On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA

Sampo Pyysalo, Filip Ginter, Katri Haverinen,
Juho Heimonen, Tapio Salakoski
Department of Information Technology
University of Turku,
Joukahaisenkatu 3-5
20014 Turku, Finland
first.last@utu.fi

Veronika Laippala
Department of French Studies
University of Turku,
Henrikinkatu 2
20014 Turku, Finland
veronika.laippala@utu.fi

Abstract

Several incompatible syntactic annotation schemes are currently used by parsers and corpora in biomedical information extraction. The recently introduced Stanford dependency scheme has been suggested to be a suitable unifying syntactic formalism. In this paper, we present a step towards such unification by creating a conversion from the Link Grammar to the Stanford scheme. Further, we create a version of the BioInfer corpus with syntactic annotation in this scheme. We present an application-oriented evaluation of the transformation and assess the suitability of the scheme and our conversion to the unification of the syntactic annotations of BioInfer and the GENIA Treebank.

We find that a highly reliable conversion is both feasible to create and practical, increasing the applicability of both the parser and the corpus to information extraction.

1 Introduction

One of the main challenges in biomedical information extraction (IE) targeting entity relationships such as protein-protein interactions arises from the complexity and variability of the natural language statements used to express such relationships. To address this complexity, many biomedical IE systems (Alphonse et al., 2004; Rinaldi et al., 2004; Fundel et al., 2007) and annotated corpora (Kim et al., 2003; Aubin, 2005; Pyysalo et al., 2007) incorporate full syntactic analysis. However, there are significant differences between the syntactic annotation schemes employed. This leads to difficulties in sharing data between corpora and establishing the relative performance of parsers, as well as to a lack of interchangeability of one parser for another in IE systems, among other issues.

Syntactic formalisms are broadly divided into constituency and dependency. Constituency schemes are dominant in many fields and are unified under the established Penn Treebank (PTB) scheme (Bies et al., 1995). However, dependency schemes have been suggested to be preferable in IE, as they represent the semantic structure of the sentences more directly (see, e.g., de Marneffe et al. (2006)). Further, Lin (1998) argues for dependency-based evaluation of both dependency and constituency parsers, since it allows evaluation metrics that are more relevant to semantic interpretation as well as intuitively more meaningful. Even though there is clearly a need for a unifying scheme for dependency comparable to that of PTB for constituency, no widely adopted standard currently exists.

In this paper, we present a step towards unifying the diverse syntactic schemes in use in IE systems and corpora such as the GENIA Treebank[1] and the recently introduced BioInfer corpus (Pyysalo et al., 2007). Clegg and Shepherd (2007) have recently proposed to use the Stanford dependency scheme (de Marneffe et al., 2006) as a common, application-oriented syntactic representation. To assess this choice, we develop a set of conversion rules for transforming the Link Grammar (LG) de-

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA

pendency scheme (Sleator and Temperley, 1993) to the Stanford scheme, and then create a version of the BioInfer corpus in the Stanford scheme by applying the conversion rules and manually correcting the errors. By making the BioInfer corpus available in the Stanford scheme, we also increase the value of the corpus for biomedical IE. The transformation has the further benefit of allowing Link Grammar output to be normalized into a more application-oriented form. Finally, to assess the practical value of the conversion method and of the BioInfer syntactic annotation in the Stanford scheme, we compare the Charniak-Lease constituency parser[2] (Charniak and Lease, 2005) and BioLG,[3] an adaptation of LG (Pyysalo et al., 2006), on the unified dataset combining the constituency-annotated GENIA Treebank with the dependency-annotated BioInfer corpus.

The transformation rules and software, as well as the Stanford annotation of the BioInfer corpus, the main practical results of this work, are freely available at `http://www.it.utu.fi/BioInfer`.

2 Motivation

To support the development of IE systems, it is important for a corpus to provide three key types of annotation capturing the named entities, their relationships and the syntax. To our knowledge, there are only two corpora in the biomedical domain that currently provide these three annotation types simultaneously—BioInfer and LLL (Aubin, 2005). In addition, GENIA, the *de facto* standard domain corpus for named entity recognition and syntactic analysis, is in the process of adding a relationship annotation. The corpora have different strengths; BioInfer provides detailed relationship annotation, while GENIA has broader coverage of named entities and a larger treebank. Unifying the syntactic annotations of these two corpora allows these strengths to be combined.

