Combining terminology resources and statistical methods for entity recognition: an evaluation

Angus Roberts, Robert Gaizauskas, Mark Hepple, Yikun Guo

Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello, Sheffield S1 4DP
{initial.surname}@dcs.shef.ac.uk

Abstract

Terminologies and other knowledge resources are widely used to aid entity recognition in specialist domain texts. As well as providing lexicons of specialist terms, linkage from the text back to a resource can make additional knowledge available to applications. Use of such resources is especially pertinent in the biomedical domain, where large numbers of these resources are available, and where they are widely used in informatics applications. Terminology resources can be most readily used by simple lexical lookup of terms in the text. A major drawback with such lexical lookup, however, is poor precision caused by ambiguity between domain terms and general language words. We combine lexical lookup with simple filtering of ambiguous terms, to improve precision. We compare this lexical lookup with a statistical method of entity recognition, and to a method which combines the two approaches. We show that the combined method boosts precision with little loss of recall, and that linkage from recognised entities back to the domain knowledge resources can be maintained.

1. Introduction

Specialist domains are characterised by extensive use of technical and domain specific terminology. Term recognition is an important step towards Named Entity Recognition (NER) in these domains: entities, or things in the real world, are often referred to by terms in the text. Large scale knowledge resources such as terminologies and ontologies are typically available in these same domains. We might expect such resources to have some use in term and entity recognition. We might also expect entity recognition to add value by linking entities back to these knowledge resources, making additional information available to applications and their users.

Although large scale resources offer big advantages, they also have a major disadvantage: most have not been designed with natural language processing in mind. They may suffer from low coverage in some area of importance to an application, and from problems of ambiguity in other areas. Through combining dictionary lookup with statistical models, we hope to overcome these disadvantages, while retaining the advantages of linking to the underlying resources. Can, in practice, use of these large scale resources be shown to benefit entity recognition? This is our research question. Our question parallels a long-standing question of gazetteer use for NER in Information Extraction: are large gazetteers useful for NER (Stevenson and Gaizauskas, 2000), or can statistical models of context alone provide sufficient performance (Mikheev et al., 1999)? We examine this question with respect to biomedicine. Specifically, we look at clinical documents. This domain is characterised by complex terminologies, and by a wealth of large terminology resources. Our question is, however, pertinent to any technical domain.

Dictionary lookup in the biomedical domain is especially prone to problems of ambiguity. This has been noted for gene names (Proux et al., 1998; Hirschman et al., 2002), but is also true for clinical text. Large numbers of abbreviations are used, and these are often ambiguous with short words in general language. For example, many one and two character words are abbreviations for chemical elements, after which medical investigations are named. “I” is an abbreviation for Iodine, and used to mean an Iodine test, but of course most commonly appears as the personal pronoun. Some dictionary lookup methods, our own included, match morphological roots of tokens, rather than token strings. For example, the verb “be” (and therefore its derivatives if matching morphological roots) is ambiguous with “BE”, an abbreviation for Bacterial Endocarditis.

Hirschman et al. (2002), looking at gene names, demonstrated the scale of this problem with a simple baseline experiment. Using a standard resource, they extracted gene names from research paper abstracts, with a precision of 7% and a recall of 31%. By eliminating potential names of three or less characters, precision rose to 29%, while recall only dropped to 26%. Several solutions to this problem have been investigated. From the examples above, it would seem sensible to use additional information, such as part of speech, to disambiguate dictionary matches. Proux et al. (1998) used such an approach to recognise gene names ambiguous with general language words: a potential gene name was eliminated from consideration if it had a non-noun part-of-speech. Other solutions have shown that syntactic information is not always necessary, instead using the domain specificity of potential terms. For example, Stevenson and Gaizauskas (2000) looked at entity recognition in newswire, showing that large gazetteers can improve recall, but that they may also introduce ambiguity. They used two methods to overcome this. First, they removed those words from the gazetteer that also occur in a standard dictionary. Second, they removed those words that occurred more frequently in their training corpus as non-terms than terms. Both of these methods showed improved results.

