@icmb.utexas.edu
marcotte@icmb.utexas.edu

Razvan Bunescu, Raymond Mooney
Department of Computer Sciences
University of Texas at Austin
1 University Station C0500
Austin, TX 78712
razvan@cs.utexas.edu
mooney@cs.utexas.edu

Abstract

This paper presents the results of a large-scale effort to construct a comprehensive database of known human protein interactions by combining and linking known interactions from existing databases and then adding to them by automatically mining additional interactions from 750,000 Medline abstracts. The end result is a network of 31,609 interactions amongst 7,748 proteins. The text mining system first identifies protein names in the text using a trained Conditional Random Field (CRF) and then identifies interactions through a filtered co-citation analysis. We also report two new strategies for mining interactions, either by finding explicit statements of interactions in the text using learned pattern-based rules or a Support-Vector Machine using a string kernel. Using information in existing ontologies, the automatically extracted data is shown to be of equivalent accuracy to manually curated data sets.

1 Introduction

Proteins are often considered in terms of their networks of interactions, a view that has spurred considerable effort in mapping large-scale protein interaction networks. Thus far, the most complete protein networks are measured for yeast and derive from the synthesis of varied large scale experimental interaction data and in-silico interaction predictions (summarized in (von Mering et al., 2002; Lee et al., 2004; Jansen et al., 2003)). Unlike the case of yeast, only minimal progress has been made with respect to the human proteome. While some moderate-scale interaction maps have been created, such as for the purified TNFa/NFKB protein complex (Bouwmeester et al., 2004) and the proteins involved in the human Smad signaling pathway (Colland et al., 2004), the bulk of known human protein interaction data derives from individual, small-scale experiments reported in Medline. Many of these interactions have been collected in the Reactome (Joshi-Tope et al., 2005), BIND (Bader et al., 2003), DIP (Xenarios et al., 2002), and HPRD (Peri et al., 2004) databases, with Reactome contributing 11,000 interactions that have been manually entered from articles focusing on interactions in core cellular pathways, and HPRD contributing a set of 12,000 interactions recovered by manual curation of Medline articles using teams of readers. Additional interactions have been transferred from other organisms based on orthology (Lehner and Fraser, 2004).

A comparison of these existing interaction data sets is enlightening. Although the interactions from these data sets are in principle derived from the same source (Medline), the sets are quite disjoint (Figure 1) implying either that the sets are biased for different classes of interactions, or that the actual number of interactions in Medline is quite large. We suspect both reasons. It is clear that each data set has a different explicit focus (Reactome towards core cellular machinery, HPRD towards disease-linked genes, and DIP and BIND more randomly

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 46–53, Detroit, June 2005.

distributed). Due to these biases, it is likely that many interactions from Medline are still excluded from these data sets. The maximal overlap between interaction data sets is seen for BIND: 25% of these interactions are also in HPRD or Reactome; only 1% of Reactome interactions are in HPRD or BIND.

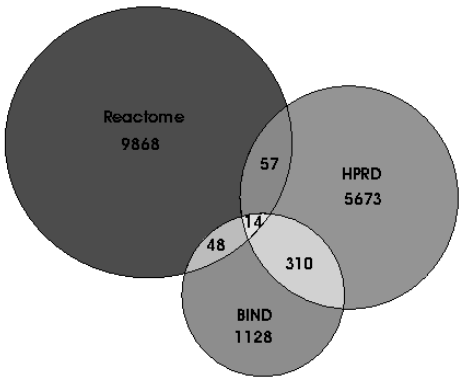


Figure 1: Overlap diagram for known datasets.

Medline now has records from more than 4,800 journals accounting for around 15 million articles. These citations contain thousands of experimentally recorded protein interactions, and even a cursory investigation of Medline reveals human protein interactions not present in the current databases. However, retrieving these data manually is made difficult by the large number of articles, all lacking formal structure. Automated extraction of information would be preferable, and therefore, mining data from Medline abstracts is a growing field (Jenssen et al., 2001; Rzhetsky et al., 2004; Liu and Wong, 2003; Hirschman et al., 2002).

In this paper, we describe a framework for automatic extraction of protein interactions from biomedical literature. We focus in particular on the difficult and important problem of identifying interactions concerning human proteins. We describe a system for first accurately identifying the names of human proteins in the documents, then on identifying pairs of interacting human proteins, and demonstrate that the extracted protein interactions are comparable to those extracted manually. In the process, we consolidate the existing set of publically-available human protein interactions into a network of 31,609 interactions between 7,748 proteins.

2 Assembling existing protein interaction data

We previously gathered the existing human protein interaction data sets ((Ramani et al., 2005); summarized in Table 1), representing the current status of the publically-available human interactome. This required unification of the interactions under a shared naming and annotation convention. For this purpose, we mapped each interacting protein to LocusLink (now EntrezGene) identification numbers and retained only unique interactions (i.e., for two proteins A and B, we retain only A–B or B–A, not both). We have chosen to omit self-interactions, A–A or B–B, for technical reasons, as their quality cannot be assessed on the functional benchmark that we describe in Section 3. In most cases, a small loss of proteins occurred in the conversion between the different gene identifiers (e.g., converting from the NCBI 'gi' codes in BIND to LocusLink identifiers). In the case of Human Protein Reference Database (HPRD), this processing resulted in a significant reduction in the number of interactions from 12,013 total interactions to 6,054 unique, non-self interactions, largely due to the fact that HPRD often records both A-B and B-A interactions, as well as a large number of self interactions, and indexes genes by their common names rather than conventional database entries, often resulting in multiple entries for different synonyms. An additional 9,283 (or 60,000 at lower confidence) interactions are available from orthologous transfer of interactions from large-scale screens in other organisms (orthology-core and orthology-all) (Lehner and Fraser, 2004).

3 Two benchmark tests of accuracy for interaction data

To measure the relative accuracy of each protein interaction data set, we established two benchmarks of interaction accuracy, one based on shared protein function and the other based on previously known interactions. First, we constructed a benchmark in which we tested the extent to which interaction partners in a data set shared annotation, a measure previously shown to correlate with the accuracy of functional genomics data sets (von Mering et al., 2002; Lee et al., 2004; Lehner and Fraser, 2004). We used the functional annotations listed in the KEGG

Dataset	Version	Total Is (Ps)	Self (A-A) Is (Ps)	Unique (A-B) Is (Ps)
Reactome	08/03/04	12,497 (6,257)	160 (160)	12,336 (807)
BIND	08/03/04	6,212 (5,412)	549 (549)	5,663 (4,762)
HPRD*	04/12/04	12,013 (4,122)	3,028 (3,028)	6,054 (2,747)
Orthology (all)	03/31/04	71,497 (6,257)	373 (373)	71,124 (6,228)
Orthology (core)	03/31/04	11,488 (3,918)	206 (206)	11,282 (3,863)

Table 1: **Is** = **I**nteraction**s**, **Ps** = **P**rotein**s**.

