BaseNPs that contain gene names: domain specificity and genericity

Ian Lewin
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
ian.lewin@cl.cam.ac.uk

Abstract

The names of *named entities* very often occur as constituents of larger noun phrases which denote different types of entity. Understanding the structure of the embedding phrase can be an enormously beneficial first step to enhancing whatever processing is intended to follow the named entity recognition in the first place. In this paper, we examine the integration of general purpose linguistic processors together with domain specific named entity recognition in order to carry out the task of baseNP detection. We report a best F-score of 87.17% on this task. We also report an inter-annotator agreement score of 98.8 Kappa on the task of baseNP annotation of a new data set.

1 Introduction

Base noun phrases (baseNPs), broadly "the initial portions of non-recursive noun phrases up to the head" (Ramshaw and Marcus, 1995), are valuable pieces of linguistic structure which minimally extend beyond the scope of named entities. In this paper, we explore the integration of different techniques for detecting baseNPs that contain a named entity, using a domain-trained named entity recognition (NER) system but in combination with other linguistic components that are "general purpose". The rationale is simply that domain-trained NER is clearly a necessity for the task; but one might expect to be able to secure good coverage at the higher syntactic level by intelligent integration of general purpose syntactic processing without having to undergo a further round of domain specific annotation and training. We present a number of experiments exploring different ways of integrating NER into general purpose linguistic processing. Of course, good results can also be used subsequently to help reduce the effort required in data annotation for use in dedicated domain-specific machine learning systems for baseNP detection.

First, however, we motivate the task itself. Enormous effort has been directed in recent years to the automatic tagging of named entities in bio-medical texts and with considerable success. For example, iHOP reports gene name precision as being between 87% and 99% (depending on the organism) (Hoffman and Valencia, 2004). Named entities are of course only sometimes identical in scope with noun phrases. Often they are embedded within highly complex noun phrases. Nevertheless, the simple detection of a name by itself can be valuable. This depends in part on the intended application. Thus, iHOP uses gene and protein names to hyperlink sentences from Medline and this then supports a browser over those sentences with additional navigation facilities. Clicking on *Dpp* whilst viewing a page of information about *hedgehog* leads to a page of information about *Dpp* in which sentences that relate both *Dpp* and *hedgehog* are prioritized.

One of the application advantages of iHOP is that the discovered gene names are presented to the user in their original context and this enables users to compensate for problems in reliability and/or contextual relevance. In many Information Extraction (IE) systems, relations between entities are detected and extracted into a table. In this case, since the im-

BioNLP 2007: Biological, translational, and clinical language processing, pages 163–170,
Prague, June 2007.

mediate surrounding context of the gene name may be simply lost, the reliability of the original identification becomes much more important. In section 2 below, we explain our own application background in which our objective is to increase the productivity of human curators whose task is to read particular scientific papers and fill in fields of a database of information about genes. Directing curators' attention to sentences which contain gene names is clearly one step. Curators additionally report that an index into the paper that uses the gene name and its embedding baseNP is even more valuable (reference omitted for anonymity). This often enables them to predict the possible relevance of the name occurrence to the curation task and thus begin ordering their exploration of the paper. Consequently, our technical goal of baseNP detection is linked directly to a valuable application task. We also use the baseNP identification in order to type the occurrence semantically and use this information in an anaphora resolution process (Gasperin, 2006).

The detection of baseNPs that contain a named entity is a super-task of NER, as well as a sub-task of NP-chunking. Given that NER is clearly a domain specific task, it is an interesting question what performance levels are achievable using domain trained NER in combination with general purpose linguistic processing modules.

