Parsing Clinical Finnish: Experiments with Rule-Based and Statistical Dependency Parsers

Katri Haverinen,[1] **Filip Ginter,**[1] **Veronika Laippala,**[1,2] **and Tapio Salakoski**[1]
[1]Department of Information Technology
[2]Department of French Studies
University of Turku
Joukahaisenkatu 3-5
20520 Turku, Finland
first.last@utu.fi

Abstract

In this paper, we present a new syntactically annotated corpus consisting of daily notes from an intensive care unit in a Finnish hospital. Using the corpus, we perform experiments with both rule-based and statistical parsers. We apply an existing rule-based parser specifically developed for this clinical language and create a set of conversion rules for transforming the constituency scheme of this parser into the dependency scheme of the corpus. The statistical parser is induced from the corpus using the MaltParser system.

We find that even with a modestly-sized corpus, the statistical parser achieves results comparable to those previously reported on a number of languages using considerably larger corpora. The accurate constituency-to-dependency conversion improves the applicability of the rule-based parser by inferring grammatical roles, thus deepening its analyses.

1 Introduction

The potential advantages of applying natural language processing methods in the clinical domain are numerous, with many useful applications in decision support, patient management and profiling, and mining trends (see, e.g., the recent review by Friedman and Johnson (2006)). While certain applications, such as document retrieval and trend mining, can solely rely on word frequency-based statistical methods, a number of applications build on a detailed analysis of the text, typically involving syntactic parsing.

In this paper, we describe experiments on full parsing of Finnish intensive care unit (ICU) nursing documents written in a specific language referred to as ICU Finnish throughout the paper. The main contributions of this work are a corpus of ICU Finnish, syntactically annotated in an adapted version of the Stanford dependency (SD) scheme, and both rule-based and statistical parsing experiments on this corpus. We apply the rule-based parser of Laippala et al. (2009) developed for ICU Finnish, and develop a conversion from its native constituency scheme to the SD scheme. We also conduct experiments with a statistical parser induced from the ICU Finnish corpus using the MaltParser (Nivre et al., 2007) system. This allows us to evaluate and contrast the relative advantages of the two parsing approaches in this domain.

2 Related work

There are numerous applications of full syntactic parsers in the clinical domain. For instance, the Stanford parser has been applied to the extraction of noun phrases with full phrase structures and to negation detection in clinical radiology reports (Huang and Lowe, 2007; Huang et al., 2005). There have also been many studies on the adaptation of existing parsers to the specific domain of biomedical language. For example, Szolovits (2003) describes a method for expanding the Link Grammar (LG) lexicon with UMLS Specialist lexicon terms to improve its applicability to medical texts and Pyysalo et al. (2006) incorporate into LG a domain-adapted part-of-speech tagger.

The different ways to represent natural language syntax can be broadly distinguished into two categories. A constituency analysis divides the sentence into nested phrases, whereas a dependency analysis consists of a set of labelled dependencies between pairs of words. In this work, we focus on dependency parsing because of its benefits in applications and parser evaluation (see for example Lin (1998), Clegg and Shepherd (2007), and Nivre (2008b)), as well as its applicability to languages with a relatively free word order, such

Kristiina Jokinen and Eckhard Bick (Eds.)
NODALIDA 2009 Conference Proceedings, pp. 65–72

Yövuoro	Nightshift
Potilas levoton, valittaa kipua.	Patient restless, complains of pain.
Annettu 100mg [lääke] hieman rauhottui.	Given 100mg [drugname] a little calmed down.
HENGITS: Hapettuu hyvin repiraattorissa.	BREATING: Oxidates well in repirator.
Putkesta hiukan nest. illalla.	A little liq. from the drain in the evening.
Diureesi: riittävää.	Diuresis: sufficient.
Hemodyn: annettu 50 mg/h [lääke],	Hemodyn: given 50 mg/h [drugname],
heikohko vaste vaihdettu [lääke].	rather weak response changed to [drugname].
OMAISET: vaimo soittanut jutellut lääkärin kanssa.	RELATIVES: wife called talked to doctor.

Figure 1: Example of ICU Finnish (left column) and its exact translation (right column), including spelling errors, capitalization, and the like.

as Finnish. We apply the Stanford dependency scheme (de Marneffe et al., 2006; de Marneffe and Manning, 2008), which has recently been employed in several studies especially in the biomedical domain, but also in other contexts. For an extensive list of applications, we refer to the review by de Marneffe and Manning (2008).

While numerous corpora and parsers exist for English and many other languages, resources for Finnish are scarce. For instance, there is no publicly available syntactically annotated corpus suitable for statistical parser induction. The only publicly available full parser is Connexor Machinese Syntax,[1] a closed-source commercial dependency parser for the general language. Other tools include FinTWOL and FinCG,[2] a morphological analyzer and a Constraint Grammar parser that resolves morphological ambiguity (Koskenniemi, 1983; Karlsson, 1990). The rule-based parser of Laippala et al. (2009) used in this work was developed for the clinical domain, and builds full constituency analyses on top of the morpholexical analyses provided by FinTWOL and FinCG.