(Kanehisa et al., 2004) and Gene Ontology (Ashburner et al., 2000) annotation databases. These databases provide specific pathway and biological process annotations for approximately 7,500 human genes, assigning human genes into 155 KEGG pathways (at the lowest level of KEGG) and 1,356 GO pathways (at level 8 of the GO biological process annotation). KEGG and GO annotations were combined into a single composite functional annotation set, which was then split into independent testing and training sets by randomly assigning annotated genes into the two categories (3,800 and 3,815 annotated genes respectively). For the second benchmark based on known physical interactions, we assembled the human protein interactions from Reactome and BIND, a set of 11,425 interactions between 1,710 proteins. Each benchmark therefore consists of a set of binary relations between proteins, either based on proteins sharing annotation or physically interacting. Generally speaking, we expect more accurate protein interaction data sets to be more enriched in these protein pairs. More specifically, we expect true physical interactions to score highly on both tests, while non-physical or indirect associations, such as genetic associations, should score highly on the functional, but not physical interaction, test.

For both benchmarks, the scoring scheme for measuring interaction set accuracy is in the form of a log odds ratio of gene pairs either sharing annotations or physically interacting. To evaluate a data set, we calculate a log likelihood ratio (LLR) as:

$$LLR = ln\frac{P(D|I)}{P(D|\neg I)} = ln\frac{P(I|D)P(\neg I)}{P(\neg I|D)P(I)} \quad (1)$$

where $P(D|I)$ and $P(D|\neg I)$ are the probability of observing the data D conditioned on the genes sharing benchmark associations (I) and not sharing benchmark associations ($\neg I$). In its expanded form (obtained by applying Bayes theorem), $P(I|D)$ and $P(\neg I|D)$ are estimated using the frequencies of interactions observed in the given data set D between annotated genes sharing benchmark associations and not sharing associations, respectively, while the priors $P(I)$ and $P(\neg I)$ are estimated based on the total frequencies of all benchmark genes sharing the same associations and not sharing associations, respectively. A score of zero indicates interaction partners in the data set being tested are no more likely than random to belong to the same pathway or to interact; higher scores indicate a more accurate data set.

Among the literature-derived interactions (Reactome, BIND, HPRD), a total of 17,098 unique interactions occur in the public data sets. Testing the existing protein interaction data on the functional benchmark reveals that Reactome has the highest accuracy (LLR = 3.8), followed by BIND (LLR = 2.9), HPRD (LLR = 2.1), core orthology-inferred interactions (LLR = 2.1) and the non-core orthology-inferred interaction (LLR = 1.1). The two most accurate data sets, Reactome and BIND, form the basis of the protein interaction–based benchmark. Testing the remaining data sets on this benchmark (i.e., for their consistency with these accurate protein interaction data sets) reveals a similar ranking in the remaining data. Core orthology-inferred interactions are the most accurate (LLR = 5.0), followed by HPRD (LLR = 3.7) and non-core orthology inferred interactions (LLR = 3.7).

4 Framework for Mining Protein–Protein Interactions

The extraction of interacting proteins from Medline abstracts proceeds in two separate steps:

1. First, we automatically identify protein names

using a CRF system trained on a set of 750 abstracts manually annotated for proteins (see Section 5 for details).

2. Based on the output of the CRF tagger, we filter out less confident extractions and then try to detect which pairs of the remaining extracted protein names are interaction pairs.

For the second step, we investigate two general methods:

- Use co-citation analysis to score each pair of proteins based on the assumption that proteins co-occurring in a large number of abstracts tend to be interacting proteins. Out of the resulting protein pairs we keep only those that co-occur in abstracts likely to discuss interactions, based on a Naive Bayes classifier (see Section 6 for details).

- Given that we already have a set of 230 Medline abstracts manually tagged for both proteins and interactions, we can use it to train an interaction extractor. In Section 7 we discuss two different methods for learning this interaction extractor.

5 A CRF Tagger for Protein Names

The task of identifying protein names is made difficult by the fact that unlike other organisms, such as yeast or E. coli, the human genes have no standardized naming convention, and thus present one of the hardest sets of gene/protein names to extract. For example, human proteins may be named with typical English words, such as ”light”, ”map”, ”complement”, and ”Sonic Hedgehog”. It is therefore necessary that an information extraction algorithm be specifically trained to extract gene and protein names accurately.

We have previously described (Bunescu et al., 2005) effective protein and gene name tagging using a Maximum Entropy based algorithm. Conditional Random Fields (CRF) (Lafferty et al., 2001) are new types of probabilistic models that preserve all the advantages of Maximum Entropy models and at the same time avoid the label bias problem by allowing a sequence of tagging decisions to compete against each other in a global probabilistic model.

In both training and testing the CRF protein-name tagger, the corresponding Medline abstracts were processed as follows. Text was tokenized using white-space as delimiters and treating all punctuation marks as separate tokens. The text was segmented into sentences, and part-of-speech tags were assigned to each token using Brill’s tagger (Brill, 1995). For each token in each sentence, a vector of binary features was generated using the feature templates employed by the Maximum Entropy approach described in (Bunescu et al., 2005). Generally, these features make use of the words occurring before and after the current position in the text, their POS tags and capitalization patterns. Each feature occurring in the training data is associated with a parameter in the CRF model. We used the CRF implementation from (McCallum, 2002). To train the CRF’s parameters, we used 750 Medline abstracts manually annotated for protein names (Bunescu et al., 2005). We then used the trained system to tag protein and gene names in the entire set of 753,459 Medline abstracts citing the word “human”.

In Figure 2 we compare the performance of the CRF tagger with that of the Maximum Entropy tagger from (Bunescu et al., 2005), using the same set of features, by doing 10-fold cross-validation on Yapex – a smaller dataset of 200 manually annotated abstracts (Franzen et al., 2002). Each model assigns to each extracted protein name a normalized confidence value. The precision–recall curves from Figure 2 are obtained by varying a threshold on the minimum accepted confidence. We also plot the precision and recall obtained by simply matching textual phrases against entries from a protein dictionary.

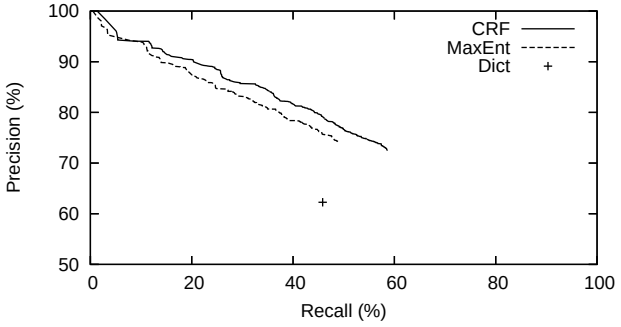


Figure 2: Protein Tagging Performance.

The dictionary of human protein names was assembled from the LocusLink and Swissprot databases by manually curating the gene names and synonyms (87,723 synonyms between 18,879 unique gene names) to remove genes that were referred to as 'hypothetical' or 'probable' and also to omit entries that referred to more than one protein identifier.

6 Co-citation Analysis and Bayesian Classification

In order to establish which interactions occurred between the proteins identified in the Medline abstracts, we used a 2-step strategy: measure co-citation of protein names, then enrich these pairs for physical interactions using a Bayesian filter. First, we counted the number of abstracts citing a pair of proteins, and then calculated the probability of co-citation under a random model based on the hypergeometric distribution (Lee et al., 2004; Jenssen et al., 2001) as:

$$P(k|N,m,n) = \frac{\left(\begin{array}{c} n \\ k \end{array}\right)\left(\begin{array}{c} N-n \\ m-k \end{array}\right)}{\left(\begin{array}{c} N \\ m \end{array}\right)} \qquad (2)$$

where N equals the total number of abstracts, n of which cite the first protein, m cite the second protein, and k cite both.