There is a further motivation for the task. The distinction between a named entity and an embedding noun phrase is one with critical importance even for the sub-task of NER. Dingare et al (2005) conclude, from their analysis of a multi-feature maximum entropy NER module, that increases in performance of biomedical NER systems will depend as much upon qualitative improvements in annotated data as in the technology underlying the systems. The claim is that quality problems are partly due to confusion over what lies in the scope of a named entity and what lies at higher syntactic levels. Current biomedical annotations are often inconsistent partly because annotators are left with little guidance on how to handle complexities in noun phrases, especially with respect to premodifiers and conjunctions. For example, which premodifiers are part of the *named entity* and which are "merely" part of the embedding noun phrase? Is *human* part of the named entity in *the regulation of human interleukin-2 gene expression*,

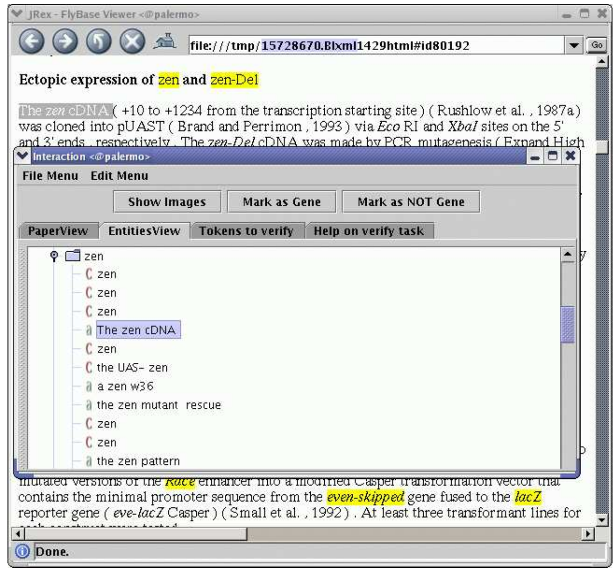

Figure 1: Paper Browser showing baseNP index

or not?

By focussing attention instead on the baseNPs that contain a named entity, one can clearly sidestep this issue to some extent. After all, increasing the accuracy of an NER module with respect to premodifier inclusion is unlikely to affect the overall accuracy of detection of the embedding noun phrases.

2 FlyBase curation

The intended application for our work is a software environment for FlyBase curators that includes an NLP-enhanced Browser for Scientific Papers. FlyBase is the world's leading genomics database for the fruitfly Drosophila (melanogaster and other species) (Crosby et al., 2007). FlyBase is largely updated through a paper-by-paper methodology in which research articles likely to contain information relevant for the FlyBase database are first put in a priority list. Subsequently, these are read by skilled geneticists (at post-doctoral level) who distil gene related information into the database itself. Although this is a paradigm example of IE, our objective is not to fully automate this task itself, simply because the expected accuracy rates are unlikely to be high enough to provide a genuinely useful tool. Rather, our task is to enable curators to explore the gene related sections of papers more efficiently. The Browser currently highlights potential items of interest for curators and provides novel indexing and navigation possibilities. It is in this context that the identification of baseNPs that contain gene names is carried out. An individual sentence that contains a gene name is very often not enough, considered in isolation, for curators to fill in a required database field. Information often needs to be gathered from across a paragraph and even the whole paper. So extraction of sentences is not an attractive option. Equally, a whole sentence is unfeasibly large to serve simply as an indexing term into the paper. Noun phrases provide more information than simply gene names, but post-modification can also lead to extremely long terms. BaseNPs are therefore a useful compromise, these being short enough to display whole in a window (i.e. no scrolling is required) and often bearing enough information for the user to understand much more of the context in which the gene name itself appears. Furthermore, the baseNP is both a natural "unit" of information (whereas a window of n tokens around a gene name is not) and it supports further processing. BaseNPs are typed according to whether they denote genes or various gene products and linked together in anaphoric chains.

In our navigation panel for the Browser, the baseNPs are sorted according to the gene name that they contain (and then by order in which they appear within the paper), and hyperlinked to their occurrence in the paper. This enables users to explore papers gene-by-gene but also, when considering a particular gene, to understand more about the reference to the gene - for example whether gene products or promoters are being referenced. Figure 1 contains an example screenshot.

