LEARNING TO EXTRACT GENE-PROTEIN NAMES FROM WEAKLY-LABELED TEXT

RICHARD C. WANG[1], ANTHONY TOMASIC[2],
ROBERT E. FREDERKING[1], WILLIAM W. COHEN[3]

[1]*Language Technologies Institute*
[2]*Institute for Software Research International*
[3]*Machine Learning Department*

Carnegie Mellon University, 5000 Forbes Ave.
Pittsburgh, PA 15213, U.S.A.

Training a named entity recognizer (NER) has always been a difficult task due to the effort required to generate a significant amount of annotated training data. In this paper, we reduce or eliminate the effort required to create training data by automatically converting other sources of data into annotated training data. The performance of this approach is tested on a gene-protein name extractor by using the mouse and fly data obtained from the BioCreAtIvE challenge. Results show that our methods are effective and that our trained NER system outperforms all of our baseline results.

1. Introduction

Many prior research papers on biological text-mining have developed machine-learned *named entity recognition* (NER) systems to identify substrings in biomedical publications that correspond to gene and protein names, usually without distinguishing between them [3, 9, 11, 16, 23]. These NER systems are often trained on large amounts of manually annotated training examples, consisting of documents with the position of every named entity marked. This training data is difficult to produce.

In this paper, we explore several approaches for training a gene-protein NER system with data sources that are easier to obtain. One of these sources is a *synonym list* – a list of gene identifiers together with synonyms for each identifier. Another source is *weak labels*, which associate a document with identifiers for each gene-protein entity that appears in the document. The third source is NER annotations for a related, but slightly different corpus: this reflects the common practice of applying a learned NER system to documents that are drawn from a slightly different distribution.

One prior experimental study that exploits synonym lists and weak labels is BioCreAtIvE task 1B [14, 15], which collected common test-bed problems and a common evaluation framework for determining the identifier of every gene mentioned in biomedical abstracts. Three separate test-bed problems were developed, one for each of three model organisms: yeast, fly, and mouse.

In this paper, we utilize only the mouse and fly datasets, which were the two hardest for the BioCreAtIvE participants, for training a gene-protein NER system. As comparisons, we present results for four baseline systems. The first one is a dictionary-based extractor, which soft-matches words from a synonym list to a corpus. The second, third, and fourth baselines are machine-learned NER systems trained on the GENIA[1] [7] dataset, the YAPEX[2] [10] dataset, and small corpus of conventionally-labeled documents from the BioCreAtive datasets. We show that no baseline system performs well. We then present results for several alternative approaches that use weak labels.

Our approach for weak-label learning consists of four steps. First, we look up, for each abstract, its associated gene identifiers and we label all possible locations of synonyms associated with these identifiers in that same abstract. Second, we train extractors on these weakly labeled abstracts, using word features such as string similarity to synonyms [24]. We also investigated a pre-processing step, of removing from the training set sentences not containing any weak labels; and a post-processing step that exploits inter-document repetition of names [20] by soft-matching every instance of an extracted name against the document in which it occurs, and classifying every such soft-match as a protein name. To further evaluate our weak-label learning approach, we present also results for NER systems tuned for either precision or recall [21]. Our results show that the quality of a NER system can be improved through the use of readily available weakly-labeled data.

We use datasets from BioCreAtIvE task 1B, specifically the mouse and fly datasets, which were drawn from MGI [2] and Flybase [1] respectively.

Table 1. Distribution of abstracts among various data for each of mouse and fly dataset. Numbers embraced by brackets indicate a subset of these numbers.