Empirically, we find the co-citation probability has a hyperbolic relationship with the accuracy on the functional annotation benchmark from Section 3, with protein pairs co–cited with low random probability scoring high on the benchmark.

With a threshold on the estimated extraction confidence of 80% (as computed by the CRF model) in the protein name identification, close to 15,000 interactions are extracted with the co-citation approach that score comparable or better on the functional benchmark than the manually extracted interactions from HPRD, which serves to establish a minimal threshold for our mined interactions.

However, it is clear that proteins are co-cited for many reasons other than physical interactions. We therefore tried to enrich specifically for physical interactions by applying a secondary filter. We applied a Bayesian classifier (Marcotte et al., 2001) to measure the likelihood of the abstracts citing the protein pairs to discuss physical protein–protein interactions. The classifier scores each of the co-citing abstracts according to the usage frequency of discriminating words relevant to physical protein interactions. For a co-cited protein pair, we calculated the average score of co-citing Medline abstracts and used this to re-rank the top-scoring 15,000 co-cited protein pairs.

Interactions extracted by co-citation and filtered using the Bayesian estimator compare favorably with the other interaction data sets on the functional annotation benchmark (Figure 3). Testing the accuracy of these extracted protein pairs on the physical interaction benchmark (Figure 4) reveals that the co-cited proteins scored high by this classifier are indeed strongly enriched for physical interactions.

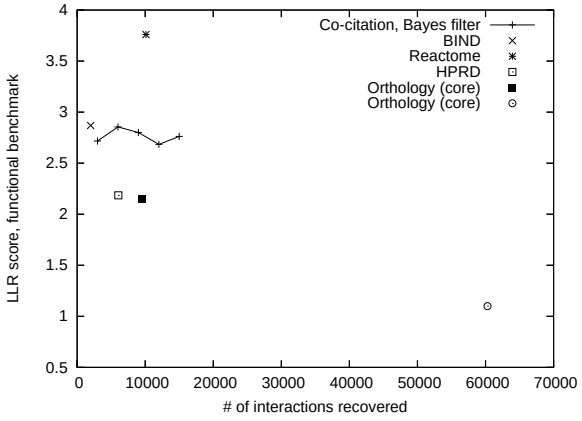


Figure 3: Accuracy, functional benchmark

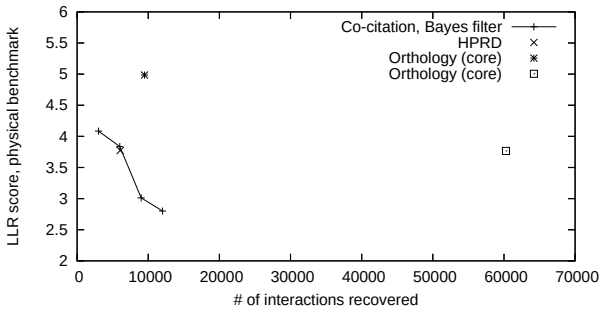


Figure 4: Accuracy, physical benchmark

Keeping all the interactions that score better than HPRD, our co-citation / Bayesian classifier analysis yields 6,580 interactions between 3,737 proteins. By combining these interactions with the 26,280 interactions from the other sources, we obtained a fi-

nal set of 31,609 interactions between 7,748 human proteins.

7 Learning Interaction Extractors

In (Bunescu et al., 2005) we described a dataset of 230 Medline abstracts manually annotated for proteins and their interactions. This can be used as a training dataset for a method that learns interaction extractors. Such a method simply classifies a sentence containing two protein names as positive or negative, where positive means that the sentence asserts an interaction between the two proteins. However a sentence in the training data may contain more than two proteins and more than one pair of interacting proteins. In order to extract the interacting pairs, we replicate the sentences having n proteins ($n \geq 2$) into C_2^n sentences such that each one has exactly two of the proteins tagged, with the rest of the protein tags omitted. If the tagged proteins interact, then the replicated sentence is added to the set of positive sentences, otherwise it is added to the set of negative sentences. During testing, a sentence having n proteins ($n \geq 2$) is again replicated into C_2^n sentences in a similar way.

7.1 Extraction using Longest Common Subsequences (ELCS)

Blaschke *et al.* (Blaschke and Valencia, 2001; Blaschke and Valencia, 2002) manually developed rules for extracting interacting proteins. Each of their rules (or frames) is a sequence of words (or POS tags) and two protein-name tokens. Between every two adjacent words is a number indicating the maximum number of intervening words allowed when matching the rule to a sentence. In (Bunescu et al., 2005) we described a new method ELCS (Extraction using Longest Common Subsequences) that automatically learns such rules. ELCS' rule representation is similar to that in (Blaschke and Valencia, 2001; Blaschke and Valencia, 2002), except that it currently does not use POS tags, but allows disjunctions of words. Figure 5 shows an example of a rule learned by ELCS. Words in square brackets separated by '|' indicate disjunctive lexical constraints, i.e. one of the given words must match the sentence at that position. The numbers in parentheses between adjacent constraints indicate the maximum number of unconstrained words allowed between the two (called a *word gap*). The protein names are denoted here with PROT. A sentence matches the rule if and only if it satisfies the word constraints in the given order and respects the respective word gaps.

```
- (7) interaction (0) [between | of] (5) PROT (9) PROT (17) .
```

Figure 5: Sample extraction rule learned by ELCS.

7.2 Extraction using a Relation Kernel (ERK)

Both Blaschke and ELCS do interaction extraction based on a limited set of matching rules, where a rule is simply a sparse (gappy) subsequence of words (or POS tags) anchored on the two protein-name tokens. Therefore, the two methods share a common limitation: either through manual selection (Blaschke), or as a result of the greedy learning procedure (ELCS), they end up using only a subset of all possible anchored sparse subsequences. Ideally, we would want to use all such anchored sparse subsequences as features, with weights reflecting their relative accuracy. However explicitly creating for each sentence a vector with a position for each such feature is infeasible, due to the high dimensionality of the feature space. Here we can exploit an idea used before in string kernels (Lodhi et al., 2002): computing the dot-product between two such vectors amounts to calculating the number of common anchored subsequences between the two sentences. This can be done very efficiently by modifying the dynamic programming algorithm from (Lodhi et al., 2002) to account only for anchored subsequences i.e. sparse subsequences which contain the two protein-name tokens. Besides restricting the word subsequences to be anchored on the two protein tokens, we can further prune down the feature space by utilizing the following property of natural language statements: whenever a sentence asserts a relationship between two entity mentions, it generally does this using one of the following three patterns:

- **[FI] F**ore–**I**nter: words before and between the two entity mentions are simultaneously used to express the relationship. Examples: 'interaction of $\langle P_1 \rangle$ with $\langle P_2 \rangle$', 'activation of $\langle P_1 \rangle$ by $\langle P_2 \rangle$'.

- **[I] I**nter: only words between the two entity mentions are essential for asserting the relationship. Examples: '$\langle P_1 \rangle$ interacts with $\langle P_2 \rangle$', '$\langle P_1 \rangle$ is activated by $\langle P_2 \rangle$'.

- **[IA] I**nter–**A**fter: words between and after the two entity mentions are simultaneously used to express the relationship. Examples: $\langle P_1 \rangle$ – $\langle P_2 \rangle$ complex', '$\langle P_1 \rangle$ and $\langle P_2 \rangle$ interact'.

