Pacific Symposium on Biocomputing 13:628-639(2008)

COMPARING USABILITY OF MATCHING TECHNIQUES FOR NORMALISING BIOMEDICAL NAMED ENTITIES

XINGLONG WANG AND MICHAEL MATTHEWS

School of Informatics, University of Edinburgh
Edinburgh, EH8 9LW, UK
{xwang,mmatsews}@inf.ed.ac.uk

String matching plays an important role in biomedical Term Normalisation, the task of linking mentions of biomedical entities to identifiers in reference databases. This paper evaluates exact, rule-based and various string-similarity-based matching techniques. The matchers are compared in two ways: first, we measure precision and recall against a gold-standard dataset and second, we integrate the matchers into a curation tool and measure gains in curation speed when they were used to assist a curator in normalising protein and tissue entities. The evaluation shows that a rule-based matcher works better on the gold-standard data, while a string-similarity based system and exact string matcher win out on improving curation efficiency.

1. Introduction

Term Normalisation (TN) [1] is the task of grounding a biological term in text to a specific identifier in a reference database. TN is crucial for automated processing of biomedical literature, due to ambiguity in biological nomenclature [2, 3, 4, 5]. For example, a system that extracts protein-protein interactions (PPIs) would ideally collapse interactions involving the same proteins, even though these are named by different word forms in the text. This is particularly important if the PPIs are to be entered into a curated database, which refers to each protein by a canonical unique identifier.

A typical TN system consists of three components: an ontology processor, which expands or prunes the reference ontology; a string matcher, which compares entity mentions in articles against entries in the processed ontology; and finally a filter (or a disambiguator) that removes false positive identifiers using rules or statistical models [6, 7]. The string matcher is arguably the core component: a matcher that searches a database and retrieves entries that exactly match an entity mention can form a simple TN system. The other two components are important but they can be viewed as extras that may help further improve the performance of the matcher. A reasonable assumption is that if a matching system

Pacific Symposium on Biocomputing 13:628-639(2008)

can help improve curation speed, then more complex TN systems should be even more helpful. Indeed, the matching systems described in this paper can be used as stand-alone TN modules, and can also work in conjunction with external ontology processors and filters.

Much work has been carried out on evaluating performance of TN systems on Gold Standard datasets [6, 8]. However, whether such systems are really helpful in speeding up curation has not yet been adequately addressed. This paper focuses on investigating matching techniques and attempts to answer which ones are most helpful in assisting biologists to perform TN curation. We emphasise *assisted*, rather than *automated* curation because, at least in the short term, replacing human curators is not practical [9, 10], particularly on TN tasks that involve multiple types of biological entities across numerous organisms. We believe that designing tools that help improve curation efficiency is more realistic. This paper compares different techniques for implementing matching: exact, rule-based, and string similarity methods. These are tested by measuring recall and precision over a Gold Standard dataset, as well as by measuring the time taken to carry out TN curation when using each of the matching systems. In order to examine whether the matching techniques are portable to new domains, we tested them on two types of entities in the curation experiment — proteins and tissues (of human species).

This paper is organised as follows: Section 2 gives a brief overview of related work. Section 3 summarises the matching algorithms that we studied and compared. Section 4 presents experiments that evaluated the matching techniques on Gold Standard datasets, while Section 5 describes an assisted curation task and discusses how the fuzzy matching systems helped. Section 6 draws conclusions and discusses directions of future work.

2. Related Work

TN is a difficult task because of the pervasive variability of entity mentions in the biomedical literature. Thus, a protein will typically be named by many orthographic variants (e.g., *IL-5* and *IL5*) and by abbreviations (e.g., *IL5* for *Interleukin-5*), etc. The focus of this paper is how fuzzy matching techniques [11] can handle such variability. Two main factors affect performance of fuzzy matching: first, the quality of the lexicon, and second, the matching technique adopted. Assuming the same lexicon is used, there are three classes of matching techniques: those that rely on exact searches, those that search using hand-written rules, and those that compute string-similarity scores.