Dataset	**Mouse**			**Fly**		
Data	**Eval.**	**Weak-train**	**Curated**	**Eval.**	**Weak-train**	**Curated**
# of Abstracts	50	200	1000	51	57	1000
Abstract IDs	100-149	1-99, 150-250	4000-4999	[1-298]	[308-494]	4000-4999

For each dataset, we constructed three corpora for our experiments: *evaluation*, *weak-train*, and *curated*. The *evaluation* and *weak-train* data are subsets of the BioCreAtIvE "devtest" set, and *curated* data is a subset of the "training" set. Table 1 summarizes, for both datasets, their size, and also lists the specific abstracts (by BioCreAtIvE ID) that were used to form the dataset. In *curated*, each abstract is associated with gene identifiers of all genes that are mentioned in the full text of the abstract. However, in *weak-label*, each abstract

[1] Available from http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA
[2] Available from http://www.sics.se/humle/projects/prothalt

is only associated with identifiers of some genes mentioned in the abstract. Hence, *curated* is noisier than *weak-label*. We also utilize the synonym lists provided by the mouse and fly datasets, which contain associations between synonyms and unique gene identifiers. The list for the mouse dataset consists of 183,142 synonyms for 52,594 identifiers, and for the fly dataset, 135,471 synonyms for 35,970 identifiers. To evaluate our NER systems, the abstracts in the evaluation data were manually annotated with gene-protein entity names.

2. Baseline Methods

2.1. *Global Edit Distance*

In order to train a gene-protein NER system using a synonym list, we devise a feature that indicates how similar each word in the abstracts is to the most similar word in the entire (global) synonym list. The similarity measure incorporates Levenshtein Distance [19], and thus we call this the *global edit distance* (GED) feature. More specifically, it case-insensitively calculates a similarity score between two strings, *s* and *s'*, as:

$$SimScore(s,s') = 1 - \frac{LD(s,s')}{\max(length(s),length(s'))} \quad (1)$$

where $LD(s, s')$ is the Levenshtein Distance between string *s* and *s'*, and $length(s)$ is the number of characters in *s*. We determine and assign similarity scores to each word in the abstracts by traversing through each synonym in a given list. For each synonym *s*, we determine number of words *n* contained in *s*, and create sliding windows of size ranging from $\lceil 0.5n \rceil$ to $\lfloor 1.5n \rfloor$ on the abstract. For each string *s'* contained within each sliding window, we assign $SimScore(s, s')$ to each word *w* in *s'* unless one of the following conditions is met: a) *w* has higher similarity to some other *s''* in the synonym list, b) *s* or *s'* has only one character, c) *s* or *s'* case-insensitively matches any word in a list of common stop-words (see Appendix A), or d) the first and last characters of *s* are not identical to those of *s'*.

2.2. *Soft Matching*

Biological scientists often use novel variations of existing gene names in their papers; thus, in order to match these names from abstracts to the synonym list, we incorporate an approximate string matching technique called *soft matching*, which identifies strings that are similar but not necessarily identical. This method has been proven to be useful [13]; however, our method is on the character-level instead of word-level. We perform soft matching by first

assigning similarity scores to words in given abstracts using a given synonym list, as described in 3.1. We then label all the longest consecutive sequences of words that have similarity scores above a given similarity threshold as a gene-protein entity name.

2.3. *NER on YAPEX & GENIA*

We use an off-the-shelf machine learning system for NER called Minorthird [6] for training our gene-protein NER system on the YAPEX and GENIA corpora. We used Minorthird's default feature set, which contains basic features such as word identity and capitalization patterns. In addition, we used Minorthird's implementation of VP-HMM – a voted-perceptron based training scheme for HMMs due to Collins [8]. More specifically, as we configured this learner, NER is reduced to the problem of classifying each token as the *beginning* or *continuation* of a multi-token gene-protein name; or as *outside* of any gene-protein name. We configured the extractor to make 20 passes (epochs) over the training data using, and to use a window size of three words.

The YAPEX dataset consists of a training corpus of 99 Medline abstracts and a testing corpus of 101 Medline abstracts. These documents deal primarily with protein-protein interactions, and are annotated for gene-protein entities. We trained a VP-HMM extractor on the training corpus of YAPEX using Minorthird's default features. The GENIA dataset consists of a training corpus of 500 Medline abstracts and a testing corpus of 300 Medline abstracts, mostly concerning cell signaling for human cells. We trained a VP-HMM extractor on the training corpus of GENIA using default features, plus protein-specific features described elsewhere [18].

