Evaluating the Effects of Treebank Size in a Practical Application for Parsing

Kenji Sagae[1], **Yusuke Miyao**[1], **Rune Sætre**[1] and **Jun'ichi Tsujii**[1,2,3]
[1]Department of Computer Science, Univerisity of Tokyo, Japan
[2]School of Computer Science, University of Manchester
[3]National Center for Text Mining, Manchester, UK
{sagae,yusuke,rune.saetre,tsujii@is.s.u-tokyo.ac.jp}

Abstract

Natural language processing modules such as part-of-speech taggers, named-entity recognizers and syntactic parsers are commonly evaluated in isolation, under the assumption that artificial evaluation metrics for individual parts are predictive of practical performance of more complex language technology systems that perform practical tasks. Although this is an important issue in the design and engineering of systems that use natural language input, it is often unclear how the accuracy of an end-user application is affected by parameters that affect individual NLP modules. We explore this issue in the context of a specific task by examining the relationship between the accuracy of a syntactic parser and the overall performance of an information extraction system for biomedical text that includes the parser as one of its components. We present an empirical investigation of the relationship between factors that affect the accuracy of syntactic analysis, and how the difference in parse accuracy affects the overall system.

1 Introduction

Software systems that perform practical tasks with natural language input often include, in addition to task-specific components, a pipeline of basic natural language processing modules, such as part-of-speech taggers, named-entity recognizers, syntactic parsers and semantic-role labelers. Although such building blocks of larger language technology solutions are usually carefully evaluated in isolation using standard test sets, the impact of improvements in each individual module on the overall performance of end-to-end systems is less well understood. While the effects of the amount of training data, search beam widths and various machine learning frameworks have been explored in detail with respect to speed and accuracy in basic natural language processing tasks, how these trade-offs in individual modules affect the performance of the larger systems they compose is an issue that has received relatively little attention. This issue, however, is of great practical importance in the effective design and engineering of complex software systems that deal with natural language.

In this paper we explore some of these issues empirically in an information extraction task in the biomedical domain, the identification of protein-protein interactions (PPI) mentioned in papers abstracts from MEDLINE, a large database of biomedical papers. Due in large part to the creation of biomedical treebanks (Kulick et al., 2004; Tateisi et al., 2005) and rapid progress of data-driven parsers (Lease and Charniak, 2005; Nivre et al., 2007), there are now fast, robust and accurate syntactic parsers for text in the biomedical domain. Recent research shows that parsing accuracy of biomedical corpora is now between 80% and 90% (Clegg and Shepherd, 2007; Pyysalo et al., 2007; Sagae et al., 2008). Intuitively, syntactic relationships between words should be valuable in determining possible interactions between entities present in text. Recent PPI extraction systems have confirmed this intuition (Erkan et al., 2007; Sætre et al., 2007; Katrenko and Adriaans, 2006).

While it is now relatively clear that syntactic parsing is useful in practical tasks that use natural language corpora in bioinformatics, several ques-

Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pages 14–20,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

tions remain as to research issues that affect the design and testing of end-user applications, including how syntactic analyses should be used in a practical setting, whether further improvements in parsing technologies will result in further improvements in practical systems, whether it is important to continue the development of treebanks and parser adaptation techniques for the biomedical domain, and how much effort should be spent on comparing and benchmarking parsers for biomedical data. We attempt to shed some light on these matters by presenting experiments that show the relationship of the accuracy of a dependency parser and the accuracy of the larger PPI system that includes the parser. We investigate the effects of domain-specific treebank size (the amount of available manually annotated training data for syntactic parsers) and final system performance, and obtain results that should be informative to researchers in bioinformatics who rely on existing NLP resources to design information extraction systems, as well as to members of the parsing community who are interested in the practical impact of parsing research.

In section 2 we discuss our motivation and related efforts. Section 3 describes the system for identification of protein-protein interactions used in our experiments, and in section 4 describes the syntactic parser that provides the analyses for the PPI system, and the data used to train the parser. We describe our experiments, results and analysis in section 5, and conclude in section 6.