3 Scope of the Data

Complex nominals have long been held to be a common feature in scientific text. The corpus of Vlachos and Gasperin (2006) contains 80 abstracts (600 sentences) annotated with gene names. In this data-set, noun phrases that contain gene names (excluding post-modifiers) of 3 words or more comprise more than 40% of the data and exhibit primarily: strings of premodifiers *tudor mutant females*, *zygotic Dnop5 expression*; genitives: *Robo 's cytoplasmic domain*, *the rdgB protein 's amino terminal 281 residues*; coordination *the copia and mdg-1 elements* and parenthetical apposition *the female-specific gene Sex lethal (Sxl)*, and *the SuUR (suppressor of underreplication) gene*. Only 41% of the baseNPs containing a gene name consist of one token only. 16% have two tokens. The two token baseNPs include large numbers of combinations of gene names with more general words such as *Ras activity*, *vnd mutants*, *Xiro expression*, *IAP localization* and *vasa protein*. In general, the gene name appears in modifier position although species modifiers are common, such as *Drosophila Tsg*, and there are other possibilities: *truncated p85*.

Our intention is to categorize this data using the concept of "baseNP" and build effective computational models for recognizing instances. Although baseNP is a reasonably stable linguistic concept, its application to a new data-set is not completely straightforward. Ramshaw and Marcus (1995) state that a baseNP aims "to identify essentially the initial portions of nonrecursive noun phrases up to the head, including determiners but not including postmodifying prepositional phrases or clauses". However, work on baseNPs has essentially always proceeded via algorithmic extraction from fully parsed corpora such as the Penn Treebank. BaseNPs have therefore depended on particular properties of the annotation framework and this leads to certain aspects of the class appearing unnatural.

The clearest case is single element conjunction, which Penn Treebank policy dictates is annotated at word-level with a flat structure like this *[lpl and xsl]* (brackets indicate baseNP boundaries). As soon as one of the elements is multi-word however, then separate structures are to be identified *[lpl] and [the sxl gene]*. The dependency on numbers of tokens becomes clearly problematic in the bio-medical domain. Quite different structures will be identified for *lpl and fasciclin*, *lpl and fasciclin 1* and possibly *lpl and fasciclin-1*, depending on how tokenization treats hyphens. Furthermore, nothing here depends on the motivating idea of "initial segments up to the head". In order to provide a more natural class, our guidelines are that unless there is a shared modifier to account for (as in *[embryonic lgl and sxg]*), all coordinations are split into separate baseNPs. All other cases of coordination follow the standard guidelines of the Penn Treebank.

A second difficult case is possessives. BaseNP extraction algorithms generally split possessives like this: [fra] ['s ectodomain], corresponding (somewhat) to an intuition that there are two NPs whilst assigning each word to some baseNP chunk and not introducing recursiveness. This policy however causes a sharp division between this case and *the fra ectodomain* following the Penn Treebank bracketing guideline that nominal modifiers are never labelled. Since our interest is "the smallest larger NP containing a gene name", we find it much more natural to treat *fra's* as just another modifier of the head *ectodomain*. Whether it recursively contains a single word NP *fra* (or just a single word NNP) is again not something that is motivated by the idea of "initial segments up to the head". Similarly, we mark one baseNP in *the rdgB protein's amino terminal 281 residues*, viz. *the rdgB protein*.

Apposition, as in *Sex lethal (Sxl)* and *the gene sex lethal* , is a further interesting case. In the first case, "Sex lethal" and "Sxl" stand in apposition. Both are gene names. The former is the head. In the second, "gene" is the head and "sex lethal" is a name that stands in apposition. In each case, we have a head and post-modifiers which are neither clausal nor prepositional. It is unclear whether the rubric "clausal or prepositional" in Ramshaw and Marcus' statement of intent is merely illustrative or definitive. On the grounds that a sharp division between the non-parenthetical case *the gene sex lethal* and the pre-modifier *the sex lethal gene* is unnatural, our intuition is that the baseNP does cover all 4 tokens in this case. All (post-head) parentheticals are however to be treated more like optional adjuncts and therefore not included with the head to which they attach.

In order to verify the reliability of baseNP annotation, two computational linguists (re)annotated the 600 sentences (6300 tokens) of Vlachos and Gasperin (2006) with baseNPs and heads using the published guidelines. We added material concerning head annotation. Vlachos and Gasperin did not quote agreement scores for baseNP annotation. Their interest was directed at gene name agreement between a linguist and a biologist. Our 2-person inter-annotator Kappa scores were 0.953 and 0.988 on head and baseNP annotation respectively representing substantial agreement.[1] .