Dictionary lookup can be contrasted to machine learning approaches. Such techniques are widespread in the biomedical domain, especially for term and entity recognition of proteins and genes (see Ananiadou and Nenadic (2006) and Park and Kim (2006) for reviews). Several applica-

Entity type	Brief description	Number of instances
Condition	Symptom, diagnosis, complication, conditions, problems, injuries etc.	739
Drug or device	Usually a drug, but can be other prescribed items such as medical devices	272
Intervention	Action performed by a clinician, targeted at a patient, locus, or condition	298
Investigation	Tests, measurements, and studies	325
Locus	Anatomical structure or location, body substance etc.	490
Total		**2124**

Table 1: Entity types and numbers of instances in a gold standard corpus of 77 narratives.

tions have used a "pure" machine learning approach, in which no external dictionaries are used. Tanabe and Wilbur (2002), for example, used transformation based learning to build ABGENE, a gene and protein name recogniser. ABGENE includes a Brill POS tagger trained on a corpus that has been hand-labelled with gene and protein names. Others have combined dictionary lookup and machine learning of statistical models. Mika and Rost (2004) trained several Support Vector Machines (SVMs) on lexical features. A further SVM was trained on the outputs of these, combined with a dictionary lookup. Use of the dictionary increased performance significantly. Yamamoto et al. (2003) used an SVM to find protein names in text. Features included several that encoded whether a term appeared in a dictionary, which was built from a biomedical corpus and protein knowledge bases. These lookup features proved crucial.

We examine entity recognition of medically important entities in texts from patient records. Although statistical and machine learning techniques have been used in this domain (see Pakhomov et al. (2005) for example), they are not as widely used as for protein and gene recognition. In clinical text, dictionary lookup combined with syntactic parsing is much more common. Our experiments use a system which contains a dictionary based lookup of terms from large scale terminologies, filtering of ambiguity from this dictionary lookup, and supervised learning of statistical entity recognition models. As with protein and gene recognition, these approaches are not mutually exclusive: a dictionary based term lookup can be used to provide features for statistical models. We therefore examine these components independently, and in combination. We also look at whether a combined method can retain a major advantage of dictionary lookup, linkage from recognised entities back to domain resources.

2. Corpus

A major difficulty when evaluating natural language processing (NLP) over clinical texts, is the almost complete absence of gold standards for the domain. This is largely due to issues of data confidentiality. The CLEF project (Rector et al., 2003) has been fortunate in obtaining a large corpus of over 500K documents from over 20K patients. We have used a small subset of these documents to build a gold standard of manually annotated entities and relations. The gold standard has been carefully constructed using best practice methods, as described fully in (Roberts et al., 2007). Documents were annotated by two independent, clinically trained, annotators, and a consensus annotation created by a third.

For the experiments reported in this paper, we used 77 gold standard documents of a single type, clinical narratives (generally letters from one clinician to another that describe a patient's progress). We used consensus annotations of five entity types on these narratives. By *entity*, we mean some real-world thing, event or state referred to in the text. The entity types are shown in Table 1, together with the total number of instances of each type in all 77 documents. In addition to the annotated gold standard, we have also built an unannotated development corpus of similar documents. This was used whenever inspection of documents was required as part of system development. The annotated corpus was never inspected in development.

3. Algorithms and resources

The corpus is pre-processed by tokenisation, sentence splitting, and part of speech tagging, using the GATE text mining toolkit (Cunningham et al., 2002). Our entity recognition components are also implemented in GATE.

3.1. Dictionary based term recognition

For dictionary based lookup, we use Termino: a large-scale terminological resource designed specifically for text processing (Harkema et al., 2004). Termino consists of two parts. The first is a database constructed from existing terminology resources. Termino provides uniform access to these resources, and links from recognised terms to resource entries. The second part consists of finite state recognisers (FSRs) compiled from the database. Terms found by a FSR are associated with a unique ID linking back to the external resource, and with a semantic type derived from the external resource.

Our principle terminology resource in CLEF is the Unified Medical Language System (UMLS) (Lindberg et al., 1993) [1]. UMLS is the largest source of medical vocabulary, being a superset of other resources, and provides links from terms to other information, such as semantic types.