First, with a well constructed lexicon, **exact matching** can yield good results [12, 13]. Second, **rule-based methods**, which are probably the most widely

Pacific Symposium on Biocomputing 13:628-639(2008)

used matching mechanism for TN, have been reported as performing well. Their underlying rationale is to alter the lexical forms of entity mentions in text with a sequence of rules, and then to return the first matching entry in the lexicon. For example, one of the best TN systems submitted to the recent BioCreAtIvE 2 Gene Normalisation (GN) task [14] exploited rules and background knowledge extensively.[a] The third category is **string-similarity matching** approaches. A large amount of work has been carried out on matching by string similarity in fields such as database record linkage. Cohen et al. [15] provided a good overview on a number of metrics, including edit-distance metrics, fast heuristic string comparators, token-based distance metrics, and hybrid methods. In the BioCreAtIvE 2 GN task, several teams used such techniques, including Edit Distance [16], Soft-TFIDF [17] and JaroWinkler [18].

Researchers have compared the matching techniques with respect to performance on Gold Standard datasets. For example, Fundel et al. [12] compared their exact matching approach to a rule-based approximate matching procedure implemented in ProMiner [19] in terms of recall and precision. They concluded that approximate search did not improve the results significantly. Fang et al. [20] compared their rule-based system against six string-distance based matching algorithms. They found that by incorporating approximate string matching, overall performance was slightly improved. However, in most scenarios, approximate matching only improved recall slightly and had a non-trivial detrimental effect upon precision. Results reported by Fang et al. [20] and Fundel et al. [12] were based on measuring precision and recall on Gold Standard datasets which contained species-specific gene entities. However, in practice, curators might need to curate not only genes, but many other types of entities.Section 5 presents our investigation on whether matching techniques can assist curation in a setup more analogous to these real-world situations.

3. Matching Techniques

This section outlines the rule-based and the string similarity-based algorithms that were used in our experiments. Evaluation results from the BioCreAtIvE 2 GN task on human genes seem to indicate that rule-based systems perform better. The weakness of rule-based systems, however, is that they may be less portable to new domains. By contrast, string similarity-based matching is more generic and can be easily deployed to deal with new types of entities in new domains.

[a]See Hirschman et al. [6] for an overview of the BioCreAtIvE 1 GN task and Morgan and Hirschman [8] for the BioCreAtIvE II GN task.

Pacific Symposium on Biocomputing 13:628-639(2008)

3.1. *Rule-based Matching*

For each protein mention, we used the following rules[b] to create an ordered list of possible RefSeq[c] identifiers.

(1) Convert the entity mention to lowercase and look up the synonym in a lowercase version of the RefSeq database.

(2) Normalise the mention[d] (NORM MENTION), and look up the synonym in a normalised version of the RefSeq database (NORM lexicon).

(3) Remove prefixes (*p*, *hs*, *mm*, *m*, *p* and *h*), add and remove suffixes (*p*, 1, 2) from the NORM MENTION and look up result in the NORM lexicon.

(4) Look up the NORM MENTION in a lexicon derived from RefSeq (DERIVED lexicon).[e]

(5) Remove prefixes (*p*, *hs*, *mm*, *m*, *p* and *h*), add and remove suffixes (*p*, 1, 2) from the NORM MENTION, and look up result in the DERIVED lexicon.

(6) Look up the mention in the abbreviation map created using the Schwartz and Hearst [21] abbreviation tagger. If this mention has a corresponding long form or corresponding short form, repeat steps 1 through 5 for the corresponding form.

3.2. *String Similarity Measures*

We considered six string-similarity metrics: *Monge-Elkan*, *Jaro*, *JaroWinkler*, *mJaroWinkler*, *SoftTFIDF* and *mSoftTFIDF*. Monge-Elkan is an affine variant of the Smith-Waterman distance function with particular cost parameters, and scaled to the interval $[0, 1]$. The Jaro metric is based on the number and order of the common characters between two strings. A variant of the Jaro measure due to Winkler uses the length of the longest common prefix of the two strings and rewards strings which have a common prefix. A recent addition to this family is a modified JaroWinkler [18] (mJaroWinkler), which adapts the weighting parameters and takes into account factors such as whether the lengths of the two strings are comparable and whether they end with common suffixes.