2 Motivation and related work

While recent work has addressed questions relating to the use of different parsers or different types of syntactic representations in the PPI extraction task (Sætre et al., 2007, Miyao et al., 2008), little concrete evidence has been provided for potential benefits of improved parsers or additional resources for training syntactic parsers. In fact, although there is increasing interest in parser evaluation in the biomedical domain in terms of precision/recall of brackets and dependency accuracy (Clegg and Shepherd, 2007; Pyysalo et al., 2007; Sagae et al., 2008), the relationship between these evaluation metrics and the performance of practical information extraction systems remains unclear. In the parsing community, relatively small accuracy gains are often reported as success stories, but again, the precise impact of such improvements on practical tasks in bioinformatics has not been established.

One aspect of this issue is the question of domain portability and domain adaptation for parsers and other NLP modules. Clegg and Shepherd (2007) mention that available statistical parsers appear to overfit to the newswire domain, because of their extensive use of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1994) during development and training. While this claim is supported by convincing evaluations that show that parsers trained on the WSJ Penn Treebank alone perform poorly on biomedical text in terms of accuracy of dependencies or bracketing of phrase structure, the benefits of using domain-specific data in terms of practical system performance have not been quantified. These expected benefits drive the development of domain-specific resources, such as the GENIA treebank (Tateisi et al., 2005), and parser domain adaption (Hara et al., 2007), which are of clear importance in parsing research, but of largely unconfirmed impact on practical systems.

Quirk and Corston-Oliver (2006) examine a similar issue, the relationship between parser accuracy and overall system accuracy in syntax-informed machine translation. Their research is similar to the work presented here, but they focused on the use of varying amounts of out-of-domain training data for the parser, measuring how a translation system for technical text performed when its syntactic parser was trained with varying amounts of Wall Street Journal text. Our work, in contrast, investigates the use of domain-specific training material in parsers for biomedical text, a domain where significant amounts of effort are allocated for development of domain-specific NLP resources in hope that such resources will result in better overall performance in practical systems.

3 A PPI extraction system based on syntactic parsing

PPI extraction is an NLP task to identify protein pairs that are mentioned as interacting in biomedical papers. Figure 2 shows two sentences that include protein names: the former sentence mentions a protein interaction, while the latter does not. Given a protein pair, PPI extraction is a task of binary classification; for example, <IL-8, CXCR1>

This study demonstrates that **IL-8** recognizes and activates **CXCR1**, **CXCR2**, and the **Duffy antigen** by distinct mechanisms.

The molar ratio of serum **retinol-binding protein** (**RBP**) to **transthyretin** (**TTR**) is not useful to assess vitamin A status during infection in hospitalized children.

Figure 2: Example sentences with protein names

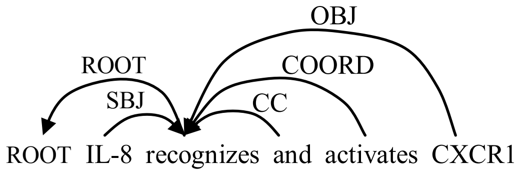

Figure 1: A dependency tree

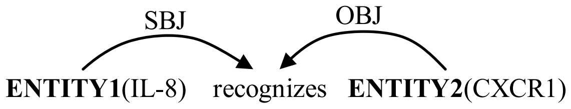

Figure 3: A dependency path between protein names

is a positive example, and <RBP, TTR> is a negative example.

Following recent work on using dependency parsing in systems that identify protein interactions in biomedical text (Erkan et al., 2007; Sætre et al., 2007; Katrenko and Adriaans, 2006), we have built a system for PPI extraction that uses dependency relations as features. As exemplified, for the protein pair **IL-8** and **CXCR1** in the first sentence of Figure 2, a dependency parser outputs a dependency tree shown in Figure 1. From this dependency tree, we can extract a dependency path between **IL-8** and **CXCR1** (Figure 3), which appears to be a strong clue in knowing that these proteins are mentioned as interacting.

The system we use in this paper is similar to the one described in Sætre et al. (2007), except that it uses syntactic dependency paths obtained with a dependency parser, but not predicate-argument paths based on deep-parsing. This method is based on SVM with SubSet Tree Kernels (Collins, 2002; Moschitti, 2006). A dependency path is encoded as a flat tree as depicted in Figure 4. Because a tree kernel measures the similarity of trees by counting common subtrees, it is expected that the system finds effective subsequences of dependency paths. In addition to syntactic dependency features, we incorporate bag-of-words features, which are regarded as a strong baseline for IE systems. We use lemmas of words before, between and after the pair of target proteins.