2.4. *Single Document Repetition*

When a substring is identified as a named entity in a document, it is highly possible that all other occurrences of that substring in the same document are also named entities. Repetition of names in text has proven useful on many occasions [4, 17, 20, 25]. We incorporate a post-processing step that exploits repetition of entity names within a single document using the gene-protein names extracted by our trained NER systems. More specifically, for each abstract, it collects all the extracted names from that abstract, and soft-matches these names against the words in the same abstract, using a constant threshold of 0.5 throughout our experiments; we refer to this as *single document repetition* (SDR) labeling.

3. Approach

3.1. *Grounding Weak Labels*

In the BioCreAtIvE challenge, one unique characteristic of the datasets is that there are synonym lists and weak labels. Therefore, for each abstract, we can approximately locate gene names by soft-matching synonyms of identifiers associated with that abstract against the words in the same abstracts. For this process, we used a similarity threshold of 0.50 for the mouse and 0.65 for the fly datasets; we will refer to this process as *grounding* the weak labels. A preliminary evaluation of our grounding method on the evaluation data shows an entity-level precision of 81%, recall of 65%, and F_1 of 72% for the mouse dataset, and precision of 73%, recall of 70%, and F_1 of 71% for the fly dataset.

3.2. *Sentence Filtering*

Abstracts with grounded gene-protein synonyms are training examples that are weakly-labeled. Often genes that are mentioned but not associated with new results are not curated; thus, they become false negatives in our training set. One method for enhancing the quality of the training set is by eliminating false negatives. We incorporate a pre-processing step of filtering training examples that may be false negatives: specifically, we split abstracts into sentences using regular expressions, and then remove those sentences in the training data that do not contain any grounded gene-protein synonyms. We call this the *sentence filtering* process. Recently, the same sentence filtering technique was independently described by Vlachos and Gasperin [26].

4. Experiments

4.1. *Settings*

We trained a VP-HMM extractor on each of the following three datasets: *weak-train*, *curated*, and a combined set, *merged*; all weakly-labeled with grounded gene-protein synonyms using the approach described in 3.1. Each trained extractor is evaluated with various combinations of sentence filtering, SDR labeling, and GED features. These extractors are evaluated on the evaluation data at the entity-level (i.e., no partial credit is given for nearly-correct entity boundaries). We compare our NER system's performance to four baselines: a) an extractor trained on YAPEX, b) another trained on GENIA, c) 10-fold cross validation on the evaluation data, and d) a global dictionary soft-matcher which soft-matches every synonym from an entire synonym list to the evaluation data, with similarity thresholds set to a high 0.85 for mouse dataset and 0.95 for fly

dataset. These thresholds were pre-determined to optimize F_1 measure on the evaluation data (and specifically, exact-matching performed worse).

4.2. *Results*

Table 2. Performance of the four baselines and our NER systems at *entity-level* (denoted by **E.**) tested on the mouse evaluation data. Bold F_1 scores represent scores that are higher than any corresponding baseline. Extractors denoted by * will be tuned for section 6.