Another useful observation is that all these patterns use at most 4 words to express the relationship (not counting the two entities). Consequently, when computing the relation kernel, we restrict the counting of common anchored subsequences only to those having one of the three types described above, with a maximum word-length of 4. This type of feature selection leads not only to a faster kernel computation, but also to less overfitting, which results in increased accuracy (we omit showing here comparative results supporting this claim, due to lack of space).

We used this kernel in conjunction with Support Vector Machines (Vapnik, 1998) learning in order to find a decision hyperplane that best separates the positive examples from negative examples. We modified the **libsvm** package for SVM learning by plugging in the kernel described above.

7.3 Preliminary experimental results

We compare the following three systems on the task of retrieving protein interactions from the dataset of 230 Medline abstracts (assuming gold standard proteins):

- **[Manual]**: We report the performance of the rule-based system of (Blaschke and Valencia, 2001; Blaschke and Valencia, 2002).

- **[ELCS]**: We report the 10-fold cross-validated results from (Bunescu et al., 2005) as a precision-recall graph.

- **[ERK]**: Based on the same splits as those used by ELCS, we compute the corresponding precision-recall graph.

The results, summarized in Figure 6, show that the relation kernel outperforms both ELCS and the manually written rules. In future work, we intend to analyze the complete Medline with ERK and integrate the extracted interactions into a larger composite set.

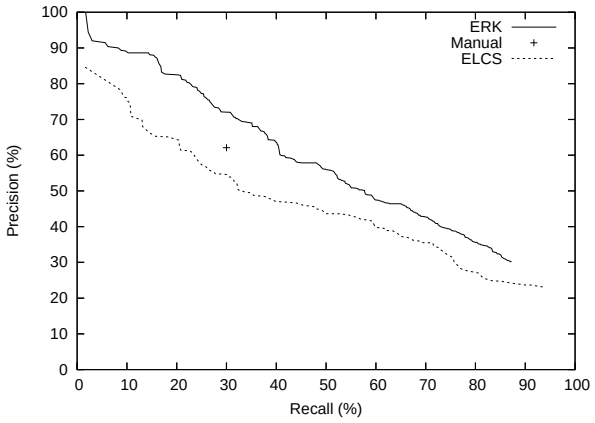


Figure 6: PR curves for interaction extractors.

8 Conclusion

Through a combination of automatic text mining and consolidation of existing databases, we have constructed a large database of known human protein interactions containing 31,609 interactions amongst 7,748 proteins. By mining 753,459 human-related abstracts from Medline with a combination of a CRF-based protein tagger, co-citation analysis, and automatic text classification, we extracted a set of 6,580 interactions between 3,737 proteins. By utilizing information in existing knowledge bases, this automatically extracted data was found to have an accuracy comparable to manually developed data sets. More details on our interaction database have been published in the biological literature (Ramani et al., 2005) and it is available on the web at http://bioinformatics.icmb.utexas.edu/idserve. We are currently exploring improvements to this database by more accurately identifying assertions of interactions in the text using an SVM that exploits a relational string kernel.

9 Acknowledgements

This work was supported by grants from the N.S.F. (IIS-0325116, EIA-0219061), N.I.H. (GM06779-01), Welch (F1515), and a Packard Fellowship (E.M.M.).

References

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, and J. T. et al. Eppig. 2000. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25(1):25–29.

G. D. Bader, D. Betel, and C. W. Hogue. 2003. Bind: the biomolecular interaction network database. *Nucleic Acids Research*, 31(1):248–250.

C. Blaschke and A. Valencia. 2001. Can bibliographic pointers for known biological data be found automatically? protein interactions as a case study. *Comparative and Functional Genomics*, 2:196–206.

C. Blaschke and A. Valencia. 2002. The frame-based module of the Suiseki information extraction system. *IEEE Intelligent Systems*, 17:14–20.

T. Bouwmeester, A. Bauch, H. Ruffner, P. O. Angrand, G. Bergamini, K. Croughton, C. Cruciat, D. Eberhard, J. Gagneur, S. Ghidelli, and et al. 2004. A physical and functional map of the human tnf-alpha/nf-kappa b signal transduction pathway. *Nature Cell Biology*, 6(2):97–105.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine (special issue on Summarization and Information Extraction from Medical Documents)*, 33(2):139–155.

F. Colland, X. Jacq, V. Trouplin, C. Mougin, C. Groizeleau, A. Hamburger, A. Meil, J. Wojcik, P. Legrain, and J. M. Gauthier. 2004. Functional proteomics mapping of a human signaling pathway. *Genome Research*, 14(7):1324–1332.

K. Franzen, G. Eriksson, F. Olsson, L. Asker, P. Liden, and J. Coster. 2002. Protein names and how to find them. *International Journal of Medical Informatics*, 67(1-3):49–61.

L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu. 2002. Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, 18(12):1553–1561.

R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. 2003. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453.

T. K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28.

G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, and et al. 2005. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33 Database Issue:D428–432.

M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. 2004. The kegg resource for deciphering the genome. *Nucleic Acids Research*, 32 Database issue:D277–280.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning (ICML-2001)*, pages 282–289, Williamstown, MA.

I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. 2004. A probabilistic functional network of yeast genes. *Science*, 306(5701):1555–1558.

B. Lehner and A. G. Fraser. 2004. A first-draft human protein-interaction map. *Genome Biology*, 5(9):R63.

H. Liu and L. Wong. 2003. Data mining tools for biological sequences. *Journal of Bioinformatics and Computational Biology*, 1(1):139–167.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

E. M. Marcotte, I. Xenarios, and D. Eisenberg. 2001. Mining literature for protein-protein interactions. *Bioinformatics*, 17(4):359–363.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

S. Peri, J. D. Navarro, T. Z. Kristiansen, R. Amanchy, V. Surendranath, B. Muthusamy, T. K. Gandhi, K. N. Chandrika, N. Deshpande, S. Suresh, and et al. 2004. Human protein reference database as a discovery resource for proteomics. *Nucleic Acids Research*, 32 Database issue:D497–501.

A. K. Ramani, R. C. Bunescu, R. J. Mooney, and E. M. Marcotte. 2005. Consolidating the set of know human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biology*, 6(5):r40.

A. Rzhetsky, I. Iossifov, T. Koike, M. Krauthammer, P. Kra, M. Morris, H. Yu, P. A. Duboue, W. Weng, W. J. Wilbur, V. Hatzivassiloglou, and C. Friedman. 2004. Geneways: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *Journal of Biomedical Informatics*, 37(1):43–53.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.

C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. 2002. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403.

I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg. 2002. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305.

IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text

Syed Toufeeq Ahmed, Deepthi Chidambaram, Hasan Davulcu*, Chitta Baral.
Department of Computer Science and Engineering,
Arizona State University,
Tempe, AZ 85287.
{toufeeq, deepthi, hdavulcu, chitta}@asu.edu

Abstract

In this paper, we present a fully automated extraction system, named IntEx, to identify gene and protein interactions in biomedical text. Our approach is based on first splitting complex sentences into simple clausal structures made up of syntactic roles. Then, tagging biological entities with the help of biomedical and linguistic ontologies. Finally, extracting complete interactions by analyzing the matching contents of syntactic roles and their linguistically significant combinations. Our extraction system handles complex sentences and extracts multiple and nested interactions specified in a sentence. Experimental evaluations with two other state of the art extraction systems indicate that the IntEx system achieves better performance without the labor intensive pattern engineering requirement.