In this paper, we use Aimed (Bunescu and Mooney, 2004), which is a popular benchmark for the evaluation of PPI extraction systems. The Aimed corpus consists of 225 biomedical paper abstracts (1970 sentences), which are sentence-split, tokenized, and annotated with proteins and PPIs.

4 A data-driven dependency parser for biomedical text

The parser we used as component of our PPI extraction system was a shift-reduce dependency parser that uses maximum entropy models to determine the parser's actions. Our overall parsing approach uses a best-first probabilistic shift-reduce algorithm, working left-to right to find labeled dependencies one at a time. The algorithm is essentially a dependency version of the constituent parsing algorithm for probabilistic parsing with LR-like data-driven models described by Sagae and Lavie (2006). This dependency parser has been shown to have state-of-the-art accuracy in the CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre, 2007). Sagae and Tsujii (2007) present a detailed description of the parsing approach used in our work, including the parsing algorithm and the features used to classify parser actions. In summary, the parser uses an algorithm similar to the LR parsing algorithm (Knuth, 1965), keeping a stack of partially built syntactic structures, and a queue of remaining input tokens. At each step in the parsing process, the parser can apply a shift action (remove a token from the front of the queue and place it on top of the stack), or a reduce action (pop the two topmost

```
(dep_path (SBJ (ENTITY1 ecognizes))
          (rOBJ (recognizes ENTITY2)))
```

Figure 4: A tree kernel representation of the dependency path

stack items, and push a new item composed of the two popped items combined in a single structure). This parsing approach is very similar to the one used successfully by Nivre et al. (2006), but we use a maximum entropy classifier (Berger et al., 1996) to determine parser actions, which makes parsing considerably faster. In addition, our parsing approach performs a search over the space of possible parser actions, while Nivre et al.'s approach is deterministic.

The parser was trained using 8,000 sentences from the GENIA Treebank (Tateisi et al., 2005), which contains abstracts of papers taken from MEDLINE, annotated with syntactic structures. To determine the effects of training set size on the parser, and consequently on the PPI extraction system, we trained several parsing models with different amounts of GENIA Treebank data. We started with 100 sentences, and increased the training set by 100 sentence increments, up to 1,000 sentences. From that point, we increased the training set by 1,000 sentence increments. Figure 5 shows the labeled dependency accuracy for the varying sizes of training sets. The accuracy was measured on a portion of the GENIA Treebank reserved as development data. The result clearly demonstrates that the increase in the size of the training set contributes to increasing parse accuracy. Training the parser with only 100 sentences results in parse accuracy of about 72.5%. Accuracy rises sharply with additional training data until the size of the training set reaches about 1,000 sentences (about 82.5% accuracy). From there, accuracy climbs consistently, but slowly, until 85.6% accuracy is reached with 8,000 sentences of training data.

It should be noted that parser accuracy on the Aimed data used in our PPI extraction experiments may be slightly lower, since the domain of the GENIA Treebank is not exactly the same as the Aimed corpus. Both of them were extracted from MEDLINE, but the criteria for data selection were not the same in the two corpora, creating possible differences in sub-domains. We also note that the accuracy of a parser trained with more than 40,000 sentences from the Wall Street Journal portion of the Penn Treebank is under 79%, a level equivalent to that obtained by training the parser with only 500 sentences of GENIA data.

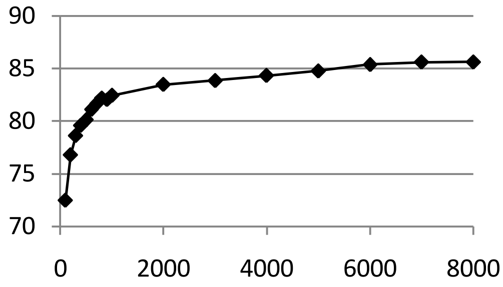

Figure 5: Data size vs. parse accuracy

5 Experiments and Results

In this section we present our PPI extraction experiments applying the dependency parsers trained with the different amounts of the GENIA Treebank in our PPI system. As we mentioned, the GENIA Treebank is used for training the parser, while the Aimed is used for training and evaluation of PPI extraction. A part-of-speech tagger trained with GENIA and PennBioIE was used. We do not apply automatic protein name detection, and instead use the gold-standard protein annotations in the Aimed corpus. Before running a parser, multiword protein names are concatenated and treated as single words. As described in Section 3, bag-of-words and syntactic dependency paths are fed as features to the PPI classifier. The accuracy of PPI extraction is measured by the abstract-wise 10-fold cross validation (Sætre et al, 2007).