4 Methodology

A reasonable and simple baseline system for extracting baseNPs that contain a gene name is to use an off-the-shelf baseNP extractor and simply filter the results for those that contain a gene name. To simplify analysis of results, except where otherwise noted this filter and subsequent uses of NER are based on a gold standard gene name annotation. In this way, the contributions of different components can be compared without factoring in relative errors of NER. Naturally, in the live system, an automated NER process is used (Vlachos and Gasperin, 2006). For the baseline we chose an implementation of the Ramshaw and Marcus baseNP detector distributed with GATE[2] pipelined with the Stanford maximum entropy part of speech tagger [3]. The Stanford tagger is a state of the art tagger incorporating a number of features including use of tag contexts, lexical features, a sophisticated smoothing technique, and features for unknown words (including 4-gram prefixes and suffixes). Both components of the baseline systems utilize the 48 tag Penn Treebank tagset. Results however showed that poor performance of the part of speech tagger could have a disastrous effect on baseNP detection. A simple extension of the baseline is to insert a module in between POS tagging and NP detection. This module revises the POS tags from the tagger in the light of NER results, essentially updating the tags of tokens that are part of named entities. This is essentially a simple version of the strategy mooted by Toutanova at el (2003) that the traditional order of NER and tagging be reversed. It is simpler because, in a maximum entropy framework, NER results can function as one extra feature amongst many in POS detection; whereas here it functions merely as an override. Retraining the tagger did not form part of our current exploration.

[1] In fact, although the experiment can be considered a classification of 6300 tokens in IOB format, the counting of classifications is not completely straightforward. The task was "annotate the baseNP surrounding each gene name" rather than "annotate each token". In principle, each token is examined; in practice a variable number is examined. If we count all tokens classified into NPs plus one token of context either side, then both annotators annotated over 930 tokens.

[2] http://www.gate.ac.uk

[3] http://nlp.stanford.edu/software/tagger.shtml

We adopted a similar strategy with the domain independent full parsing system RASP (Briscoe et al., 2006). RASP includes a simple 1st order HMM POS tagger using 149 of the CLAWS-2 tagset. The tagger is trained on the manually corrected subsets of the (general English) Susanne, LOB and BNC corpora. The output of the tagger is a distribution of possible tags per token (all tags that are at least 1/50 as probable as the top tag; but only the top tag if more than 90% probable). The tagger also includes an unknown word handling module for guessing the possible tags of unknown words. The RASP parser is a probabilistic LALR(1) parser over the CLAWS-2 tags, or, more precisely, a unification grammar formalism whose lexical categories are feature based descriptions of those tags. The parser has no access to lexical information other than that made available by the part of speech tags. Although the output of RASP is a full parse (or a sequence of fragments, if no connected parse can be found) and baseNPs may not be constituents of NPs, baseNPs can be extracted algorithmically from the full parse.

Some more interesting pre-parsing integration strategies are available with RASP because it does not demand a deterministic choice of tag for each word. We experimented with both a deterministic re-write strategy (as for the baseline system) and with various degrees of interpolation; for example, adjusting the probability distribution over tags so that proper noun tags receive 50% of the probability mass if the token is recognized by NER, and the other tags receive the remaining 50% in direct proportion to the amount they would receive from the POS tagger alone. In this set-up, the NER results need not function simply as an override, but equally they do not function simply as a feature for use in part of speech tagging. Rather, the parser may be able to select a best parse which makes use of a sequence of tags which is not itself favoured by the tagger alone. This allows some influence to the grammatical context surrounding the gene name and may also permit tags within phrasal names such as *transforming growth factor* to propagate.