We import UMLS terms into Termino. A significant number of the terms in UMLS are of little value for medical NLP tasks. For example, they represent non-medical concepts, are case variants of other terms, or are complex knowledge engineering class names that are unlikely to be found in text. These terms degrade the performance of NLP applications based on UMLS (Aronson, 2005). We filter out such terms, prior to importing into Termino. For example, we reject long terms (> 5 words) and terms containing certain constructs that mark them as class names. The full set of rejection criteria is derived from McCray et al. (2001), McCray et al. (2002), and Aronson (2005).

3.1.1. Filter and Supplementary Term Lists

Despite this rejection of many UMLS terms that are not suitable for NLP, described above, we still found that Termino falsely matched common general language words. To

[1] We use the UMLS Metathesaurus 2007AB release, taking terms from license category 3 source vocabularies.

identify these, we ran Termino over our development corpus, and manually inspected the results. From all matches, we created a list of spurious terms in the development corpus, as follows:

1. Add all unique terms of length $= 1$ to the list.
2. For all unique terms of length ≤ 6, manually inspect, and for each:
    - add to the list if it matches a common general language word, a common abbreviation (e.g. *pm*, or *Mr*), or an SI unit;
    - add to the list if it has a numeric component;
    - reject from the list if an obvious technical term;
    - reject from the list if none of the above apply.

This gave a list of 232 terms, which we call the *filter list*. This list was added to Termino, as a list of terms to ignore. The list counters the tendencies of dictionary lookup methods to over-recognise. In use, it performs a similar function to the methods of Hirschman et al. (2002) and Stevenson and Gaizauskas (2000) discussed in the Introduction. Filtering uses no syntactic information, and instead relies on simple heuristics (such as term length), and on knowledge of the domain specificity of terms.

A second list was created at the same time, of terms that were not recognised by Termino, and of special significance according to domain experts. This list consists of 6 terms, mainly of type `Intervention`. This list, called the *supplementary list*, was added to Termino as a list of additional terms to recognise. Neither of these lists took more than a few hours to construct. Their benefits will be demonstrated in the results section.

3.2. Statistical entity recognition

There are many algorithms suitable for statistical entity recognition. We build supervised statistical entity recognition models using SVMs, which have the advantage of good performance over the sparse training data commonly found in text applications. By using SVMs, we are comparing our dictionary based lookup with an approach used in many popular and state of the art systems. We use a variant SVM algorithm, SVM with uneven margins, as provided with the GATE text mining toolkit's Learning API (Li et al., 2005). Kernel parameters were set to those that gave the best results in initial experiments with a pilot corpus, prior to the construction of the corpus used for the experiments reported here.[2] All other GATE Learning API parameters were left at their defaults.

SVMs are binary classifiers, and so different classifiers must be trained to recognise the different entity types. Furthermore, our classifiers apply to individual tokens, and so multi-token entities are recognised using a BE (Begin/End) style of boundary learning. This is handled by the GATE Learning API. A pair of binary classifiers are trained for each entity type: one for the begin (B) token, and one for the end (E) token. For our five entity types, ten binary classifiers are therefore built. Each is applied independently of the others.

For each entity type, post-processing combines pairs of B and E tokens to find the boundaries of candidate entities, according to these rules:

1. each token classified as a B is paired with all following tokens classified as E;
2. a token that is classified as both a B and an E by a pair of classifiers will be considered a candidate single token entity;
3. for overlapping candidates:
    (a) remove those candidates that do not have the same length as any training entity of the same type;
    (b) select the remaining candidate with the maximum confidence, where candidate confidence is the product of confidences calculated from the outputs of the B and E classifiers.

We use a very simple set of token features for our models. Features are constructed for a window of one token on either side of the token being classified.

The features and window size used have been derived by trial of various combinations, and are those that gave the best results (some of this experimentation is currently under review for publication). It is possible that better results can be achieved by extending and tailoring the feature set, but those used give reasonable performance, and are an easily implemented basis for the experiments reported. Our purpose is a comparison of statistical and non-statistical methods, not optimisation of SVMs. The following token features are used:

- Morphological root
- Affix
- Generalised part of speech (POS) category
- Orthographic type (e.g. lower case, upper case)
- Token kind (e.g. number, word)

Most of these features are provided by the standard tokeniser and POS tagger components of the GATE toolkit. The exception is generalised POS category, which is the first two characters of the full POS tag. This takes advantage of the Penn Treebank tagset used by GATE's POS tagger, in which related POS tags share the first two characters. For example, all six verb POS tags start with the letters "VB".