When we use the part-of-speech tagger and the dependency parser trained with WSJ, the accuracy (F-score) of PPI extraction on this data set is 55.2. The accuracy increases to 56.9 when we train the part-of-speech tagger with GENIA and Penn BioIE, while using the WSJ-trained parser. This confirms the claims by Lease and Charniak (2005) that subsentential lexical analysis alone is helpful in adapting WSJ parsers to the biomedical domain. While Lease and Charniak looked only at parse accuracy, our result shows that the increase in parse accuracy is, as expected, beneficial in practice.

Figure 6 shows the relationship between the amount of parser training data and the F-score for the PPI extraction. The result shows that the accuracy of PPI extraction increases with the use of more sentences to train the parser. The best accuracy was obtained when using 4,000 sentences, where parsing accuracy is around 84.3. Although it may appear that further increasing the training data for the parser may not improve the PPI extraction accuracy (since only small and inconsistent variations in F-score are observed in Figure 6), when we plot the curves shown in Figures 5 and 6 in a single graph (Figure 7), we see that the two curves match each other to a large extent. This is supported by the strong correlation between parse accuracy and PPI accuracy observed in Figure 8. While this suggests that training the parser with a larger treebank may result in improved accuracy in PPI extraction, we observe that a 1% absolute improvement in parser accuracy corresponds roughly to a 0.25 improvement in PPI extraction F-score. Figure 5 indicates that to obtain even a 1% improvement in parser accuracy by using more training data, the size of the treebank would have to increase significantly.

Although the results presented so far seem to suggest the need for a large data annotation effort to achieve a meaningful improvement in PPI extraction accuracy, there are other ways to improve the overall accuracy of the system without an improvement in parser accuracy. One obvious alternative is to increase the size of the PPI-annotated corpus (which is distinct from the treebank used to train the parser). As mentioned in section 3, our system is trained using the Aimed corpus, which contains 225 abstracts from biomedical papers with manual annotations indicating interactions between proteins. Pairs of proteins with no interaction described in the text are used as negative examples, and pairs of proteins described as interacting are used as positive examples. The corpus contains a total of roughly 9,000 examples. Figure 9 shows how the overall system accuracy varies when different amounts of training data (varying amounts of training examples) are used to train the PPI system (keeping the parse accuracy constant, using all of the available training data in the GENIA treebank to train the parser). While Figure 5 indicates that a significant improvement in parse accuracy requires a large increase in the treebank used to train the parser, and Figure 7 shows that improvements in PPI extraction accuracy may require a sizable improvement in parse accuracy, Figure 9 suggests that even a relatively small increase in the PPI corpus may lead to a significant improvement in PPI extraction accuracy.

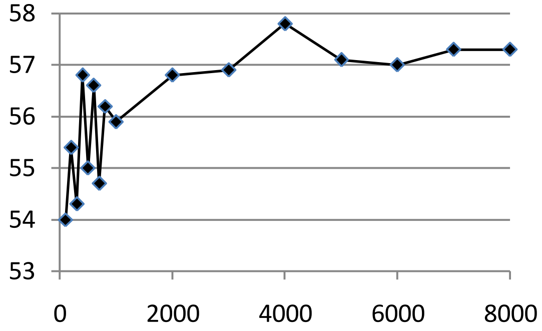

Figure 6: Parser training data size vs. PPI extraction accuracy

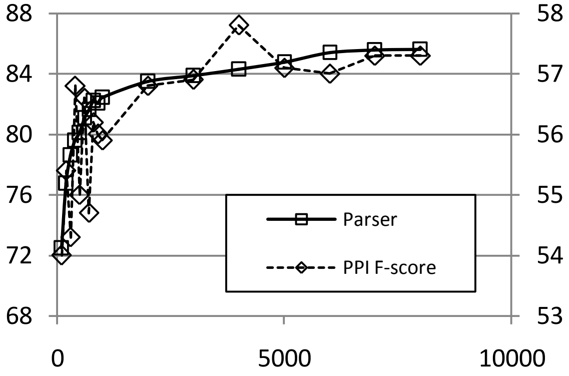

Figure 7: Parser training data size vs. parser accuracy and PPI extraction accuracy