			-SDR			+SDR		
			E. Prec.	**E. Recall**	**E. F1**	**E. Prec.**	**E. Recall**	**E. F1**
		YAPEX	68.36	27.56	39.29	69.28	48.29	56.91
		GENIA	66.46	24.37	35.67	67.45	39.18	49.57
		Dict.	50.34	67.43	57.64	47.56	66.51	55.46
	C.V. eval.	**-GED**	54.81	29.84	38.64	49.28	38.95	43.51
		+GED	59.05	53.53	56.15	54.75	60.36	57.42
	Weak-train	**-GED**	82.39	26.65	40.28	78.76	34.62	48.10
		+GED	78.47	48.97	**60.31**	75.58	59.23	**66.41**
Curated	**-GED**	**-Filter**	90.82	20.27	33.15	90.96	34.40	49.92
		+Filter	74.67	38.27	50.60	71.97	60.82	**65.93**
	+GED	**-Filter**	87.83	46.01	**60.39**	83.59	61.50	**70.87**
		+Filter	80.91	56.95	**66.84**	76.10	66.74	**71.12**
Merged	**-GED**	**-Filter**	90.35	23.46	37.25	78.63	41.91	54.68
		+Filter*	78.30	41.91	54.60	73.10	61.28	**66.67**
	+GED	**-Filter**	87.40	50.57	**64.07**	84.13	60.36	**70.29**
		+Filter*	79.57	58.54	**67.45**	75.90	67.43	**71.41**

Table 3. Performance of the four baselines and our NER systems at *entity-level* (denoted by **E.**) tested on the fly evaluation data. Bold F_1 scores represent scores that are higher than any corresponding baseline. Extractors denoted by * will be tuned for section 6.

			-SDR			+SDR		
			E. Prec.	**E. Recall**	**E. F1**	**E. Prec.**	**E. Recall**	**E. F1**
		YAPEX	66.00	23.32	34.46	68.79	34.28	45.75
		GENIA	44.16	12.01	18.89	59.06	26.50	36.59
		Dict.	28.92	70.32	40.99	27.75	70.32	39.80
	C.V. eval.	**-GED**	39.13	9.54	15.34	46.38	22.61	30.40
		+GED	37.59	36.40	36.98	35.68	46.64	40.43
	Weak-train	**-GED**	37.50	3.18	5.86	38.46	7.07	11.94
		+GED	51.89	38.87	**44.44**	56.79	57.60	**57.19**
Curated	**-GED**	**-Filter**	78.43	28.27	**41.56**	75.71	47.35	**58.26**
		+Filter	63.64	44.52	**52.39**	55.25	57.60	**56.40**
	+GED	**-Filter**	73.19	60.78	**66.41**	64.31	64.31	**64.31**
		+Filter	65.73	66.43	**66.08**	57.82	69.26	**63.02**
Merged	**-GED**	**-Filter**	78.26	31.80	**45.23**	74.30	47.00	**57.58**
		+Filter*	64.47	44.88	**52.92**	55.70	58.66	**57.14**
	+GED	**-Filter**	70.76	59.01	**64.35**	62.46	64.66	**63.54**
		+Filter*	64.38	66.43	**65.39**	56.20	68.90	**61.90**

In addition to the four baseline performances, we present our NER systems performance at the entity-level in Table 2 and 3. For the mouse dataset, filtering, SDR, and GED are always effective, reaching a maximum F_1 measure of 71.4%

when all three methods are combined. However, for the fly dataset, only GED is always effective. SDR is effective when not combined with GED: we conjecture that when precision is high and recall is low, SDR is more likely to label false negatives than true negatives as gene-protein names. Filtering is most effective when not combined with SDR, because filtering removes false negatives. The maximum F_1 score obtained from the fly dataset is 66.4%. These results also indicate that the gene-protein synonyms in the fly dataset are more ambiguous. In the mouse dataset, the highest F_1 score among the baselines is about 57.5%. Our NER systems outperform this baseline when it was trained with GED, or when it was filtered and SDR labeled. The highest F_1 measure of the baselines in the fly dataset is about 41% without SDR and 46% with SDR, and our NER systems almost always outperform these baselines.

5. Extractor Tweaking

5.1. *Method*

Curators of biological databases sometimes prefer a high-recall gene-protein name extractor to assist them in identifying most gene-protein candidate names. To create such an extractor we tune or *tweak* [21] the threshold term of our trained extractors (marked with * in Table 2 and 3) on the word-level recall of the tuning data *weak-train* (less noisier than *curated*) with respect to a specific β value in the complete F-measure formula:

$$F_\beta(P,R) = \frac{(\beta^2+1)PR}{\beta^2 P + R} \tag{2}$$

Here P is word-level precision and R is word-level recall. A β value of greater than 1 assigns higher importance to recall; for instance, F_2 weights recall twice as much as precision. These tweaked extractors are evaluated on the evaluation data as well.