1 Introduction

Genomic research in the last decade has resulted in the production of a large amount of data in the form of micro-array experiments, sequence information and publications discussing the discoveries. The data generated by these experiments is highly connected; the results from sequence analysis and micro-arrays depend on functional information and signal transduction pathways cited in peer-reviewed publications for evidence. Though scientists in the field are aided by many online databases of biochemical interactions, currently a majority of these are curated labor intensively by domain experts. Information extraction from text has therefore been pursued actively as an attempt to extract knowledge from published material and to speed up the curation process significantly.

In the biomedical context, the first step towards information extraction is to recognize the names of proteins (Fukuda, Tsunoda et al. 1998), genes, drugs and other molecules. The next step is to recognize interaction events between such entities (Blaschke, Andrade et al. 1999; Blaschke, Andrade et al. 1999; Hunter 2000; Thomas, Milward et al. 2000; Thomas, Rajah et al. 2000; Ono, Hishigaki et al. 2001; Hahn and Romacker 2002) and then to finally recognize the relationship between interaction events. However, several issues make extracting such interactions and relationships difficult since (Seymore, McCallum et al.1999) (i) the task involves free text – hence there are many ways of stating the same fact (ii) the genre of text is not grammatically simple (iii) the text includes a lot of technical terminology unfamiliar to existing natural language processing systems (iv) information may need to be combined across several sentences, and (v) there are many sentences from which nothing should be extracted.

In this paper, we present a fully automated extraction approach to identify gene and protein interact-

* To whom correspondence should be addressed

Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pages 54–61, Detroit, June 2005.

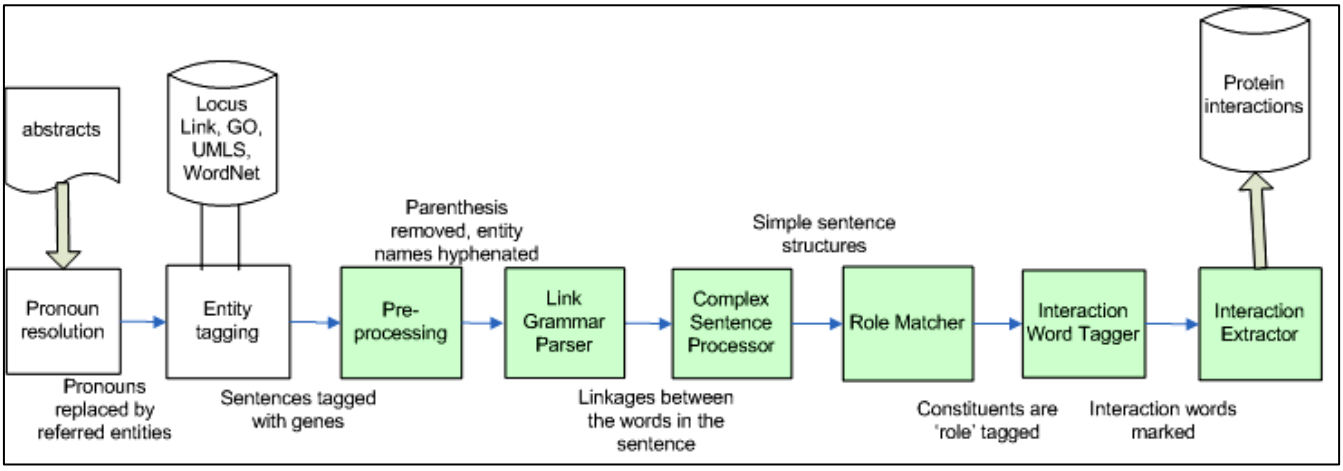


Figure 1: System Architecture

tions in natural language text with the help of biomedical and linguistic ontologies. Our approach works in three main stages:

1. **Complex Sentence Processor (CSP):** First, is splitting complex sentences into simple clausal structures made of up syntactic roles.
2. **Tagging:** Then, tagging biological entities with the help of biomedical and linguistic ontologies.
3. **Interaction Extractor:** Finally, extracting complete interactions by analyzing the matching contents of syntactic roles and their linguistically significant combinations.

The novel aspects of our system are its ability to handle complex sentence structures using the Complex Sentence Processor (CSP) and to extract multiple and nested interactions specified in a sentence using the Interaction Extractor without the labor intensive pattern engineering requirement. Our approach is based on identification of syntactic roles, such as subject, objects, verb and modifiers, by using the word dependencies. We have used a dependency based English grammar parser, the Link Grammar (Sleator and Temperley 1993), to identify the roles. Syntactic roles are utilized to transform complex sentences into their multiple clauses each containing a single event. This clausal structure enables us to engineer an automated algorithm for the extraction of events thus overcoming the burden of labor intensive pattern engineering for complex and compound sentences. Pronoun resolution module assists Interaction Extractor in identifying interactions spread across multiple sentences using pronominal references. We performed comparative experimental evaluations with two state of the art systems. Our experimental results show that the IntEx system presented here achieves better performance without the labor intensive rule engineering step which is required for these state of the art systems.

The rest of the paper is organized as follows. In Section 2 we survey the related work. In Section 3 we present an architectural overview of the IntEx system. Sections 4 and 5 explain and illustrate the individual modules of the IntEx system. A detailed evaluation of our system with the BioRAT (Corney, Buxton et al. 2004) and GeneWays (Rzhetsky, Iossifov et al. 2004) is presented in Section 6. Section 7 concludes the paper.

2 Related Work

Information extraction is the extraction of salient facts about pre-specified types of events, entities (Bunescu, Ge et al. 2003) or relationships from free text. Information extraction from free-text utilizes shallow-parsing techniques (Daelemans, Buchholz et al. 1999), Parts-of-Speech tagging(Brill 1992), noun and verb phrase chunking (Mikheev and Finch 1997), verb subject and object relationships (Daelemans, Buchholz et al. 1999), and learned (Califf and Mooney 1998; Craven and Kumlein 1999; Seymore, McCallum et al. 1999) or hand-build patterns to automate the creation of specialized databases.

Manual pattern engineering approaches employ shallow parsing with patterns to extract the interactions. In the (Ono, Hishigaki et al. 2001) system,

sentences are first tagged using a dictionary based protein name identifier and then processed by a module which extracts interactions directly from complex and compound sentences using regular expressions based on part of speech tags.

The SUISEKI system of Blaschke (Blaschke, Andrade et al. 1999) also uses regular expressions, with probabilities that reflect the experimental accuracy of each pattern to extract interactions into predefined frame structures.

GENIES (Friedman, Kra et al. 2001) utilizes a grammar based NLP engine for information extraction. Recently, it has been extended as GeneWays (Rzhetsky, Iossifov et al. 2004), which also provides a Web interface that allows users to search and submit papers of interest for analysis. The BioRAT system (Corney, Buxton et al. 2004) uses manually engineered templates that combine lexical and semantic information to identify protein interactions. The GeneScene system(Leroy, Chen et al. 2003) extracts interactions using frequent preposition-based templates.

Grammar engineering approaches, on the other hand use manually generated specialized grammar rules (Rinaldi, Schneider et al. 2004) that perform a deep parse of the sentences. Temkin (Temkin and Gilder 2003) addresses the problem of extracting protein interactions by using an extendable but manually built Context Free Grammar (CFG) that is designed specifically for parsing biological text. The PathwayAssist system uses an NLP system, MedScan (Novichkova, Egorov et al. 2003), for the biomedical domain that tags the entities in text and produces a semantic tree. Slot filler type rules are engineered based on the semantic tree representation to extract relationships from text. Recently, extraction systems have also used link grammar (Grinberg, Lafferty et al. 1995) to identify interactions between proteins (Ding, Berleant et al. 2003). Their approach relies on various linkage paths between named entities such as gene and protein names. Such manual pattern engineering approaches for information extraction are very hard to scale up to large document collections since they require labor-intensive and skill-dependent pattern engineering.