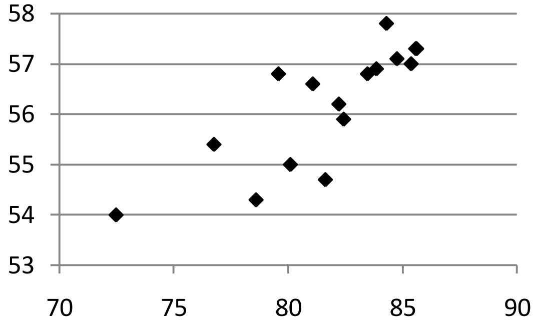

Figure 8: Parse accuracy vs. PPI extraction accuracy

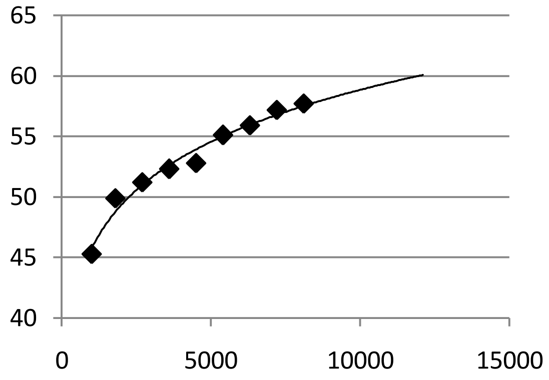

Figure 9: Number of PPI training examples vs. PPI extraction accuracy

While some of the conclusions that can be drawn from these results may be somewhat surprising, most are entirely expected. However, even in these straightforward cases, our experiments provide some empirical evidence and concrete quantitative analysis to complement intuition. We see that using domain-specific training data for the parsing component for the PPI extraction system produces superior results, compared to using training data from the WSJ Penn Treebank. When the parser trained on WSJ sentences is used, PPI extraction accuracy is about 55, compared to over 57 when sentences from biomedical papers are used. This corresponds fairly closely to the differences in parser accuracy: the accuracy of the parser trained on 500 sentences from GENIA is about the same as the accuracy of the parser trained on the entire WSJ Penn Treebank, and when these parsers are used in the PPI extraction system, they result in similar overall task accuracy. However, the results obtained when a domain-specific POS tagger is combined with a parser trained with out-of-domain data, overall PPI results are nearly at the same level as those obtained with domain-specific training data (just below 57 with a domain-specific POS tagger and out-of-domain parser, and just above 57 for domain-specific POS tagger and parser). At the same time, the argument against annotating domain-specific data for parsers in new domains is not a strong one, since higher accuracy levels (for both the parser and the overall system) can be obtained with a relatively small amount of domain-specific data.

Figures 5, 6 and 7 also suggest that additional efforts in improving parser accuracy (through the use of feature engineering, other machine learning techniques, or an increase in the size of its training set) could improve PPI extraction accuracy, but a large improvement in parser accuracy may be required. When we combine these results with the findings obtained by Miyao et al. (2008), they suggest that a better way to improve the overall system is to spend more effort in designing a specific syntactic representation that addresses the needs of the system, instead of using a generic representation designed for measuring parser accuracy. Another potentially fruitful course of action is to design more sophisticated and effective ways for information extraction systems to use NLP tools, rather than simply extracting features that correspond to small fragments of syntactic trees. Of course, making proper use of natural language analysis is a considerable challenge, but one that should be kept in mind through the design of practical systems that use NLP components.