We also tested a 'soft' version of the TF-IDF measure [22], in which similar tokens are considered as well as identical ones that appear in both strings. The similarity between tokens are determined by a similarity function, where we used

[b] Some of the rules were developed with reference to previous work [13, 20].

[c] See `http://www.ncbi.nlm.nih.gov/RefSeq/`.

[d] Normalising a string involves converting Greek characters to English (e.g., α →alpha), converting to lowercase, changing sequential indicators to integer numerals (e.g., *i*, *a*, *alpha*→1, etc.) and removing all spaces and punctuation. For example, *rab1*, *rab-1*, *rab*α, *rab I* are all normalised to *rab1*.

[e] The lexicon is derived by adding the first and last word of each synonym entry in the RefSeq database to the lexicon and also by adding acronyms for each synonym created by intelligently combining the initial characters of each word in the synonym. The resulting list is pruned to remove common entries.

Pacific Symposium on Biocomputing 13:628-639(2008)

JaroWinkler for SoftTFIDF and mJaroWinkler for mSoftTFIDF. We deem two tokens similar if they have a similarity score that is greater than or equal to 0.95 [17], according to the corresponding similarity function.

4. Experiments on Gold Standard Datasets

We evaluated the competing matching techniques on a Gold Standard dataset over a TN task defined as follows: given a mention of a protein entity in a biomedical article, search the ontology and assign one or more IDs to this protein mention.

4.1. *Datasets and Ontologies*

We conducted the experiments on a protein-protein interaction (PPI) corpus annotated for the TXM [18, 23] project, which aims at producing NLP-based tools to aid curation of biomedical papers. Various types of entities and PPIs were annotated by domain experts, whereas only the TN annotation on proteins was of interest in the experiments presented in this section.[f] 40% of the papers were doubly annotated and we calculated inter-annotator agreement (IAA) for TN on proteins, which is high at 88.40%.

We constructed the test dataset by extracting all 1,366 unique protein mentions, along with their manually normalised IDs, from the PPI corpus. A lexicon customised for this task was built by extracting all synonyms that are associated with RefSeq IDs that were assigned to the protein mentions in the test dataset. In this way, the lexicon was guaranteed to have an entry for every protein mention and the normalisation problem can be simplified as a string matching task.[g] Note that as our data contains proteins from various model organisms, and thus this TN task is more difficult than the corresponding BioCreAtIvE 1 & 2 GN tasks, which dealt with species-specific genes.

4.2. *Experimental Setup*

We applied the rule-based matching system and six similarity-based algorithms to the protein mentions in the test dataset.[h] A case-insensitive (CI) exact match baseline system was also implemented for comparison purpose.

[f] We have an extended version of this dataset in which more entity types are annotated. The curation experiment described in Section 5 used protein and tissue entities in that new dataset.

[g] Although we simplified the setup for efficiency, the comparison was fair because all matching techniques used the same lexicon.

[h] We implemented the string-similarity methods based on the *SecondString* package. See `http://secondstring.sourceforge.net/`

Pacific Symposium on Biocomputing 13:628-639(2008)

Given a protein mention, a matcher searches the protein lexicon, and returns one match. The exact and rule-based matchers return the first match according to the rules and the similarity-based matchers return the match with the highest confidence score. It is possible that a match maps to multiple identifiers, in which case all identifiers were considered as answers.

In evaluation, for a given protein mention, the ID(s) associated with a match retrieved by a matcher are compared to the manually annotated ID. When a match has multiple IDs, we count it as a hit if one of the IDs is correct. Although this setup simplifies the TN problem and assumes a perfect filter that always successfully removes false positives, it allows us to focus on investigating the matching performance without interference from NER errors or errors caused by ambiguity.

4.3. *Results and Discussion*

We used metrics precision (P), recall (R) and $F1$, for evaluation. Table 1 shows performance of the matchers.