Use of word-level precision and recall rather than entity-level precision and recall gives some credit to nearly-correct entity boundaries – for instance, an extractor that extends slightly past an entity boundary will receive credit for word recall, but be penalized for word precision.

5.2. *Results*

In Fig. 1 (mouse) and 2 (fly), each shows two precision-recall curves on the word-level: one is a curve of tweaked extractors trained without GED features and the other with GED features. Each data point on a line represents an

extractor tweaked for a different β value (0.1, 0.2, …, 0.9, 1, 2, …, 10) trained on filtered examples and has extractions SDR labeled.

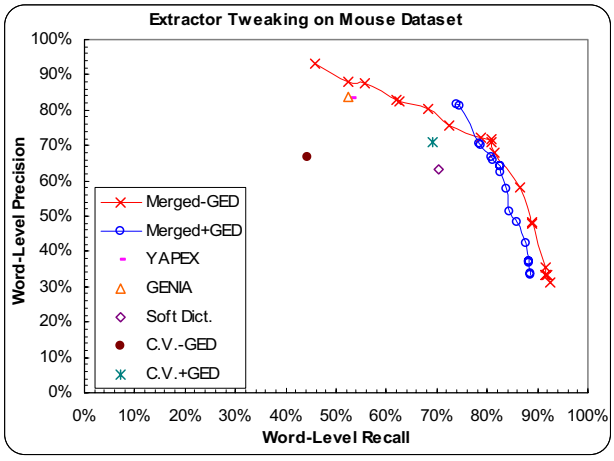

Figure 1. Tweaking extractors trained on the mouse dataset for β values from 0.1 to 10 on the *word-level* recall of *weak-train* data. The four baselines are also shown. *Merged* was filtered, and all extractions were SDR labeled.

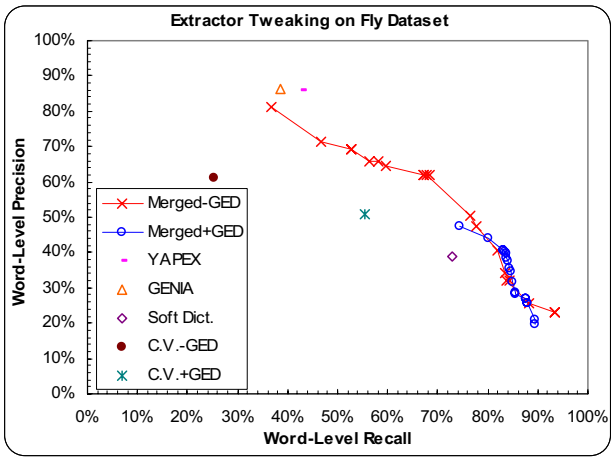

Figure 2. Tweaking extractors trained on the fly dataset for β values from 0.1 to 10 on the *word-level* recall of *weak-train* data. The four baselines are also shown. *Merged* was filtered, and all extractions were SDR labeled.

.

As comparisons, we also show the four baselines: YAPEX, GENIA, soft matching using dictionary, and 10-fold cross validation (with and without GED features), also on the word-level. The higher the β value, the higher the word-level recall of the tweaked extractor. We were able to generate a high-recall and medium-precision extractor, tweaked for $\beta = 3$ without GED features, that has a word-level precision, recall, and F_1 of about 58%, 87%, and 70% respectively for the mouse dataset and 48%, 78%, and 59% respectively for the fly dataset.