The BioInfer syntactic annotation follows the LG dependency scheme, addressing the recent interest in LG in the biomedical NLP community (Ding et al., 2003; Alphonse et al., 2004; Aubin et al., 2005). However, the LG scheme has been criticized for being oriented more towards structural than semantic relations and having excessively detailed link types whose functional meaning and value for semantic analysis is questionable (Schneider, 1998; de Marneffe et al., 2006). Our experience with LG leads us to largely agree with these criticisms.

De Marneffe et al. (2006) have recently introduced a transformation from PTB to the Stanford scheme. Clegg and Shepherd (2007) have applied this transformation to perform a dependency-based comparison of several statistical constituency parsers on the GENIA Treebank and have argued for the adoption of the Stanford scheme in biomedical IE. Moreover, the IE system of Fundel et al. (2007), which employs the Stanford scheme, was shown to notably outperform previously applied systems on the LLL challenge dataset, achieving an F-score of 72% against a previous best of 54%. This further demonstrates the suitability of the Stanford scheme to IE applications.

3 Dependency schemes

In this section, we present the Stanford and LG dependency schemes and discuss their relative strengths.

3.1 Stanford dependency scheme

A parse in the Stanford scheme (SF) is a directed graph where the nodes correspond to the words and the edges correspond to pairwise syntactic dependencies between the words. The scheme defines a hierarchy of 48 grammatical relations, or dependency types. The most generic relation, *dependent*, can be specialized as *auxiliary*, *argument*, or *modifier*, which again have several subtypes (de Marneffe et al., 2006).

The Stanford conversion transforms phrase structure parses into the Stanford scheme. First, the semantic head of each constituent is identified using head rules similar to those of Collins (1999) and untyped dependencies are then extracted and labeled with the most specific grammatical relations possible using Tregex rules (Levy and Andrew, 2006).

The system additionally provides a set of *collapsing rules*, suggested to be beneficial for IE applications (de Marneffe et al., 2006; Clegg and Shepherd, 2007). These rules collapse some dependencies by incorporating certain parts of speech (mostly

[2] `http://nlp.stanford.edu/software/`, version 1.5.1

[3] `http://www.it.utu.fi/BioLG`, version 1.2.4

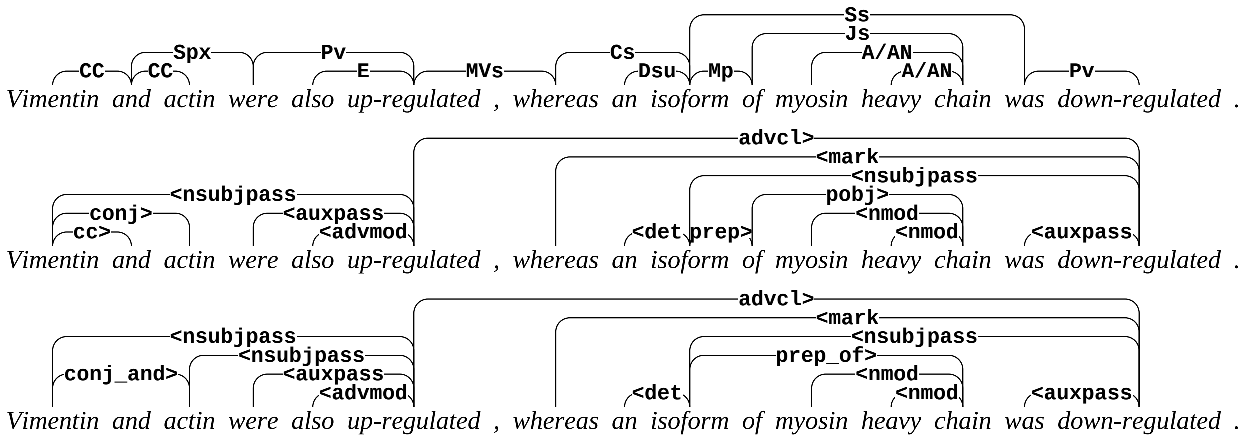
Ss
Js
CC
Spx
CC
Pv
E
MVs
Cs
Dsu
Mp
A/AN
A/AN
Pv
Vimentin and actin were also up-regulated , whereas an isoform of myosin heavy chain was down-regulated .
advcl>
<mark
<nsubjpass
<nsubjpass
pobj>
conj>
<auxpass
<nmod
<cc>
<advmod
<det
prep>
<nmod
<auxpass
Vimentin and actin were also up-regulated , whereas an isoform of myosin heavy chain was down-regulated .
advcl>
<mark
<nsubjpass
<nsubjpass
<nsubjpass
prep_of>
conj_and>
<auxpass
<nmod
<advmod
<det
<nmod
<auxpass
Vimentin and actin were also up-regulated , whereas an isoform of myosin heavy chain was down-regulated .