Machine learning approaches have also been used to learn extraction rules from user tagged training data. These approaches represent the rules learnt in various formats such as decision trees (Chiang, Yu et al. 2004) or grammar rules (Phuong, Lee et al. 2003). Craven et al (Craven and Kumlien 1999) explored an automatic rule-learning approach that uses a combination of FOIL (Quinlan 1990) and Naïve Bayes Classifier to learn extraction rules.

3 System Architecture

The sentences in English are classified as either simple, complex, compound or complex-compound based on the number and types of clauses present in them. Our extraction system resolves the complex, compound and complex-compound sentence structures (collectively referred to as complex sentence structures in this document) into simple sentence clauses which contain a subject and a predicate. These simple sentence clauses are then processed to obtain the interactions between proteins. The architecture of the IntEx system is shown in Figure 1, and the following Sections 4 and 5 explain the workings of its modules.

4 Complex Sentence Processing

4.1 Pronoun Resolution

Interactions are often specified through pronominal references to entities in the discourse, or through co references where, a number of phrases are used to refer to the same entity. Hence, a complete approach to extracting information from text should also take into account the resolution of these references. References to entities are generally categorized as co-references or anaphora and has been investigated using various approaches (Castaño, Zhang et al. 2002). IntEx anaphora resolution subsystem currently focuses on third person pronouns and reflexives since the first and second person pronouns are frequently used to refer to the authors of the papers.

Our pronoun resolution module uses a heuristic approach to identify the noun phrases referred by the pronouns in a sentence. The heuristic is based on the number of the pronoun (singular or plural) and the proximity of the noun phrase. The first noun phrase that matches the number of the pronoun is considered as the referred phrase.

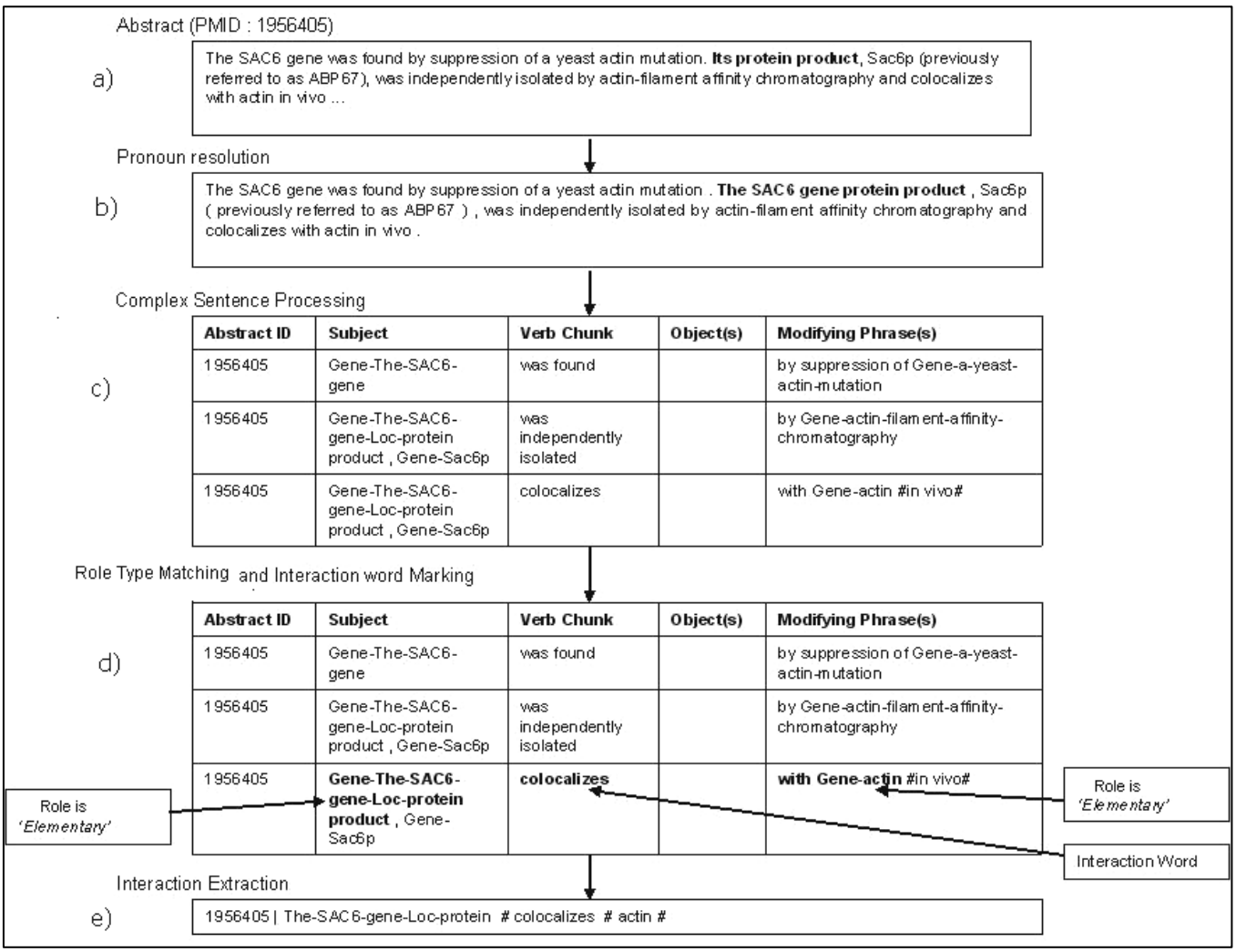


Fig. 3 Example - a) A Sentence from an abstract (PMID: 1956405). b) Pronoun 'it's' is resolved with 'The SAC6 gene'. c) Each row represents a simple sentence, d) for each constituent, role type is resolved and interaction words are tagged, e) Protein-Protein interaction is extracted.

4.2 Entity Tagger

The entity tagging module marks the names of genes, and proteins in text. The process of tagging is a combination of dictionary look up and heuristics. Regular expressions are also used to mark the names that do not have a match in the dictionaries. The protein name dictionaries for the entity tagger are derived from various biological sources such as UMLS[1], Gene Ontology[2] and Locuslink[3] database

4.3 Preprocessor

The tagged sentences need to be pre-processed to replace syntactic constructs, such as parenthesized nouns and domain specific terminology that cause the Link Grammar Parser to produce an incorrect output. This problem is overcome by replacing such elements with alternative formats that is recognizable by the parser.

4.4 Link Grammar and the Link grammar parser

Link grammar (LG) introduced by Sleator and Temperley (Sleator and Temperley 1991) is a dependency based grammatical system. The basic idea of link grammar is to connect pairs of words

[1] http://www.nlm.nih.gov/research/umls/
[2] http://www.geneontology.org/
[3] http://www.ncbi.nlm.nih.gov/LocusLink/

in a sentence with various syntactically significant links. The LG consists of set of words, each of which has various alternative linking requirements.