To combine dictionary lookup with statistical entity recognition, we augment token features with a term type feature. If a token is part of a term recognised by Termino, this feature takes the term's type as its value. Otherwise, it is given a value of `null`. The final recognition decision is made by an SVM, using this feature amongst others. Again, we use a window of one token on each side of a candidate token.

[2] Specifically, we used a polynomial kernel with degree 3, cost parameter c of 0.7, and the uneven margins parameter τ set to 0.6.

Entity type	Metric	Termino			SVM + tokens	SVM + tokens + best Termino	IAA
		UMLS	UMLS+filter	UML+filter +supplementary			
Condition	P	0.1971	0.4656	0.4656	0.7994	0.8186	
	R	0.7224	0.7171	0.7171	0.5670	0.6540	
	F1	0.3097	0.5646	0.5646	0.6604	0.7242	0.7504
Drug or device	P	0.2680	0.6224	0.6224	0.7333	0.8301	
	R	0.7308	0.7205	0.7205	0.4433	0.5920	
	F1	0.3922	0.6679	0.6679	0.5456	0.6840	0.7808
Intervention	P	0.2921	0.5158	0.5272	0.8102	0.7500	
	R	0.5582	0.5582	0.6301	0.5753	0.6157	
	F1	0.3835	0.5362	0.5741	0.6504	0.6649	0.5535
Investigation	P	0.1841	0.5438	0.5438	0.8349	0.8308	
	R	0.6941	0.6763	0.6763	0.5608	0.6592	
	F1	0.2910	0.6029	0.6029	0.6671	0.7300	0.7448
Locus	P	0.4453	0.5654	0.5654	0.8057	0.8004	
	R	0.7409	0.7409	0.7409	0.5298	0.6158	
	F1	0.5563	0.6413	0.6413	0.6347	0.6940	0.7925
Overall	P	0.2458	0.5224	0.5238	0.7931	0.8065	
	R	0.6999	0.6939	0.7042	0.5417	0.6308	
	F1	0.3638	0.5961	0.6008	0.6423	0.7071	0.7373

Table 2: Entities found by Termino using UMLS and other term lists; entities found by SVM trained with token features; and entities found by SVM trained with token features plus features from the best Termino configuration. All scored on corpus C77, and shown with inter-annotator agreement for the same corpus.

4. Evaluation

Evaluation metrics are defined in terms of true positive, false positive and false negative matches between entities in a system annotated *response* document and a gold standard *key* document. A response entity is a true positive if an entity of the same type, and with the exact same text span, exists in the key. Matching of response entities to key entities is therefore strict (i.e. overlapping key and response entities do not contribute to scoring). Corresponding definitions apply for false positive and false negative. Counts of these matches are used to calculate standard metrics of Recall (R), Precision (P) and $F1$ measure.

As Termino does not need any gold standard training data, evaluation of Termino is by a direct comparison of the gold standard entities to the terms matched by Termino, assuming that each term matched corresponds to an entity. For Termino, we report metrics for entity types macro-averaged across all documents. As our statistical entity recognition is supervised, we need the gold standard for training data. We have therefore trained and evaluated using ten fold cross-validation, with metrics macro-averaged over all ten folds.

The metrics do not say how hard entity recognition is: there is nothing against which to compare the system. We therefore provide Inter Annotator Agreement (IAA) scores from the gold standard. The IAA score gives the agreement between the two independent double annotators. It is equivalent to scoring one annotator against the other using the $F1$ metric (Hripcsak and Rothschild, 2005). Note that the measure compares two human annotators. As the system is trained on a third *consensus* annotation, the IAA does not give an upper bound on performance. It is possible for the system to score a higher $F1$ than the IAA for the same entity type.

5. Results

5.1. Dictionary Lookup

The first set of experiments looked at various configurations of Termino, with and without filter terms and supplementary terms. These show the performance of simple dictionary lookup based on UMLS, and of dictionary lookup tailored with additional cheaply constructed lists. The results of these experiments are reported on the left of Table 2. The table shows that Termino loaded with just UMLS gave a recall of > 0.69 for all entity types except `Intervention` at 0.55. Precision, however, was low at between 0.18 and 0.47. Overall precision was 0.25. Error analysis with our development corpus showed that the low precision was due to the large amount of ambiguity inherent in such large scale resources, as discussed above. The second column shows the effect of using a filter term list to disallow these common spurious matches. Precision more than doubles in most cases, to 0.52 overall. Recall drops by less than 2% in all cases, not changing at all in some. The filter list clearly makes a big difference to performance. The terms that it removes are almost always spurious, and rarely genuine.