Table 1. Precision (P), recall (R) and $F1$ of fuzzy matching techniques as tested on the PPI corpus. Figures are in percentage.

Matcher	*P*	*R*	$F1$	*Matcher*	*P*	*R*	$F1$
MongeElkan	51.3	51.3	51.3	mSoftTFIDF	66.0	61.5	63.7
Jaro	59.4	59.3	59.3	**Rule-based**	81.4	57.5	**67.4**
JaroWinkler	61.7	61.6	61.7	mJaroWinkler	61.2	61.1	61.1
SoftTFIDF	66.5	62.2	64.3	Exact Match (CI)	77.2	33.8	47.0

Both the rule-based and the string-similarity based approaches outperformed the exact match baseline, and rule-based system outperformed the string-similarity-based ones. Nevertheless, the SoftTFIDF matcher performed only slightly worse than the winner,[i] and we should note that string-similarity based matchers have the advantage of portability, so that they can be easily adopted to other types of biomedical entities, such as tissues and experimental methods, as long as the appropriate lexicons are available.

Among the similarity-based measures, the two SoftTFIDF-based methods outperformed others. As discussed in [22], two advantages of the SoftTFIDF over other similarity-based approaches are: first, token order is not important so permu-

[i]The rule-based system yields higher recall but lower precision than the similarity-based systems. Tuning the balance between recall and precision may be necessary for different curation tasks. See [23] for more discussion on this issue.

tation of tokens are considered the same, and second, common but uninformative words do not greatly affect similarity.

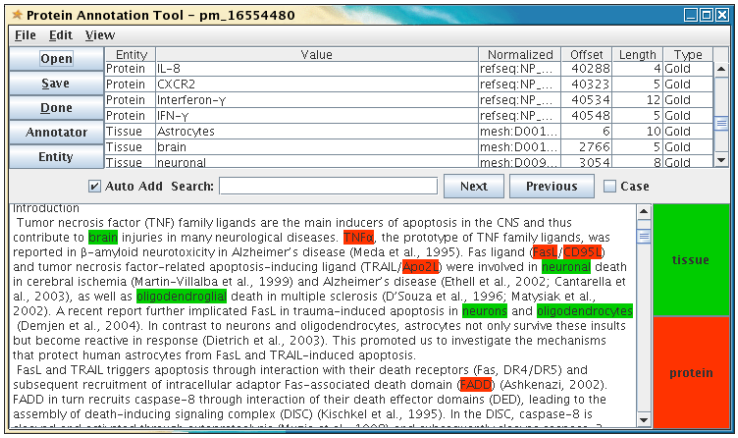
5. Curation Experiment

We carried out a TN curation experiment where three matching systems were supplied to a curator to assist in normalising a number of tissue and protein entities. A matcher resulting in faster curation is considered to be more helpful.

5.1. *Experimental Setup*

We designed a realistic curation task on TN as follows: a curator was asked to normalise a number of tissue and protein entities that occurred in a set of 78 PubMed articles.[j] Tissues were to be assigned to MeSH[k] IDs and proteins to RefSeq IDs. We selected only human proteins for this experiment, because although species is a major source of ambiguity in biological entities [7], we wanted to focus on investigating how matching techniques affect curation speed in this work.

Figure 1. A screenshot of the curation tool.



Curation was carried out using an in-house curation tool (as shown in Figure 1). When loaded, the tool displays a full-length article and highlights a number

[j]The articles were taken from an extended version of the dataset described in Section 4.1, in which tissues and proteins were already manually marked up and normalised. The normalisations were concealed from the curator and only used after the experiment to assess the quality of the curation.
[k]See `http://www.nlm.nih.gov/mesh/MBrowser.html`.