RASP is also a non-deterministic parser and consequently a further possible integration strategy is to examine the output n-best list of parses to find baseNPs, rather than relying on simply the 1-best output. The n-best parses are already scored according to a probabilistic model trained on general text. Our strategy is to re-score them using the additional knowledge source of domain specific NER. We explored a number of re-scoring hypotheses. First, a cut-off of 20 on n-best lists was found to be optimal. That is, correct analyses tended to either be in the top 20 or else not in the top 100 or even 1000. Secondly, differences in score between the incorrect 1-best and the correct nth hypothesis were not a very reliable indicator of "almost right". This is not surprising as the scores are probabilities calculated over the complete analysis, whereas our focus is one small part of it. Consequently, the re-scoring system uses the probabilistic model just to generate the top 20 analyses; and those analyses are then re-scored using 3 features. Analyses that concur with NER in having a named entity within an NP receive a reward of +1. Secondly, NP analyses that contain N+1 genes (as in a co-ordination) receive a score of +N, so long as the NP is single headed. For example, "gurken or torpedo females" will receive a preferred analysis in which "gurken" and "torpedo" are both modifiers of "females". The "single headedness" constraint rules out very unlikely NP analyses that the parser can return as legal possibilities. Finally, analyses receive a score of -1 if the NP contains a determiner but the head of the NP is a gene name. The top 20 parses may include analyses in which, for example, "the hypothesis that phenylalanine hydroxylase" contains "that phenylalanine hydroxylase" as an NP constituent.

Finally, we also experimented with using both the full parsing and shallow baseNP spotter together; here, the idea is simply that when two analyses overlap, then the analysis from full parsing should be preferred on the grounds that it has more information available to it. However, if the shallow spotter detects an analysis when full parsing detects none then this is most likely because full parsing has been led astray rather than it has discovered a more likely analysis not involving any baseNP.

5 Experimental Results

Table 1 gives the precision, recall and (harmonic) F-score measures for the baseline NP system with and without the extra pre-parsing retagging module; and table 2 gives similar figures for the generic full parsing system. Scores for the left boundary only, right boundary only and full extent ('correct') are shown. The extra retagging module (i.e. override tagger results, given NER results) improves results in both systems and by similar amounts. This is nearly always on account of gene names being mis-tagged as verbal which leads to their exclusion from the set of baseNP chunks. The override mechanism is of course a blunt instrument and only affects the tags of tokens within gene names and not those in its surrounding context.

Table 3 shows the results from interpolating the POS tag distribution P with the NER distribution N linearly using different levels of λ. For example, $\lambda = 1.00$ is the simple retagging approach in which all the probability is assigned to the NER suggested tag; whereas $\lambda = 0.25$ means that only 25% is allocated by NER. The figures shown are for one variant of the full parsing system which included n-best selection but other variants showed similar behaviour (data not shown). The results from interpolation show that the extra information available in the parse does not prove valuable overall. Decreasing values of λ lead to decreases in performance. These results can be interpreted as similar in kind to Charniak et al (1996) who found that a parser using multiple POS tag inputs could not improve on the *tag accuracy* of a tagger outputting single POS tags. Our results differ in that the extra tag possibilities are derived from an alternative knowledge source and our measurement is baseNP detection. Nevertheless the conclusion may be that the best way forward here is a much tighter integration between NER and POS tagging itself.

POS tagging errors naturally affect the performance of both shallow and full parsing systems, though not necessarily equally. For example, the tagger in the shallow system tags *ectopic* as a verb in *vnd-expression leads to ectopic Nk6 expression* and this is not corrected by the retagging module because *ectopic* is not part of the gene name. Consequently the baseNP spotter is led into a left boundary error. Nevertheless, the distribution of baseNPs from the two systems do appear to be complementary in a rather deeper fashion. Analysis of the results indicates that parentheticals in pre-modifier positions appears to throw the shallow parser severely off course. For example, it generates the analysis

	R	P	F
retag+shallow			
(correct)	80.21	75.92	78.01
(left b)	92.40	87.46	89.86
(right b)	90.81	85.95	88.32
shallow only			
(correct)	74.03	76.32	75.16
(left b)	84.28	86.89	85.56
(right b)	82.69	85.25	83.95