We also added a small list of terms considered important by domain experts in the CLEF project, but not included in UMLS. The results for Termino with this list included are show in the third column. The supplementary list only has an effect on `Intervention`, where recall increases by $> 7\%$. This is clearly significant in the case of a specific entity type, but has little overall impact ($< 0.5\%$ increase in overall $F1$)

With both lists added, Termino achieves an $F1$ around 10% to 20% below IAA for most entity types. The exception to this is `Intervention`, which is $> 1.5\%$ above the IAA. `Intervention` has the lowest IAA, 0.55, indicating that is difficult for human annotators to reach agreement on this entity type. This difficulty is reflected by the fact that a dictionary lookup performs just as well.

5.2. Statistical Models

The second set of experiments looked at SVM entity learning. The first of these experiments used simple token features. The second experiment combined simple token features with a Termino feature, as described above, in Section 3.2. The results of these experiments are also reported in Table 2.

The **SVM + tokens** column in Table 2 shows the performance of a system trained with no terminological knowledge. The only features used were those that described the surface form of the token (e.g. string and orthography), and its POS. For each entity type, recall is below that of the best Termino system, with differences in the range 5% to 28%. These results show that Termino contains a reasonable proportion of the entity terms appearing in the gold standard, and this will presumably also be true for the remainder of the corpus. The SVM, on the other hand, is limited to build a model only based on terms annotated in the gold standard. Turning to precision, we find that it is 10% to 30% above that of the best Termino system. Despite being limited in its scope, the model that the SVM does build is accurate, avoiding the ambiguity from which dictionary lookup with Termino suffers. The increase in precision is not mirrored exactly by a drop in recall: $F1$ does not stay the same for all entities. While for most, $F1$ is higher with the SVM, for `Locus` it is slightly lower, and for `Drug or device` it is 12% lower, showing that higher precision is at the expense of a much bigger drop in recall than for other entity types. Dictionary lookup appears to be especially useful in this case.

The **SVM + tokens + best Termino** column of Table 2 shows the SVM with term features added to the previous token features. A feature is added that records whether a dictionary lookup term coincides with a token. The features used were from the best Termino, using UMLS, filter terms, and supplementary terms. The most consistent trend over SVM with token features only, is an increase in recall, of between 4% and 15%. The additional terminological information has presumably enabled the SVM to build a more general model that is able to exploit the broader knowledge that Termino contributes. While precision also improves, overall ($> 1\%$), the improvement is not so clear cut. For two entity types (`Locus` and `Investigation`, it drops very slightly. For `Intervention`, it drops by 6%. While generally good, the SVM has not always been able to overcome the ambiguity inherent in dictionary lookup. In terms of $F1$, SVM with token and Termino features consistently outperforms SVM with token features only, by around 6% overall.

Across all systems, SVM with token and Termino features performs best, with $F1$ 3% to 10% below IAA (and in one case, `Intervention`, 11% above). The combined systems manages to gain from the higher recall of dictionary lookup, while not suffering from a loss in the precision of the statistical method.

5.3. Linkage of Entities to External Resources

An advantage of dictionary-based term recognition over statistical methods is that a dictionary-based system such as Termino can provide entry points into the source terminologies and ontologies. These entry points make the information from the external resources available for further text processing steps, for querying, and for other applications. Can this advantage be carried through to the combined dictionary-lookup and statistical method?

In Termino, entry points to source terminologies and ontologies are implemented by annotating each term with unique identifiers for entries in those resources. In the case of the UMLS, this is a *Concept Unique Identifier*, or CUI. In the Termino-only system, every term found will have at least one CUI. Some terms will be ambiguous in UMLS, and may have more than one CUI. For example, the term *chemotherapy* is ambiguous between a type of drug therapy, and a course of treatment (*chemotherapy regimen* elided).