Figure 1: A sentence from the BioInfer corpus with its LG linkage (top), the Stanford parse (middle), and the collapsed Stanford parse (bottom). The < and > symbols denote the direction of dependencies.

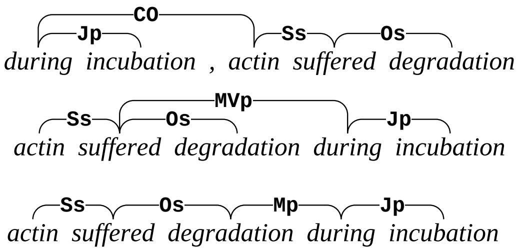
CO
Jp
Ss
Os
during incubation , actin suffered degradation
MVp
Ss
Os
Jp
actin suffered degradation during incubation
Ss
Os
Mp
Jp
actin suffered degradation during incubation

Figure 2: Variation in the link type connecting a preposition: *CO* to the main noun in topicalized prepositional phrases, *MVp* when modifying a verb, and *Mp* when modifying a noun.

conjunctions and prepositions) in grammatical relations. This is realized by combining two relations and denominating the resulting dependency with a type based on the word to which the original two relations were linked (see Figure 1).

In the LG-SF conversion, we target the uncollapsed Stanford scheme, as the collapsing rules have already been developed and reported by de Marneffe et al.; reimplementing the collapsing would be an unnecessary duplication of efforts. Also, the collapsed relations can easily be created based on the uncollapsed ones, whereas reversing the conversion would be more complicated.

3.2 LG dependency scheme

Link Grammar (Sleator and Temperley, 1993) is closely related to dependency formalisms. It is based on the notion of typed *links* connecting words. While links are not explicitly directional, the roles of the words can be inferred from their left-to-right order and the link type. An LG parse, termed *linkage*, consists of a set of links that connect the words so that no two links cross or connect the same two words. When discussing LG, we will use the terms dependency and link interchangeably.

Compared to the 48 dependency types of the Stanford scheme, the LG English grammar defines over 100 main link types which are further divided into 400 subtypes. The unusually high number of distinct types is one of the properties of the LG English grammar that complicate the application of LG in information extraction. Consider, for instance, the case of prepositional phrase attachment illustrated in Figure 2, where all the alternative attachment structures receive different types. Arguably, this distinction is unimportant to current IE systems and therefore should be normalized. This normalization is inherent in the Stanford scheme, where the preposition always attaches using a *prep* dependency.

In contrast to such unnecessarily detailed distinctions, in certain cases LG types fail to make semantically important distinctions. For instance, the *CO* link type is used to mark almost all clause openers, not distinguishing between, for example, adverbial and prepositional openers.

4 Our contributions

In this section, we describe the LG-SF conversion as well as SF BioInfer, the BioInfer corpus syntactic annotation in the Stanford scheme. These are the two primary contributions of this study.

4.1 LG-SF conversion

The LG-SF conversion transforms the undirected LG links into directed dependencies that follow the Stanford scheme. The transformation is based on handwritten rules, each rule consisting of a pattern that is matched in the LG linkage and generating a single dependency in the Stanford parse. Since the conversion rules only refer to the LG linkage, they do not influence each other and are applied independently in an arbitrary order. The pattern of each rule is expressed as a set of positive or negative constraints on the presence of LG links. The constraints typically restrict the link types and may also refer to the lexical level, restricting only to links connecting certain word forms. Since LG does not define link directionality, the patterns refer to the left-to-right order of tokens and the rules must explicitly specify the directionality of the generated SF dependencies.