6. Related Work

The identification of gene-protein names has received substantial attention in the bioinformatics community. Some prior research involves training an extractor on weakly-labeled gene-protein synonyms. Hachey *et al.* [12] automatically labeled gene text fragments by identifying potential genes using regular expression fuzzy matching, and then trained a tagger for each organism. Morgan *et al.* [22] perform pattern matching to find candidate mentions in FlyBase abstracts using synonym lists and trained a HMM-based tagger on these noisy training data, achieving a F_1 of 67% with 522,825 tokens of training data and a F_1 of 75% with 1,342,039 tokens of training data. The additional contributions of this work are to study: the generality weak-labeling methods (our system is the same for FlyBase and MGI, except for thresholds); the use of intra-document repetition; the effect of sentence filtering; and the range of points reachable on a recall-precision curve. Morgan *et al.,* however, exploited orthographic preprocessing steps that we did not use, and the effect of using much larger training sets. Our F_1 performance for the fly data with 1057 abstracts is comparable to that obtained by Morgan *et al.* with 522,825 tokens (approximately 2000-2500 abstracts). Unfortunately we cannot compare directly on the same test set, due to technical issues involving tokenization.

Some other prior related research involves unsupervised identification of gene-protein names. Wellner [27] incorporates part-of-speech as factors for proposing gene phrases and performs exact matching from a synonym list to abstracts for annotating candidate gene-protein synonyms. Cohen [5] generates orthographic variants of gene-protein entities, separates out regular English words by using English word dictionaries, and matches the remaining variants against biomedical abstracts.

The contribution of this paper is to explore and systematically evaluate several different techniques, in isolation and in combination, for the gene-protein NER task: sentence filtering, GED features [24], SDR labeling [20], training on weakly-labeled examples [22], and tuning trained extractors [21]. We also contribute to the community, for each of fly and mouse organism, two

organism-specific gene-protein name extractors[3]; one has high precision but medium recall and the other high recall but medium precision.

7. Conclusions

In this paper, we trained a gene-protein NER system, without manually annotating any documents, by utilizing the mouse and fly dataset from BioCreAtIvE task 1B. We presented an automatic approach for creating training corpora by soft matching gene synonyms into abstracts. We illustrated that the NER systems trained on these annotated abstracts, combined with sentence filtering, SDR labeling, and/or GED features, can outperform all baselines. Furthermore, we also demonstrated the possibility of converting a gene-protein NER system with decent performance into a high-recall gene-protein name extractor. Our results demonstrate that the quality of named entity recognition systems can be significantly improved through the use of readily available data and thus avoiding the difficult process of manually annotating training sets.

Acknowledgements

The authors wish to thank Isaac Simmons for his help. This material is based upon work supported by supported by NIH K25 grant DA017357-0, and the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NIH, the Defense Advanced Research Projects Agency (DARPA), or the Department of Interior-National Business Center (DOI-NBC).

Appendix A.

List of common English words that are used as stop-words in our system: *all, an, and, are, as, at, between, but, by, can, for, from, has, in, into, is, it, less, likely, more, most, much, not, of, on, or, per, such, that, the, through, to, via, was, we, were, whereas, whole, with.*