Pacific Symposium on Biocomputing 13:628-639(2008)

of randomly selected protein and tissue entities Only unique entity mentions in each article were highlighted. To make sure that the numbers of entities were distributed evenly in the articles, a maximum of 20 tissues and 20 proteins were highlighted in each article.[l]

We integrated three matching techniques into the curation tool to assist curation: (1) SoftTFIDF, the best performing string-similarity-based matching method in our previous experiment; (2) rule-based matching;[m] and (3) exact matching. The 78 articles were randomly divided into three sets, each of which used a different matching technique, and then the articles were randomly presented to the curator. When an article was loaded into the tool, the normalisations guessed by one of the matchers were also added. When the curator clicked on a highlighted entity mention, a dialogue window would pop up, showing its pre-loaded normalisations, along with a brief description of each ID in order to help the curator select the right ID. The descriptions were extracted from RefSeq and MeSH, consisting of synonyms corresponding to the ID.

The curation tool also provided a search facility. When a matcher misses the correct IDs, the curator can manually query RefSeq and MeSH lexicons. The search facility was rather basic and carried out 'exact' and 'starts with' searches. For example, if a matcher failed to suggest a correct normalisation for protein mention "α-*DG*" and if the curator happened to know that "*DG*" was an acronym for "*dystroglycan*", then she could query the RefSeq lexicon using the term "*alpha-dystroglycan*". We logged the time spent on manual searches, in order to analyse the usefulness of the matching techniques and how they can be further improved. As with the experiments carried out on the Gold Standard dataset, we followed a 'bag' approach, which means that, for each mention, a list of identifiers, instead of a single one, was shown to the curator.[n]

5.2. *Results and Discussion*

Tables 2 and 3[o] show the average curation time that the curator spent on normalising a tissue or a protein with respect to the matching techniques. There are two

[l]Because articles contain different numbers of entities, the total numbers of protein and tissue entities in this experiment are different. See Table 2 and 3 for exact figures.

[m]We used the same system as described in Section 3 for protein normalisation. For tissue normalisation, a rudimentary system was used that first carries out a case-insensitive (CI) match, followed by a CI match after adding and removing an *s* from the tissue mention, and finally adding the MeSH ID for the Cell Line if the mention ends in *cells*.

[n]This is in-line with our evaluation on the gold-standard dataset where a metric of *top* n accuracy was used.

[o]The standard deviations were high due to the fact that some entities are more difficult to normalise than others.

Pacific Symposium on Biocomputing 13:628-639(2008)

types of normalisation events: 1) a matcher successfully suggested a normalisation and the curator accepted it; and 2) a matcher failed to return a hit, and the curator had to perform manual searches to normalise the entity in question.

Table 2. Time spent on normalising tissues with three matching techniques.

	including manual searches			*excluding manual searches*		
Matcher	*# of entities*	*time(ms)*	*StdDev*	*# of entities*	*time(ms)*	*StdDev*
Exact	283	7,078	8,757	127	2,198	2,268
Rule-based	326	6,639	8,607	172	2,158	1,133
SoftTFIDF	292	**6,044**	7,596	208	2,869	2,463

Table 3. Time spent on normalising proteins with three matching techniques.

	including manual searches			*excluding manual searches*		
Matcher	*# of entities*	*time(ms)*	*StdDev*	*# of entities*	*time(ms)*	*StdDev*
Exact	196	**6,972**	8,859	147	3,714	4,479
Rule-based	129	8,615	12,809	110	6,744	11,030
SoftTFIDF	108	11,218	17,334	88	7,381	9,071

The columns titled "excluding manual searches" and "including manual searches" reflect the two types of events. By examining averaged curation time cost on each, we can see how the matchers helped. For example, from the "excluding manual searches" column in Table 2, we observe that the curator required more time (i.e., 2,869 *ms*.) to find and accept the correct ID from the candidates suggested by SoftTFIDF, whereas the time in the "including manual searches" column shows that overall using SoftTFIDF was faster than the other two matchers. This is because in the majority of cases (208 out of 292), the correct ID was in the list returned by SoftTFIDF, which allowed the curator to avoid performing manual searches and thus saved time. In other words, the curator had to perform time-consuming manual searches more often when assisted by the exact and the rule-based matchers.