Table 1: Generic shallow parsing

	R	P	F
retag+full			
(correct)	80.92	84.81	82.82
(left b)	85.69	89.81	87.70
(right b)	88.69	92.96	90.78
full only			
(correct)	75.44	85.23	80.04
(left b)	80.21	90.62	85.10
(right b)	82.51	93.21	87.54

Table 2: Generic full parsing

[the transforming growth factor-beta] ([TGF-beta]) superfamily. Also, appositions such as *the human auto antigen La* and *the homeotic genes abdominal A and abdominal B* cause problems. In these kinds of case, the full parser detects the correct analysis. On the other hand, the extraction of baseNPs from grammatical relations relies in part on the parser identifying a head correctly (for example, via a non-clausal subject relation). The shallow parser does not however rely on this depth of analysis and may succeed in such cases. There are also cases where the full parser fails to detect any analysis at all.

System	(correct)	(left b)	(right b)
λ=0.25	83.97	88.34	90.71
λ=0.50	84.16	88.69	91.22
λ=0.80	85.18	89.67	91.28
λ=1.00	85.38	89.87	91.66

Table 3: F-scores for baseNP detection for various λ

Table 4 indicates the advantages to be gained in n-best selection. The entries for *full* and *retag+full* are repeated from table 2 for convenience. The entries

System	R	P	F
retag+full	80.92	84.81	82.82
retag+full+sel	83.22	87.22	85.17
retag+full+oracle	85.87	90.17	87.96
full	75.44	85.23	80.04
full+sel	78.80	86.60	82.52
full+oracle	81.63	89.88	85.56

Table 4: Effects of n-best selection

System	R	P	F
1-best	85.69	84.35	85.01
n-best	87.63	86.71	87.17
oracle	90.28	89.49	89.89

Table 5: Combining shallow and full parsing

for *full+sel* and *retag+full+sel* show the effect of adding n-best selection. The entries for *full+oracle* and *retag+full+oracle* show the maximum achievable performance by replacing the actual selection policy with an oracle that always chooses the correct hypothesis, if it is available. The results are that, regardless of whether a retagging policy is adopted, an oracle which selects the best analysis can achieve an error reduction of well over 25%. Furthermore, the simple selection policy outlined before succeeds in achieving almost half the possible error reduction available. This result is particularly interesting because it demonstrates that the extra knowledge source available in this baseNP detection task (namely NER) can profitably be brought to bear at more than one stage in the overall processing pipeline. Even when NER has been used to improve the sequence of POS tags given to the parser, it can profitably be exploited again when selecting between parses.

The complementary nature of the two systems is revealed in Table 5 which shows the effects of integrating the two parsers. baseNPs from the shallow parser are accepted whenever it hypothesizes one and there is no competing overlapping baseNP from the full parser. Note that this is rather different from the standard method of simply selecting between an analysis from the one parser and one from another. The success of this policy reflects the fact that there remain several cases where the full parser fails to deliver "apparently" simple baseNPs either because the tagger has failed to generate a suitable hypothesis, or because parsing itself fails to find a good enough analysis in the time available to it.

Overall, the best results (87.17% F-score) are obtained by applying NER results both before parsing through the update of POS tags and after it in selection from n-best lists; and by combining the results of both full parsing in order to improve analysis of more complex structures and shallow parsing as a back-off strategy. The same strategy applied using our automated gene name recognizer results in a F-score of 73.6% F-score, which is considerably less of course, although the gene name recognizer itself operates at 82.5% F-Score, with similar precision and recall figures. This naturally limits the possible performance of our baseNP recognition task. Encouragingly, the "lost" performance (just under 11%) is actually less in this scenario than when gene name recognition is perfect.

6 Previous Work

The lack of clarity between noun phrase extents and named entity extents and its impact on evaluation and training data for NER has been noted previously, e.g. for proteins (Mani et al., 2005). Vlachos and Gasperin (2006) claim that their name versus mention distinction was helpful in understanding disagreements over gene name extents and this led, through greater clarity of *intended* coverage, to improved NER. BaseNP detectors have also been used more directly in building NER systems. Yamamoto et al (2003) describe an SVM approach to protein name recognition, one of whose features is the output of a baseNP recognizer. BaseNP recognition supplies a top-down constraint for the search for protein names within a baseNP. A similar approach albeit in a CRF framework is described in Song et al. (2005).