In the combined system, there will be an overlap between entities found by the SVM and those found using Termino terms alone. Some terms will have been found by Termino but rejected as entities by the SVM, some terms found by Termino and confirmed as entities by the SVM, and other entities will have been found by the SVM alone. As the SVM is the final arbiter in the combined system, these last two groups make up those entities ultimately recognised. We assign CUIs to entities from the combined system where Termino has also found a term at the same point in the text. The Termino term must also have the same type as the entity recognised by the SVM: for example, there is no point in assigning a CUI for a `Locus` term found by Termino, to a `Condition` entity found by the SVM.

We tested CUI assignment in the combined system, by training the system on all 77 gold standard documents, and applying it to our development corpus. The numbers of CUIs assigned are shown in Table 3. Overall, 83% of all entities were assigned at least one CUI. Only Intervention had more than 20% of entities assigned no CUIs. By this measure, it does seem that the linkage provided by a dictionary-based method is carried through to the combined method. However, there are two problems with this result. First, we cannot be sure of the precision of CUI assignment, as our gold standard does not contain CUIs. It seems likely, however, that as CUI assignment is based on a direct lookup on UMLS terms, precision will be high. Second, a considerable number of entities had more than one CUI assigned: nearly 27% overall. Most of these were assigned two, but a small number were assigned 3 or more. Clearly, some form of disambiguation is needed — this would also be true of a pure Termino approach. CUI assignment may be viewed as a form of word sense disambiguation, a topic reviewed by Schuemie et al. (2005) for the biomedical domain.

6. Conclusion

We have examined entity recognition using dictionary lookup, and using machine learning of statistical models with SVMs. Dictionary lookup based on a very large terminology resource gave good recall, but poor precision. The low precision was largely due to the ambiguity inherent in such terminology resources. We found that much of the ambiguity was due to a small number of terms, and that filtering these out doubled precision. The filter list was hand built using simple heuristics, and used no syntactic information.

SVM based entity recognition, trained on lexico-syntactic features alone, outperformed dictionary lookup in terms of precision, but gave lower recall. In terms of $F1$, the SVM system outperformed dictionary lookup overall, but was much worse for one entity type (`Drug or device`), suggesting that dictionary lookup is especially useful in some cases.

Entity type		CUIs assigned						
		0	1	2	3	4	5	> 0
Condition	Number	40	180	54	14	1	2	251
	%	13.75	61.86	18.56	4.81	0.34	0.69	86.25
Drug or device	Number	10	101	8	1	0	0	110
	%	8.33	84.17	6.67	0.83	0	0	91.67
Intervention	Number	47	21	55	0	0	0	76
	%	38.21	17.07	44.72	0	0	0	61.79
Investigation	Number	20	68	36	5	0	0	109
	%	15.50	52.71	27.91	3.88	0	0	84.50
Locus	Number	29	116	37	11	5	1	170
	%	14.57	58.29	18.59	5.53	2.51	0.50	85.43
Total	Number	146	486	190	31	6	3	716
	%	16.94	56.38	22.04	3.60	0.70	0.35	83.06

Table 3: Numbers of external resource identifiers (UMLS CUIs) assigned to terms found in a development corpus of 50 documents, by a combined SVM and Termino system.

When the SVM was combined with dictionary lookup, by training on term features in addition to the lexico-syntactic features, precision was maintained overall, although it did drop for specific entity types. Recall improved significantly in all cases, although it did not attain the overall recall levels of the best dictionary lookup. This system gave the best overall $F1$ of 0.71, 3% below the overall Inter Annotator Agreement. The combined system also retained an advantage of dictionary lookup, by achieving linkage from recognised entities to domain resources in 83% of cases.

We have shown that large scale terminology resources can be used to benefit clinical entity recognition, and that statistical models can overcome some of the shortcomings of dictionary lookup over such resources.

Availability Most of the software described here is open source and can be downloaded as part of GATE. We are currently packaging Termino for public release, at which point the whole application will be made available.

Acknowledgements

CLEF is funded by the UK Medical Research Council. We would like to thank the Royal Marsden Hospital for providing the corpus, and our clinical partners in CLEF for assistance in developing the annotation schema, and for gold standard annotation.