A linking requirement can be seen as a block with connectors above each word. A connector is satisfied by matching it with compatible connector. Fig.2 below shows how linking requirements can be satisfied to produce a parse for the example sentence "*The dog chased a cat*".

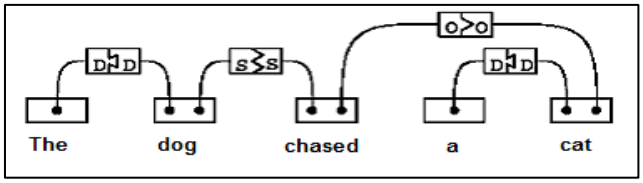


Figure 2: Link grammar representation of a sentence

Even though LG has no explicit notion of constituents or categories (Sleator and Temperley 1993), they emerge as contiguous connected sequence of words attached to the rest of sentence by a particular types of links, as in the above example where *'the dog'* and *'a cat'* are connected to the main verb via 'S' and 'O' links respectively. Our algorithms utilize this property of LG where certain link types allow us to extract the constituents of sentences irrespective of the tense. The LG parser's ability to detect multiple verbs and their constituent linkage in complex sentences makes it particularly well suited for our approach during resolving of complex sentences into their multiple clauses. The LG parsers' dictionary can also be easily enhanced to produce better parses for biomedical text (Szolovits 2003).

4.5 Complex Sentence Processor Algorithm

The complex sentence processor (CSP) component splitsthe complex sentences into a collection of simple sentence clauses which contain a subject and a predicate. The CSP follows a verb-based approach to extract the simple clauses. A sentence is identified to be complex it contains more than one verb. A simple sentence is identified to be one with a subject, a verb, objects and their modifying phrases. The example in Figure 3 illustrates the major steps involved during complex sentence processing. The following schema is used as the format to represent simple clauses:

Subject | Verb | Object | Modifying phrase to the verb

5 Interaction Extraction

Interaction Extractor (IE) extracts interactions from simple sentence clauses produced by the complex sentence processor. The highly technical terminology and the complex grammatical constructs that are present in the biomedical abstracts make the extraction task difficult, Even a simple sentence with a single verb can contain multiple and/or nested interactions. That's why our IE system is based on a deep parse tree structure presented by the LG and it considers a thorough case based analysis of contents of various syntactic roles of the sentences like their subjects (S), verbs (V), objects (O) and modifying phrases (M) as well as their linguistically significant and meaningful combinations like *S-V-O, S-O, S-V-M or S-M*, illustrated in Figure 4, for finding and extracting protein-protein interactions.

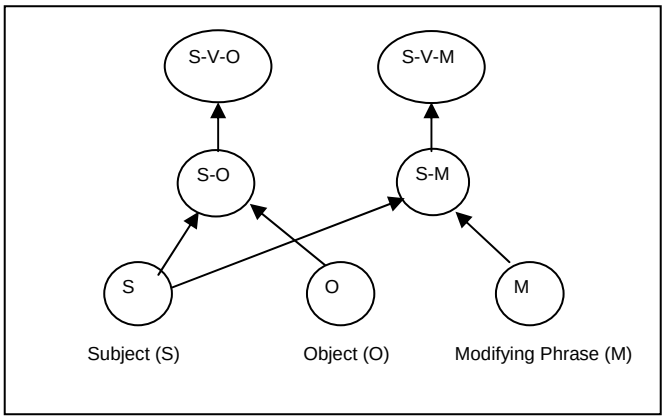


Figure 4: Interaction Extraction: Composition and analysis of various syntactic roles.

5.1 Role Type Matcher

For each syntactic constituent of the sentence, the role type matcher identifies the type of each role as either 'Elementary', 'Partial' or 'Complete' based on its matching content, as presented in Table 1.

Table 1: Role Type Matcher

Role Type	Description
Elementary	If the role contains a Protein name or an interaction word.
Partial	If the role has a Protein name and an interaction word.
Complete	If the role has at least two Protein names and an interaction word.

5.2 Interaction Word Tagger

The words that match a biologically significant action between two gene/protein names are labeled as 'interaction words'. Our gazetteer for interaction words is derived from UMLS and WordNet[4]. Porter Stemmer (Porter 1997) was also used for stemming such words before matching.

5.3 Interaction Extractor (IE)

IntEx interaction extractor works as follows. The input to IE is the preprocessed and typed simple clause structures. The IE algorithm progresses bottom up, starting from each syntactic role S, V or M, and expanding them using the lattice provided in Figure 4 until all 'Complete' singleton or composite role types are obtained.

Consider the example shown in Figure 3, for the third sentence, the boundaries of the subject and the modifying phrase are identified and both are role typed as 'Elementary' using Table 1. Since the main verb is tagged as an interaction word, IE uses the S-V-M composite role from Figure 4 to find and extract the following complete interaction:

{'The SAC 6 gene Protein', 'colocalizes', 'actin'}.

'Complete' roles also need to be analyzed in order to determine their voice as 'active' or 'passive'. Since there are only a small number of preposition combinations, such as *of-by, from-to etc.,* that occur frequently within the clauses, they can be used to distinguish the agent and the theme of the interactions.

For example, in the sentence "The kinase phosphorylation of pRb by c-Abl in the gland could inhibit ku70", the subject role is "The kinase phosphorylation of pRb by c-Abl in the gland". Since the subject has at least two protein names and an interaction word it is 'complete'. By using the *'of-by'* pattern (…<*Interaction-Word (action)*>... ***of*** ...<*theme*>…***by*** ...<*agent*>…) the IE is able to extract the correct interaction {c-Abl, phosphorylation, pRb} from the subject role alone.

[4] http://www.cogsci.princeton.edu/~wn/

6 Evaluation & discussion

We have evaluated the performance of our system with two state of the art systems - BioRAT (Corney, Buxton et al. 2004) and GeneWays (Rzhetsky, Iossifov et al. 2004).

Blaschke and Valencia (Valencia 2001) recommend DIP (Xenarios, Rice et al. 2000) dataset as a benchmark for evaluating biomedical Information Extraction systems. The first evaluation for IntEx system was performed on the same dataset [5] that was used for the BioRAT evaluation. For BioRAT evaluation, authors identified 389 interactions from the DIP database such that both proteins participating in the interaction had SwissProt entries. These interactions correspond to 229 abstracts from the PubMed. The BioRAT system was evaluated using these 229 abstracts. The interactions extracted by the system were then manually examined by a domain expert for precision and recall. Precision is a measure of correctness of the system, and is calculated as the ratio of true positives to the sum of true positives and false positives. The sensitivity of the system is given by the recall measure, calculated as the ratio of true positives to the sum of true positives and false negatives.

Table 2: Recall comparison of IntEx and BioRAT from 229 abstracts when compared with DIP database.

Recall Results	IntEx		BioRAT	
	Cases	Percent (%)	Cases	Percent(%)
Match	142	**26.94**	79	**20.31**
No Match	385	73.06	310	79.67
Totals	527	100.00	389	100.00

We have also limited our protein name dictionary to the SwissProt entries. Tables 2 and 3 present the evaluation results as compared with the BioRAT system. A detailed analysis of the sources of all types of errors is shown in Figure 6.

[5] Dataset was obtained from Dr. David Corney by personal communication.

Table 3: Precision comparison of IntEx and BioRAT from 229 abstracts.