As an example, let us consider the rule $[\mathrm{X} \overset{\mathrm{Pv}}{—} \mathrm{Y}] \Rightarrow \mathrm{Y} \overset{\mathrm{auxpass}}{\rightarrow} \mathrm{X}$. The pattern matches two tokens connected with an LG link of type *Pv* and generates the corresponding directed *auxpass* dependency. This rule applies twice in the linkage in Figure 1. It is an example of a rare case of a one-to-one correspondence between an LG and an SF type. Many-to-many correspondences are much more common: in these cases, rules specify multiple restrictions and multiple rules are needed to generate all instances of a particular dependency type. As a further example, we present the three rules below, which together generate all left-to-right *prep* dependencies. An exclamation mark in front of a restriction denotes a negative restriction, i.e., the link must not exist in order for the rule to apply. The link types are specified as regular expressions.

$$[\mathrm{A} \overset{\mathrm{Mp|MX[a\text{-}z]x}}{—} \mathrm{B}]![\mathrm{B} \overset{\mathrm{Cs}}{—} \mathrm{C}]![\mathrm{A} \overset{\mathrm{RS}}{—} \mathrm{D}] \Rightarrow \mathrm{A} \overset{\mathrm{prep}}{\rightarrow} \mathrm{B}$$

$$[\mathrm{A} \overset{\mathrm{OF|MVx}}{—} \mathrm{B}]![\mathrm{A} \overset{\mathrm{RS}}{—} \mathrm{C}] \Rightarrow \mathrm{A} \overset{\mathrm{prep}}{\rightarrow} \mathrm{B}$$

$$[\mathrm{A} \overset{\mathrm{MVp}}{—} \mathrm{B}]![\mathrm{A} \overset{\mathrm{RS}}{—} \mathrm{C}]![\mathrm{C} \overset{\mathrm{MVl}}{—} \mathrm{A}] \Rightarrow \mathrm{A} \overset{\mathrm{prep}}{\rightarrow} \mathrm{B}$$

The first of the above three rules generates the *prep* dependency in the parse in Figure 1, with *A*=*isoform* and *B*=*of*. The variables *C* and *D* are not bound to any tokens in this sentence, as they only occur in negative restrictions.

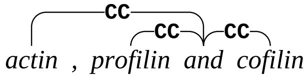

Figure 3: Example of a structure where the relative order of the first two tokens cannot be resolved by the rules.

To resolve coordination structures, it is crucial to recognize the leftmost coordinated element, i.e. the head of the coordination structure in the SF scheme. However, the conversion rule patterns are unable to capture general constraints on the relative order of the tokens. For instance, in the linkage in Figure 3, it is not possible to devise a pattern only matching one of the tokens *actin* and *profilin*, while not matching the other. Therefore, we perform a pre-processing step to resolve the coordination structures prior to the application of the conversion rules. After the pre-processing, the conversion is performed with the lp2lp software (Alphonse et al., 2004), previously used to transform LG into the LLL competition format (Aubin, 2005).

In the development of the LG-SF conversion and SF BioInfer, we make the following minor modifications to the Stanford scheme. The scheme distinguishes nominal and adjectival pre-modifiers of nouns, a distinction that is not preserved in the BioInfer corpus. Therefore, we merge the nominal and adjectival pre-modifier grammatical relations into a single relation, *nmod*. For the same reason, we do not distinguish between apposition and abbreviation, and only use the *appos* dependency type. Finally, we do not annotate punctuation.

Schneider (1998) has previously proposed a strategy for identifying the head word for each LG link, imposing directionality and thus obtaining a dependency graph. Given the idiosyncrasies of the LG linkage structures, this type of transformation into dependency would clearly not have many of the normalizing benefits of the LG-SF transformation.

4.2 SF BioInfer

For creating the BioInfer corpus syntactic annotation in the Stanford scheme, the starting point of the annotation process was the existing manual annotation of the corpus in the LG scheme to which we applied the LG-SF conversion described in Section 4.1. The resulting SF parses were then manually corrected by four annotators. In the manual correction phase, each sentence was double-annotated, that is, two annotators corrected the converted output independently. All disagreements were resolved jointly by all annotators.

To estimate the annotation quality and the stability of the SF scheme, we determined annotator agreement as precision and recall measured against the final annotation. The average annotation precision and recall were 97.5% and 97.4%, respectively. This high agreement rate suggests that the task was well-defined and the annotation scheme is stable.

The BioInfer corpus consists of 1100 sentences and, on average, the annotation consumed approximately 10 minutes per sentence in total.

5 Evaluation

In this section, we first evaluate the LG-SF conversion. We then present an evaluation of the Charniak-Lease constituency parser and the BioLG dependency parser on BioInfer and GENIA.