6 Conclusion

This paper presented empirical results on the relationship between the amount of training data used to create a dependency parser, and the accuracy of a system that performs identification of protein-protein interactions using the dependency parser. We trained a dependency parser with different amounts of data from the GENIA Treebank to establish how the improvement in parse accuracy corresponds to improvement in practical task performance in this information extraction task. While parsing accuracy clearly increased with larger amounts of data, and is likely to continue increasing with additional annotation of data for the GENIA Treebank, the trend in the accuracy of PPI extraction indicates that a sizable improvement in parse accuracy may be necessary for improved detection of protein interactions.

When combined with recent findings by Miyao et al. (2008), our results indicate that further work in designing PPI extraction systems that use syntactic dependency features would benefit from more adequate syntactic representations or more sophisticated use of NLP than simple extraction of syntactic subtrees. Furthermore, to improve accuracy in this task, efforts on data annotation should focus on task-specific data (manual annotation of protein interactions in biomedical papers), rather than on additional training data for syntactic parsers. While annotation of parser training data might seems like a cost-effective choice, since improved parser results might be beneficial in a number of systems where the parser can be used, our results show that, in this particular task, efforts should be focused elsewhere, such as the annotation of addition PPI data.

Acknowledgements

We thank the anonymous reviewers for their insightful comments. This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan), Genome Network Project (MEXT, Japan), and Grant-in-Aid for Young Scientists (MEXT, Japan).

References

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

Clegg, A. and Shepherd, A. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. BMC Bioinformatics, 8:24.

Erkan, G., A. Ozgur, and D. R. Radev. 2007. Semisupervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of CoNLL-EMNLP 2007*.

Hara, T., Miyao, Y and Tsujii, J. 2007. Evaluating Impact of Re-training a Lexical Disambiguation Model on Domain Adaptation of an HPSG Parser. In *Proceedings of the International Conference on Parsing Technologies (IWPT)*.

Katrenko, S. and P. W. Adriaans. 2006. Learning relations from biomedical corpora using dependency trees. In *Proceedings of the first workshop on Knowledge Discovery and Emergent Complexity in BioInformatics (KDECB),* pages 61–80.

Kulick, S., A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein and L. Ungar. 2004. Integrated Annotation for Biomedical Information Extraction. In *Proceedings of Biolink 2004: Linking Biological Literature, Ontologies and Databases (HLT-NAACL workshop)*.

Lease, M. and Charniak, E. 2005. Parsing Biomedical Literature. In R. Dale, K.-F. Wong, J. Su, and O. Kwong, editors, *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)*, volume 3651 of Lecture Notes in Computer Science, pages 58 – 69.

Miyao, Y., Sætre, R., Sagae, K., Matsuzaki, T. and Tsujii, J. 2008. Task-Oriented Evaluation of Syntactic Parsers and Their Representations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics.*

Nivre, J., Hall, J., Kubler, S., McDonald, R., Nilsson, J., Riedel, S. and Yuret, D. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings the CoNLL 2007 Shared Task in EMNLP-CoNLL.*

Nivre, Joakim, Johan Hall, Jens Nilsson, Gulsen Eryigit,and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, shared task session.*

Pyysalo S., Ginter F., Haverinen K., Heimonen J., Salakoski T. and Laippala V. 2007. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proceedings of BioNLP 2007: Biological, Translational and Clinical Language Processing.*

Quirk, C. and Corston-Oliver S. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of EMNLP 2007*.

Sætre, R., Sagae, K., and Tsujii, J. 2007. Syntactic features for protein-protein interaction extraction. In *Proceedings of the International Symposium on Languages in Biology and Medicine (LBM short oral presentations).*

Sagae, K. and Lavie, A. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 691–698, Sydney, Australia, July. Association for Computational Linguistics.

Sagae, K., Miyao, Y. and Tsujii, J. 2008. Challenges in Mapping of Syntactic Representations for Framework-Independent Parser Evaluation. In *Proceedings of the Workshop on Automated Syntatic Annotations for Interoperable Language Resources at the First International Conference on Global Interoperability for Language Resources (ICGL'08)*.

Tateisi, Y., Yakushiji, A., Ohta, T., and Tsujii, J. 2005. Syntax annotation for the GENIA corpus. In *Proceedings Second International Joint Conference on Natural Language Processing: Companion Volume including Posters/Demos and tutorial abstracts.*