Precision Results	IntEx		BioRAT	
	Cases	Percent (%)	Cases	Percent (%)
Correct	262	**65.66**	239	**55.07**
Incorrect	137	34.33	195	44.93
Totals	399	100.00	434	100.00

DIP contains protein interactions from both abstracts and full text. Since our extraction system was tested only on the abstracts, the system missed out on some interactions that were only present in the full text of the abstract.

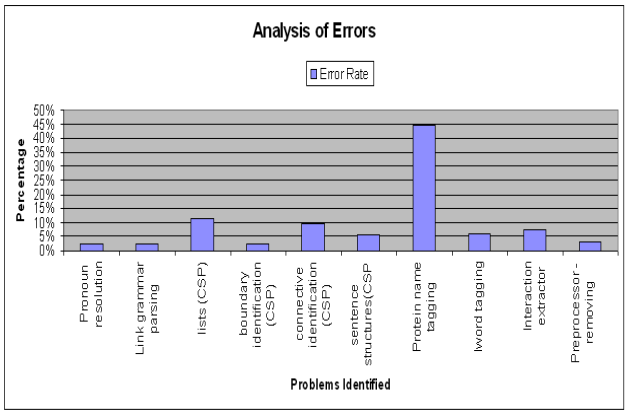


Figure 6: Analysis of types of errors encountered

Second evaluation for the IntEx system was done to test its recall performance using an article[6] that was also used by the GeneWays (Rzhetsky, Iossifov et al. 2004) system. Both systems performance was tested using the full text of the article (Friedman, Kra et al. 2001). GeneWays system achieves a recall of 65% where as IntEx extracted a total of 44 interactions corresponding to a recall measure of 66 %.

Conclusion

In this paper, we present a fully automated extraction system for identifying gene and protein interactions from biomedical text. The source code and documentation of the IntEx system, as well as all experimental documents and extracted interactions are available online at our Web site at http://cips.eas.asu.edu/textmining.htm. Our extraction system handles complex sentences and extracts multiple nested interactions specified in a sentence. Experimental evaluations of the IntEx system with the state of the art semi-automated systems -- the BioRAT and GeneWays datasets indicates that our system performs better without the labor intensive rule engineering requirement. We have shown that a syntactic role-based approach compounded with linguistically sound interpretation rules applied on the full sentence's parse can achieve better performance than existing systems which are based on manually engineered patterns which are both costly to develop and are not as scalable as the automated mechanisms presented in this paper.

[6] Dataset was obtained from Dr. Andrew Rzhetsky by personal communication.

Acknowledgements

We would like to thank to both Dr. David Corney and Dr. Andrew Rzhetsky for sharing their evaluation datasets and results.

References

Blaschke, C., M. A. Andrade, et al. (1999). "Automatic extraction of biological information from scientific text: Protein-protein interactions." Proceedings of International Symposium on Molecular Biology: 60-67.

Brill, E. (1992). "A simple rule-based part-of-speech tagger." Proceedings of ANLP 92, 3rd Conference on Applied Natural Language Processing: 152-155.

Bunescu, R., R. Ge, et al. (2003). Comparative Experiments on Learning Information Extractors for Proteins and their Interactions. Artificial Intelligence in Medicine.

Califf, M. E. and R. J. Mooney (1998). "Relational learning of pattern-match rules for information extraction." Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing: 6-11.

Castaño, J., J. Zhang, et al. (2002). Anaphora Resolution in Biomedical Literature. International Symposium on Reference Resolution. Alicante, Spain.

Chiang, J.-H., H.-C. Yu, et al. (2004). "GIS: a biomedical text-mining system for gene information discovery." Bioinformatics 20(1): 120-121.

Corney, D. P. A., B. F. Buxton, et al. (2004). "BioRAT: extracting biological information from full-length papers." Bioinformatics 20(17): 3206-3213.

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

Daelemans, W., S. Buchholz, et al. (1999). "Memory-based shallow parsing." Proceedings of CoNLL.

Ding, J., D. Berleant, et al. (2003). Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03): 467.

Friedman, C., P. Kra, et al. (2001). GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Proceedings of the International Confernce on Intelligent Systems for Molecular Biology: 574-82.

Fukuda, K., T. Tsunoda, et al. (1998). "Toward information extraction: Identifying protein names from biological papers." PSB 1998,: 705-716.

Grinberg, D., J. Lafferty, et al. (1995). "A Robust Parsing Algorithm For LINK Grammars." (CMU-CS-TR-95-125).

Hahn, U. and M. Romacker (2002). "Creating knowledge repositories from biomedical reports: The medsyndikate text mining system." Pacific Symposium on Biocomputing 2002: 338-349.

Hunter, R. T. a. C. R. a. J. (2000). "Extracting Molecular Binding Relationships from Biomedical Text." In Proceedings of the ANLP-NAACL 000,Association for Computational Linguistics: pages 188-195.

Leroy, G., H. Chen, et al. (2003). Genescene: biomedical text and data mining. Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries: 116--118.

Mikheev, A. and S. Finch (1997). "A workbench for finding structure in texts." Proceedings of Applied Natural Language Processing (ANLP-97).

Novichkova, S., S. Egorov, et al. (2003). "MedScan, a natural language processing engine for MEDLINE abstracts." Bioinformatics 19(13): 1699-1706.

Ono, T., H. Hishigaki, et al. (2001). "Automatic Extraction of Information on protein-protein interactions from the biological literature." Bioinformatics 17(2): 155-161.

Phuong, T. M., D. Lee, et al. (2003). "Learning Rules to Extract Protein Interactions from Biomedical Text." PAKDD 2003: 148-158.

Porter, M. F. (1997). "An algorithm for suffix stripping." Progam, vol. 14, no. 3, July 1980: 313--316.

Quinlan, J. R. (1990). "Learning Logical Definitions from Relations." Mach. Learn. 5(3): 239--266.

Rinaldi, F., G. Schneider, et al. (2004). Mining relations in the GENIA corpus. Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics: 61 - 68.

Rzhetsky, A., I. Iossifov, et al. (2004). "GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data." J. of Biomedical Informatics 37(1): 43--53.

Seymore, K., A. McCallum, et al. (1999). Learning hidden markov model structure for information extraction. AAAI 99 Workshop on Machine Learning for Information Extraction.

Sleator, D. and D. Temperley (1991). Parsing English with a Link Grammar.Carnegie Mellon University Computer Science technical report CMU-CS-91-196, Carnegie Mellon University.

Sleator, D. and D. Temperley (1993). Parsing English with a Link Grammar. Third International Workshop on Parsing Technologies.

Szolovits, P. (2003). "Adding a Medical Lexicon to an English Parser." Proc. AMIA 2003 Annual Symposium.: 639-643.

Temkin, J. M. and M. R. Gilder (2003). "Extraction of protein interaction information from unstructured text using a context-free grammar." Bioinformatics 19(16): 2046-2053.

Thomas, J., D. Milward, et al. (2000). "Automatic extraction of protein interactions from scientific abstracts." Proceedings of the Pacific Symposium on Biocomputing 5: 502-513.

Thomas, R., C. Rajah, et al. (2000). "Extracting molecular binding relationships from biomedical text." Proceedings of the ANLP-NAACL 2000: 188-195.

Valencia, B. a. (2001). "Can bibliographic pointers for known biological data be found automatically? Protein interactions as a case study." Comp. Funct. Genom. 2: 196-206.

Xenarios, I., D. W. Rice, et al. (2000). "DIP: the Database of Interacting Proteins." Nucl. Acids Res. 28(1): 289-291.

Author Index