5.1 Evaluation of the conversion rules

In the evaluation of the conversion rules against the gold standard SF BioInfer annotation, we find a precision of 98.0% and a recall of 96.2%. Currently, the LG-SF conversion consists of 114 rules, each of which specifies, on average, 4.4 restrictions. Altogether the rules currently generate 32 SF dependency types, thus averaging 3.5 rules per SF type. Only 9 of the SF types are generated by a single rule, while the remaining require several rules. We estimate that the current ruleset required about 100 hours to develop.

In Figure 4, we show the cumulative precision and recall of the rules when added in the descending order of their recall. Remarkably, we find that a recall of 80% is reached with just 13 conversion rules, 90% with 28 rules, and 95% with 56 rules. These figures demonstrate that while the SF and LG schemes are substantially different, a high-recall conversion can be obtained with approximately fifty carefully crafted rules. Additionally, while precision is consistently high, the highest-recall rules also have the highest precision. This may be related to the fact that the most common SF dependency types have a straightforward correspondence in LG types.

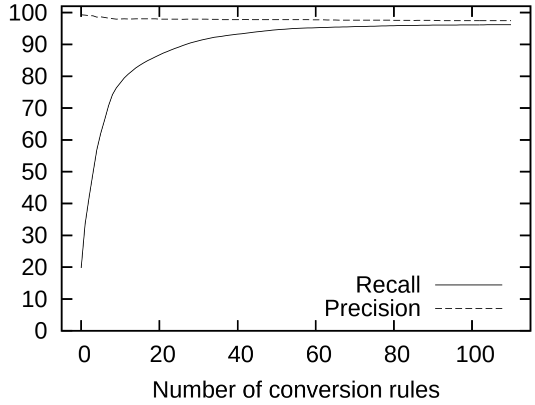

Figure 4: Cumulative precision and recall of the conversion rules.

A common source of errors in the LG-SF conversion are the Link Grammar *idiomatic expressions*, which are analyzed as a chain of *ID* links (0.7% of all links in the BioInfer corpus) and connected to the linkage always through their last word. Some examples of LG idiomatic expressions include *each other, no one*, *come of age*, *gotten rid of*, *for good*, and *the like*. These expressions are often problematic in the SF conversion as well. We did not attempt any wide-coverage systematic resolution of the idiomatic expressions and, apart from the most common cases such as *in vitro*, we preserve the LG structure of connecting these expressions through their last word. We note, however, that the list of idiomatic LG expressions is closed, and therefore a case-by-case resolution leading to full coverage is possible, although not necessarily practical.

Similar to the LG idiomatic expressions are the SF *dep* dependencies, generated when none of the SF rules assigns a more specific type. In most cases, *dep* is a result of a lack of coverage of the SF conversion rules, typically occurring in rare or idiomatic expressions. We assume that many of the *dep* dependencies will be resolved in the future, given that the SF conversion and the SF dependency scheme itself are presented by the authors as a work in progress. Therefore, we do not attempt to replicate most of the SF *dep* dependencies with the LG-SF conversion rules; much of the effort would be obsoleted by the progress of the SF conversion. The *dep* dependencies account for 23% of the total 3.8% of dependencies not recovered by the LG-SF conversion.

	Charniak-Lease			BioLG		
corpus	Prec.	Rec.	F	Prec.	Rec.	F
GENIA	81.2	81.3	81.3	76.9	72.4	74.6
BioInfer	78.4	79.9	79.4	79.6	76.1	77.8

Table 1: Parser performance. Precision, recall and F-measure for the two parsers on the two corpora.

	BioLG		
scheme	Prec.	Rec.	F
LG	78.2	77.2	77.7
SF	79.6	76.1	77.8

Table 2: BioLG performance on the BioInfer corpus with and without the LG-SF conversion.

5.2 Evaluated parsers and corpora

The Charniak-Lease parser is a statistical constituency parser developed by Charniak and Lease (2005). It is an adaptation of the Charniak parser (Charniak, 1999) to the biomedical domain. For example, it uses a POS-tagger trained on the GENIA corpus, although the parser itself has been trained on the Penn Treebank. The Charniak-Lease parser is of particular interest, because in a recent comparison performed by Clegg and Shepherd (2007) on the GENIA Treebank, it was the best-performing of several state-of-the-art statistical constituency parsers.

The LG parser is a rule-based dependency parser with a broad-coverage grammar of newspaper-type English. It has no probabilistic component and does not perform pruning of ambiguous alternatives during parsing. Instead, the parser generates all parses accepted by the grammar. Simple heuristics are applied to rank the alternative parses.