[3] Available from http://www.rcwang.com/pub/GeneNER.tar.gz

References

1. *FlyBase: A database of the drosophila genome*. [cited 2006 May 1]; Available from: http://flybase.bio.indiana.edu.
2. *MGI: mouse genome informatics*. [cited 2006 May 1]; Available from: http://www.informatics.jax.org.
3. R. Bunescu, et al., *Learning to extract proteins and their interactions from medline abstracts*, in *Proceedings of the ICML-2003 Workshop on Machine Learning in Bioinformatics*. 2003: Washington DC. p. 46-53.
4. R. Bunescu and R.J. Mooney, *Relational markov networks for collective information extraction*, in *ICML-2004 Workshop on Statistical Relational Learning*. 2004.
5. A.M. Cohen, *Unsupervised gene/protein named entity normalization using automatically extracted dictionaries*, in *Proceedings of the ACL-ISMB Workhop on Linking Biological Literature, Ontologies and Databses: Mining Biological Semantics*. 2005: Detroit. p. 14-24.
6. W.W. Cohen. *Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data*. 2004 [cited 2006 May 1]; Available from: http://minorthird.sourceforge.net.
7. N. Collier, et al., *The GENIA project: Corpus-based knowledge acquisition and information extraction from genome research papers*, in *Proceedings of EACL-99*. 1999. p. 271-272.
8. M. Collins, *Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms*, in *Empirical Methods in Natural Language Processing (EMNLP)*. 2002.
9. M. Craven and J. Kumlien, *Constructing biological knowledge bases by extracting information from text sources*, in *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*. 1999, AAAI Press. p. 77-86.
10. K. Franzén, et al., *Protein names and how to find them.* International Journal of Medical Informatics, 2002. **67**(1-3): p. 49-61.
11. K. Fukuda, et al., *Toward information extraction: Identifying protein names from biological papers*, in *Proceedings of the Pacific Symposium on Biocomputing '98 (PSB-98)*. 1998. p. 707-718.
12. B. Hachey, et al., *Grounding Gene Mentions with Respect to Gene Database Identifiers*, in *BioCreAtIvE Workshop Handouts*. 2004: Granada, Spain.
13. D. Hanisch, et al., *ProMiner: rule-based protein and gene entity recognition.* BMC Bioinformatics, 2005. **6**(Suppl 1).
14. L. Hirschman, et al., *Overview of BioCreAtIvE task 1B: normalized gene lists.* BMC Bioinformatics, 2005. **6**(Suppl 1).
15. L. Hirschman, et al., *Overview of BioCreAtIvE: critical assessment of information extraction for biology.* BMC Bioinformatics, 2005. **6**(Suppl 1).

16. K. Humphreys, G. Demetriou, and R. Gaizauskas, *Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures*, in *Proceedings of the Pacific Symposium on Biocomputing (PSB-2000)*. 2000. p. 502-513.
17. K. Humphreys, et al., *Univ. of Sheffield: Description of the LASIE-II system as used for MUC-7*, in *Message Understanding Conference Proceedings (MUC-7)*. 1998: Fairfax, Virginia.
18. Z. Kou, W.W. Cohen, and R.F. Murphy, *High-recall protein entity recognition using a dictionary.* Bioinformatics, 2005. **21**(Suppl 1): p. i266-i273.
19. V.I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals.* Soviet Physics Doklady, 1966. **10**(8): p. 707-710.
20. E. Minkov, R.C. Wang, and W.W. Cohen, *Extracting personal names from emails: Applying named entity recognition to informal text*, in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. 2005: Vancouver, B.C., Canada. p. 443-450.
21. E. Minkov, et al., *NER Systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction*, in *Human Language Technology Conference - North American Chapter of the ACL (HLT-NAACL)*. 2006: New York City.
22. A.A. Morgan, et al., *Gene name identification and normalization using a model organism database.* Journal of Biomedical Informatics, 2004. **37**(6): p. 396-410.
23. T.C. Rindflesch, et al., *Edgar: Extraction of drugs, genes and relations from the biomedical literature*, in *Proceedings of the Pacific Symposium on biocomputing (PSB-2000)*. 2000. p. 514-525.
24. S. Sarawagi and W.W. Cohen, *Semi-markov conditional random fields for information extraction*, in *NIPS*. 2004.
25. C. Sutton and A. McCallum, *Collective segmentation and labeling of distant entities in information extraction*, in *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*. 2004: Banff, Canada.
26. A. Vlachos and C. Gasperin, *Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain*, in *Proceedings of the HLT-NAACL Workshop on Linking Natural Language Processing and Biology*. 2006: Brooklyn, New York.
27. B. Wellner, *Weakly Supervised Learning Methods for Improving the Quality of Gene Name Normalization Data*, in *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*. 2005: Detroit. p. 1-8.