7. References

S. Ananiadou and G. Nenadic. 2006. Automatic terminology management in biomedicine. In S. Ananiadou and J. McNaught, editors, *Text Mining for Biology and Biomedicine*, chapter 4, pages 67–97. Artech House.

A. Aronson. 2005. Filtering the umls metathesaurus for metamap. Technical report, U.S National Library of Medicine, Lister Hill National Center for Biomedical Communications, Cognitive Science Branch.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, pages 168–175, Philadelphia, PA, USA.

H. Harkema, R. Gaizauskas, M. Hepple, N. Davis, Y. Guo, A. Roberts, and I. Roberts. 2004. A Large-Scale Resource for Storing and Recognizing Technical Terminology. In *Proceedings of 4th International Conference on Language Resources and Evaluation*, pages 83–86, Lisbon, Portugal.

L. Hirschman, A. Morgan, and A. Yeh. 2002. Rutabaga by any other name: extracting biological names. *Journal of Biomedical Informatics*, 35(4):247–259.

G. Hripcsak and A. Rothschild. 2005. Agreement, F-measure and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298.

Y. Li, K. Bontcheva, and H. Cunningham. 2005. SVM based learning system for information extraction. In *Deterministic and statistical methods in machine learning: first international workshop*, number 3635 in Lecture Notes in Computer Science, pages 319–339. Springer.

D. Lindberg, B. Humphreys, and A. McCray. 1993. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291.

A. McCray, O. Bodenreider, J. Malley, and A. Browne. 2001. Evaluating UMLS Strings for Natural Language Processing. In *Proceedings of the 2001 American Medical Informatics Association Annual Symposium*, pages 448–452, Portland, OR, USA.

A. McCray, A. Browne, and O. Bodenreider. 2002. The lexical properties of the gene ontology (go). In *Proceedings of the 2002 American Medical Informatics Association Annual Symposium*, pages 504–508, San Antonio, TX, USA.

S. Mika and B. Rost. 2004. Protein names precisely peeled off free text. *Bioinformatics*, 20(Supplement 1):i241–i247.

A. Mikheev, M. Moens, and C. Grover. 1999. Named Entity recognition without gazetteers. In *Proceedings of the ninth conference of the European chapter of the Association for Computational Linguistics (EACL'99)*, pages 1–8, Bergen, Norway.

S. Pakhomov, J. Buntrock, and P. Duffy. 2005. High throughput modularized NLP system for clinical text. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), interactive poster and demonstration sessions*, pages 25–28, Ann Arbor, MI, USA.

J. Park and J. Kim. 2006. Named entity recognition. In S. Ananiadou and J. McNaught, editors, *Text Mining for Biology and Biomedicine*, chapter 6, pages 121–142. Artech House.

D. Proux, F. Rechenmann, L. Julliard, V. Pillet, and B. Jacq. 1998. Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. *Genome Informatics*, 9:72–80.

A. Rector, J. Rogers, A. Taweel, D. Ingram, D. Kalra, J. Milan, P. Singleton, R. Gaizauskas, M. Hepple, D. Scott, and R. Power. 2003. CLEF — Joining up Healthcare with Clinical and Post-Genomic Research. In *Proceedings of UK e-Science All Hands Meeting 2003*, pages 264–267, Nottingham, UK.

A. Roberts, R. Gaizauskas, M. Hepple, N. Davis, G. Demetriou, Y. Guo, J. Kola, I. Roberts, A. Setzer, A. Tapuria, and B. Wheeldin. 2007. The CLEF Corpus: Semantic Annotation of Clinical Text. In *Proceedings of the 2007 American Medical Informatics Association Annual Symposium*, pages 625–629, Chicago, IL, USA.

M. Schuemie, J. Kors, and B. Mons. 2005. Word sense disambiguation in the biomedical domain: An overview. *Journal of Computational Biology*, 12(5):554–565.

M. Stevenson and R. Gaizauskas. 2000. Using Corpus-derived Name Lists for Named Entity Recognition. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP-2000)*, pages 84–89, Seattle, Washington, USA.

L. Tanabe and W. Wilbur. 2002. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132.

K. Yamamoto, T. Kudo, A. Konagaya, and Y. Matsumoto. 2003. Protein name tagging for biomedical annotation in text. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 65–72, Sapporo, Japan.