Here, we evaluate a recently introduced adaptation of LG to the biomedical domain, BioLG (Pyysalo et al., 2006), incorporating the GENIA POS tagger (Tsuruoka et al., 2005) as well as a number of modifications to lexical processing and the grammar.

To facilitate the comparison of results with those of Clegg and Shepherd, we use their modified subset of GENIA Treebank.[4] As 600 of the 1100 BioInfer sentences have previously been used in the development of the BioLG parser, we only use the remaining 500 blind sentences of BioInfer in the evaluation.

5.3 Parser performance

To evaluate the performance of the parsers, we determined the *precision*, *recall* and *F-measure* by comparing the parser output against the corpus gold standard dependencies. The matching criterion required the correct words to be connected and the direction and type of the dependency to be correct. The dependency-based evaluation results for the Charniak-Lease and BioLG parsers on the GENIA and BioInfer corpora are shown in Table 1. We note that Clegg and Shepherd (2007) report 77% F-score performance of Charniak-Lease on the GENIA corpus, using the collapsed variant of the SF scheme. We replicated their experiment using the uncollapsed variant and found an F-score of 80%. Therefore, most of the approximately 4% difference compared to our finding reported in Table 1 is due to this difference in the use of collapsing, with our modifications to the SF scheme having a lesser effect. The decrease in measured performance caused by the collapsing is, however, mostly an artifact caused by merging several dependencies into one; a single mistake of the parser can have a larger effect on the performance measurement.

We find that while the performance of the Charniak-Lease parser is approximately 2 percentage units better on GENIA than on BioInfer, for BioLG we find the opposite effect, with performance approximately 3 percentage units better on BioInfer. Thus, both parsers perform better on the corpora closer to their native scheme. We estimate that this total 5 percentage unit divergence represents an upper limit to the evaluation bias introduced by the two sets of conversion rules. We discuss the possible causes for this divergence in Section 5.4.

To determine whether the differences between the two parsers on the two corpora were statistically significant, we used the Wilcoxon signed-ranks test for F-score performance using the Bonferroni correction for multiple comparisons ($N = 2$), following the recent recommendation of Demšar (2006). We find that the Charniak-Lease parser outperforms BioLG statistically significantly on both the GENIA corpus ($p \ll 0.01$) and on the BioInfer corpus

[4] http://chomsky-ext.cryst.bbk.ac.uk/andrew/downloads.html

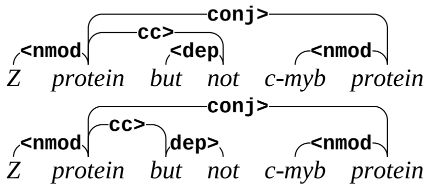
conj>
cc>
<nmod
<dep
<nmod
Z protein but not c-myb protein
conj>
cc>
<nmod
dep>
<nmod
Z protein but not c-myb protein

Figure 5: Example of divergence on the interpretation of the Stanford scheme. Above: GENIA and Stanford conversion interpretation. Below: BioInfer and LG-SF rules interpretation.

($p < 0.01$). Thus, the relative performance of the parsers can, in this case, be established even in the presence of opposing conversion biases on the two corpora.

In Table 2, we present an evaluation of the BioLG parser with and without the LG-SF conversion, specifically evaluating the effect of the conversion presented in this study. Here we find a substantially more stable performance, including even an increase in precision. This further validates the quality of the conversion rules.

Finally, we note that the processing time required to perform the conversions is insignificant compared to the time consumed by the parsers.

5.4 Discussion

Evaluating BioLG on GENIA and the Charniak-Lease parser on BioInfer includes multiple sources of divergence. In addition to parser errors, differences can be created by the LG-SF conversion and the Stanford conversion. Moreover, in examining the outputs we identified that a further source of divergence is due to differing interpretations of the Stanford scheme. One such difference is illustrated in Figure 5. Here the BioLG parser with the LG-SF conversion produces an analysis that differs from the result of converting the GENIA Treebank analysis by the Stanford conversion. This is due to the Stanford conversion producing an apparently flawed analysis that is not replicated by the LG-SF conversion. In certain cases of this type, the lack of a detailed definition of the SF scheme prevents distinguishing between conversion errors and intentional analyses. This will necessarily lead to differing interpretations, complicating precise evaluation.