Overall, on tissue entities, the curator was faster with help from the SoftTFIDF matcher, whereas on proteins the exact matcher worked better.[p] To explain this, we should clarify that the major elements that can affect curation speed are: 1) the

[p]We performed significance tests on both the protein and tissue data using R. Given that the data is not normally distributed as indicated by the Kolomorov-Smirnov normality test, we used the non-parametric Kruskal-Wallis test which indicates that the differences are significant with $p = .02$ for both data sets.

Table 4. Average bagsizes of tissue and protein entities.

Type	*Matcher*	*Cnt (bagsize>=0)*	*Avg. bagsize*	*Cnt (bagsize=0)*	*Percentage*
Tissue	Exact	283	0.43	160	56.5%
	Rule-based	326	0.66	111	34.0%
	SoftTFIDF	292	5.38	7	2.4%
Protein	Exact	196	0.90	51	26.02%
	Rule-based	129	5.12	14	10.85%
	SoftTFIDF	108	13.97	9	8.50%

performance of the matcher, 2) time cost in eyeballing the IDs suggested, and 3) the time spent on manual searches when the matcher failed.

Therefore, although we evaluated the matchers on a Gold Standard dataset and concluded that the rule-based matcher should work best on normalising protein entities (see Section 4), this does not guarantee that the rule-based matcher will lead to an improvement in curation speed.

The second factor is due to the sizes of the bags. The SoftTFIDF matcher returns smaller sets of IDs for tissues but bigger ones for proteins. Table 4 shows the average bagsizes and the percentage when bagsize is zero, in which case the matcher failed to find any ID. One reason that SoftTFIDF did not help on proteins might be the average bagsize is too big at 13.97, and the curator had to spend time reading the descriptions of all IDs.

As for the third factor, on tissues, 56.5% of the time the exact matcher failed to find any ID and the curator had to perform a manual search; by contrast, the SoftTFIDF matcher almost always returned a list of IDs (97.6%), so very few manual searches were needed.

As mentioned, the articles to curate were presented to the curator in random order, so that the potential influence to performance of normalisation resulting from training curve and fatigue should distribute evenly among the matching techniques and therefore not bias the results. On the other hand, due to limitation in time and resources, we only had one curator to carry out the curation experiment, which may cause the results to be subjective. In the near future, we plan to carry out larger scale curation experiments.

6. Conclusions and Future Work

This paper reports an investigation into the matching algorithms that are key components in TN systems. We found that a rule-based system that performed better in terms of precision and recall, as measured on a Gold Standard dataset, was not the most useful system in improving curation speed, when normalising protein and tissue entities in a setup analogous to a real-world curation scenario. This re-

Pacific Symposium on Biocomputing 13:628-639(2008)

sult highlights concerns that text mining tools achieving better results as measured by traditional metrics might not necessarily be more successful in enhancing curators' efficiency. Therefore, at least for the task of TN, it is critical to measure the usability of text mining tools extrinsically in actual curation exercises. We have learnt that, besides the performance of the matching systems, many other factors are also important. For example, the balance between precision and recall (i.e., presenting more IDs with higher chances to include the correct one, or less IDs where the answer is more likely to be missed), and the backup tool (e.g., the manual search facility in the curation tool) used when the assisting system fails, can both have significant effects on usability. Furthermore, in real-world curation tasks that often involve more than one entity type, approaches with better portability (e.g., string-similarity-based ones) may be preferred. Our results also indicated that it might be a good idea to address different types of entities with different matching techniques.

One direction for future work is to conduct more curation experiments so that the variability between curators can be smoothed (e.g., some curators may prefer seeing more accurate NLP output whereas others may prefer higher recall). Meanwhile, we plan to improve the matching systems by integrating ontology processors and species disambiguators [7].

Acknowledgements

The work reported in this paper was done as part of a joint project with Cognia (`http://www.cognia.com`), supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`). We also thank Kirsten Lillie, who carried out curation for our experiment, and Ewan Klein, Barry Haddow, Beatrice Alex and Claire Grover who gave us valuable feedback on this paper.