The concept of baseNP has undergone a number of revisions (Ramshaw and Marcus, 1995; Tjong Kim Sang and Buchholz, 2000) but has previously always been tied to extraction from a more completely annotated treebank, whose annotations are subject to other pressures than just "initial material up to the head". To our knowledge, our figures for inter-annotator agreement on the baseNP task itself (i.e. not derived from a larger annotation task) are the first to be reported. Quality measures can be indirectly inferred from a treebank complete annotation, but baseNP identification is probably a simpler task. Doddington et al (2004) report an "overall value score of 86" for inter-annotator agreement in ACE; but this is a multi-component evaluation using a complete noun phrase, but much else besides.

Improving results through the combination of different systems has also been a topic of previous work in baseNP detection. For example, Sang et al (2000) applied majority voting to the top five machine learning algorithms from a sample of seven and achieved a baseNP recognition rate that exceeded the recognition rates of any of the individual methods.

7 Conclusion

We have motivated the task of detecting baseNPs that contain a given named entity as a task both of interest from the standpoint of use within a particular application and on more general grounds, as an intermediate point between the task of general NP chunking and domain specific NER.

We have explored a variety of methods for undertaking baseNP detection using only domain specific NER in addition to otherwise general purpose linguistic processors. In particular, we have explored both shallow and full parsing general purpose systems and demonstrated that the domain specific results of NER can be applied profitably not only at different stages in the language processing pipeline but also more than once. The best overall recognition rates were obtained by a combination of both shallow and full parsing systems with knowledge from NER being applied both before parsing, at the stage of part of speech detection and after parsing, during parse selection.

References

E.J. Briscoe, J. Carroll, and R. Watson. 2006. The second release of the rasp system. *Proc. Coling/ACL 2006 Interactive Sessions*.

E. Charniak, G. Carroll, J. Adcock, A.R. Cassandra, Y. Gotoh, J. Katz, M.L. Littman, and J. McCann. 1996. Taggers for parsers. *Artificial Intelligence*, 85(1-2):45–57.

M.A. Crosby, Goodman J.L., Strelets V.B., P. Zhang, W.M. Gelbart, and the FlyBase Consortium. 2007. Flybase: genomes by the dozen. *Nucleic Acids Research*, 35:486–491.

Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, and Claire Grover. 2005. A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations:. *Comp. Funct. Genomics*, 6(1-2):77–85.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. Automatic content extraction (ace) program - task definitions and performance measures. In *Proceedings of LREC 2004*.

C. Gasperin. 2006. Semi-supervised anaphora resolution in biomedical texts. In *Proceedings of BIONLP in HLT-NAACL06, New York*, pages 96–103.

R. Hoffman and A. Valencia. 2004. A gene network for navigating the literature. *Nature Genetics*, 36:664.

I. Mani, Z. Hu, S.B. Jang, K. Samuel, M. Krause, J. Phillips, and C.H. Wu. 2005. Protein name tagging guidelines: lessons learned. *Comparative and Functional Genomics*, 6(1-2):72–76.

L.A. Ramshaw and M.P Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.

Y. Song, G. Kim, E. ad Lee, and B. Yi. 2005. Posbiotmner: a trainable biomedical named-entity recognition system. *Bioinformatics*, 21(11):2794–2796.

E.F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*.

Erik F. Tjong Kim Sang, Walter Daelemans, Hervé Déjean, Rob Koeling, Yuval Krymolowski, Vasin Punyakanok, and Dan Roth. 2000. Applying system combination to base noun phrase identification. In *COLING 2000*, pages 857–863. Saarbruecken, Germany.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part of speech tagging with a cyclic dependency network. In *HLT-NAACL*, pages 252–259.

A. Vlachos and C. Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of BIONLP in HLT-NAACL06, New York*.

K. Yamamoto, T. Kudo, T. Konagaya, and Y. Matsumoto. 2003. Protein name tagging for biomedical annotation in text. In *ACL 2003 Workshop on NLP in Biomedicine*.