6 Conclusions

We have presented a step towards unifying syntactic annotations under the Stanford dependency scheme and assessed the feasibility of this unification by developing and evaluating a conversion from Link Grammar to the Stanford scheme. We find that a highly reliable transformation can be created, giving a precision and recall of 98.0% and 96.2%, respectively, when compared against our manually annotated gold standard version of the BioInfer corpus. We also find that the performance of the BioLG parser is not adversely affected by the conversion. Given the clear benefits that the Stanford scheme has for domain analysis, the conversion increases the overall suitability of the parser to IE applications. Based on these results, we conclude that converting to the Stanford scheme is both feasible and practical.

Further, we have developed a version of the BioInfer corpus annotated with the Stanford scheme, thereby increasing the usability of the corpus. We applied the LG-SF conversion to the original LG BioInfer annotation and manually corrected the errors. The high annotator agreement of above 97% precision and recall confirms the stability of the SF scheme.

We have also demonstrated that the unification permits direct parser comparison that was previously impossible. However, we found that there is a certain accumulation of errors caused by the conversion, particularly in a case when two distinct rule sets are applied. In our case, we estimate this error to be on the order of several percentage units. Nevertheless, we were able to establish the relative performance of the parsers with strong statistical significance. These results demonstrate the utility of the Stanford scheme as a unifying syntactic representation. We note that an authoritative definition of the Stanford scheme would further increase its value.

Acknowledgments

We would like to thank Erick Alphonse, Sophie Aubin and Adeline Nazarenko for providing us with the lp2lp software and the LLL conversion rules. We would also like to thank Andrew Brian Clegg and Adrian Shepherd for making available the data and evaluation tools used in their parser evaluation. This research was supported by the Academy of Finland.

References

Erick Alphonse, Sophie Aubin, Philippe Bessières, Gilles Bisson, Thierry Hamon, Sandrine Laguarigue, Adeline Nazarenko, Alain-Pierre Manine, Claire Nédellec, Mohamed Ould Abdel Vetah, Thierry Poibeau, and Davy Weissenbacher. 2004. Event-based information extraction for the biomedical domain: the Caderige project. In N. Collier, P. Ruch, and A. Nazarenko, editors, *COLING NLPBA/BioNLP Workshop*, pages 43–49, Geneva, Switzerland.

Sophie Aubin, Adeline Nazarenko, and Claire Nédellec. 2005. Adapting a general parser to a sublanguage. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria*, pages 89–93. Incoma, Bulgaria.

Sophie Aubin. 2005. LLL challenge - syntactic analysis guidelines. Technical report, LIPN, Université Paris Nord, Villetaneuse.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for treebank II style. Technical report, Penn Treebank Project, University of Pennsylvania.

Eugene Charniak and Matthew Lease. 2005. Parsing biomedical literature. In R. Dale, K. F. Wong, J. Su, and O. Y. Kwong, editors, *Proceedings of the Second International Joint Conference on Natural Langage Processing (IJCNLP), Jeju Island, Korea*, pages 58–69.

Eugene Charniak. 1999. A maximum-entropy-inspired parser. Technical report, Brown University.

Andrew Brian Clegg and Adrian Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, and D. Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy*, pages 449–454.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Jing Ding, Daniel Berleant, Jun Xu, and Andy W. Fulmer. 2003. Extracting biochemical interactions from MEDLINE using a link grammar parser. In B. Werner, editor, *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Sacramento, CA*, pages 467–471. IEEE Computer Society, Los Alamitos, CA.

Katrin Fundel, Robert Kuffner, and Ralf Zimmer. 2007. RelEx–Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–182.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, and D. Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy*, pages 2231–2234.

Dekang Lin. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4(2):97–114.

Sampo Pyysalo, Tapio Salakoski, Sophie Aubin, and Adeline Nazarenko. 2006. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3).

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Andreas Persidis, and Ourania Konstanti. 2004. Mining relations in the GENIA corpus. In *Proceedings of the Workshop on Data Mining and Text Mining for Bioinformatics (ECML/PKDD), Pisa, Italy*, pages 61–68.

Gerold Schneider. 1998. A linguistic comparison of constituency, dependency and link grammar. Master's thesis, University of Zürich.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*, pages 277–291.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In P. Bozanis and E. N. Houstis, editors, *10th Panhellenic Conference on Informatics (PCI)*, pages 382–392.