References

1. M. Krauthammer and G. Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics (Special Issue on Named Entity Recogntion in Biomedicine)*, 37(6):512–526, 2004.
2. L. Hirschman, A. A. Morgan, and A. S. Yeh. Rutabaga by any other name: extracting biological names. *J Biomed Inform*, 35(4):247–259, 2002.
3. O. Tuason, L. Chen, H. Liu, J. A. Blake, and C. Friedman. Biological nomenclature: A source of lexical knowledge and ambiguity. In *Proceedings of PSB*, 2004.
4. L. Chen, H. Liu, and C. Friedman. Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2):248–256, 2005.
5. K. Fundel and R. Zimmer. Gene and protein nomenclature in public databases. *BMC Bioinformatics*, 7:372, 2006.

Pacific Symposium on Biocomputing 13:628-639(2008)

6. L. Hirschman, M. Colosimo, A. Morgan, and A. Yeh. Overview of BioCreAtlvE task 1B: normalised gene lists. *BMC Bioinformatics*, 6, 2005.
7. X. Wang. Rule-based protein term identification with help from automatic species tagging. In *Proceedings of CICLING 2007*, pages 288–298, Mexico City, 2007.
8. A. A. Morgan and L. Hirschman. Overview of BioCreative II gene normalisation. In *Proceedings of the BioCreAtIvE II Workshop*, Madrid, 2007.
9. I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. Bader, K. Michalickova, T. Pawson, and C. Hogue. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4:11, 2003.
10. Nikiforos Karamanis, Ian Lewin, Ruth Seal, Rachel Drysdale, and Edward Briscoe. Integrating natural language processing with FlyBase curation. In *Proceedings of PSB*, pages 245–256, Maui, Hawaii, 2007.
11. G. Nenadic, S. Ananiadou, and J. McNaught. Enhancing automatic term recognition through term variation. In *Proceedings of Coling*, Geneva, Switzerland, 2004.
12. K. Fundel, D. Güttler, R. Zimmer, and J. Apostolakis. A simple approach for protein name identification: prospects and limits. *BMC Bioinformatics*, 6(Suppl 1):S15, 2005.
13. A. Cohen. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, 2005.
14. J. Hakenberg, L. Royer, C. Plake, H. Strobelt, and M. Schroeder. Me and my friends: Gene mention normalization with background knowledge. In *Proceedings of the BioCreAtIvE II Workshop 2007*, Madrid, 2007.
15. W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IIWeb-03 Workshop*, 2003.
16. W. Lau and C. Johnson. Rule-based gene normalisation with a statistical and heuristic confidence measure. In *Proceedings of the BioCreAtIvE II Workshop 2007*, 2007.
17. C. Kuo, Y. Chang, H. Huang, K. Lin, B. Yang, Y. Lin, C. Hsu, and I. Chung. Exploring match scores to boost precision of gene normalisation. In *Proceedings of the BioCreAtIvE II Workshop 2007*, Madrid, 2007.
18. C. Grover, B. Haddow, E. Klein, M. Matthews, L. A. Nielsen, R. Tobin, and X. Wang. Adapting a relation extraction pipeline for the BioCreAtIvE II task. In *Proceedings of the BioCreAtIvE II Workshop 2007*, Madrid, 2007.
19. D. Hanisch, K. Fundel, H-T Mevissen, R Zimmer, and J Fluck. ProMiner: Organism-specific protein name detection using approximate string matching. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.
20. H. Fang, K. Murphy, Y. Jin, J. Kim, and P. White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of the HLT-NAACL BioNLP Workshop*, New York, 2006.
21. A.S. Schwartz and M.A. Hearst. Identifying abbreviation definitions in biomedical text. In *Proceedings of PSB*, 2003.
22. W. W. Cohen and E. Minkov. A graph-search framework for associating gene identifiers with documents. *BMC Bioinformatics*, 7:440, 2006.
23. B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, S. Roebuck, R. Tobin, and X. Wang. Assisted curation: does text mining really help? In *The Pacific Symposium on Biocomputing (PSB)*, 2008.