3 ICU Finnish in the Stanford dependency scheme

ICU Finnish differs from standard Finnish in many ways (for details, see the discussion by Laippala et al. (2009)). Some of the most distinguishing features present in ICU Finnish, as well as many clinical sublanguages, are frequent misspellings, abbreviations and technical terms, telegraphic sentences, syntactic structures that would not be allowed in standard language, and frequent omissions of main verbs, subjects and copulas. Figure 1 is an illustration of ICU Finnish.[3] The effects of ICU Finnish features on analyzing the syntax will be more thoroughly discussed in Section 3.2.

[1] http://www.connexor.eu

[2] http://www.lingsoft.fi

[3] Due to the confidential nature of the patient data, these, as well as all examples used in this paper, are not actual sentences from the data, but rather illustrative examples.

3.1 The SD scheme

In the SD scheme, the syntactic structure of a sentence is represented as a directed graph where the nodes correspond to words and the edges correspond to dependencies. Unlike in most dependency schemes, SD graphs are not necessarily trees and may even contain directed cycles. Each dependency is labelled with a dependency type that represents the syntactic function of the dependent word. In the latest version of the SD scheme (de Marneffe and Manning, 2008), there are in total 55 dependency types.

We have chosen the SD scheme due to its numerous successful applications in different contexts. Further, de Marneffe and Manning find the scheme applicable in parser comparison. This particular aspect of the scheme is of importance with respect to this work, as one part of this study is a comparison of two parsers. Alternative schemes, such as Grammatical Relations (Carroll et al., 1998) and the Connexor Machinese Syntax scheme, were also considered. The former has been suggested by its authors to be suitable for multiple languages, and the latter is a scheme designed for standard Finnish.

3.2 Applying the SD scheme to ICU Finnish

The SD scheme was designed for standard English. In this section, we describe the modifications made in order to adapt it to ICU Finnish. These modifications include both those that are required by Finnish in general, and those implied by the nature of the ICU sublanguage. For an illustration of the modified SD scheme, see Figure 2. As a detailed description of the SD scheme is beyond the scope of this paper, we only discuss our modifications to it and refer to the description by de Marneffe and Manning (2008).

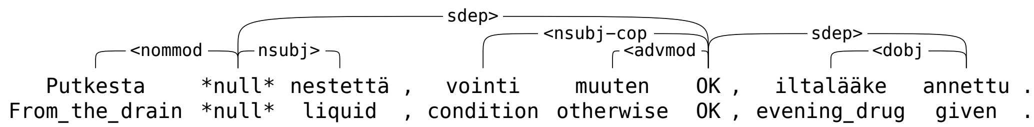

Figure 2: The modified SD scheme. Note the following features: nominal modifiers (*nommod* dependencies), dependencies between sentences (*sdep*), null verbs that represent omitted main verbs, explicit marking of copula subjects (*nsubj-cop*), and the use of direct object (*dobj*) in passive sentences. The sentence can be roughly translated as *Liquid from the drain, condition otherwise OK, evening drug given.*

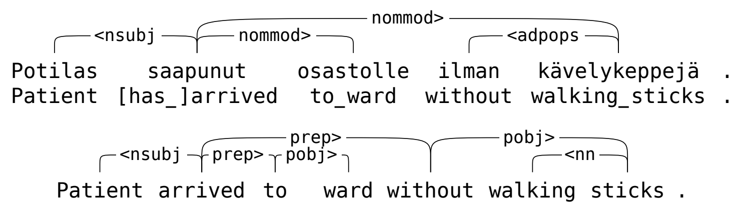

Figure 3: Top: usage of the new dependency types *nommod* and *adpos*. Bottom: the corresponding English sentence and annotation in the SD scheme. Note that the type *nommod* is used both for nominal inflection and prepositional phrases.

3.2.1 Prepositional phrases

In the Finnish language, prepositions are relatively rare. Most English clauses with prepositional phrases have Finnish equivalents that use nominal inflection. For an example of a typical case, see Figure 2.

Seeing that inflectional and prepositional structures are semantically similar, it would be desirable to represent them in a similar manner also in the dependency structure. Therefore, we introduce a new dependency type, *nommod* (*nominal modifier*), to represent inflectional structures. This same type can also be used in sentences with actual pre- and postpositions. Only one additional type is needed for prepositional structures, a type named *adpos* (*adposition*). For an illustration of the usage of these two types, see Figure 3. The structure given to prepositional phrases is similar to that used in the scheme of the Pro3Gres parser (Schneider et al., 2004).

3.2.2 Passive subjects

Certain Finnish clause types, contrary to their English counterparts, do not require a subject. One that has a particular effect on our work is the passive voice. The surface subject in English passive clauses corresponds to both surface and deep object in Finnish. Therefore, we have not used the dependency type *nsubjpass* at all, and have used *dobj* instead.

3.2.3 Dependencies between sentences

A third modification to the SD scheme is required by the nature of the ICU language: sentence boundaries are often not clearly marked, or they lack punctuation altogether (see Figure 4). We split the text into separate sentences only when there is explicit punctuation that marks the sentence boundary. Recovering sentence boundaries that have no explicit surface marking is left to the parser, as recognizing them would be difficult for standard sentence splitters that lack syntax information. We have thus introduced a new dependency type, *sdep*, to connect these isolated sentences that are not explicitly coordinated or subordinated. To produce an analysis that is æsthetic from a scheme design point of view, if several *sdep* dependencies are needed in the same surface sentence, they are chained. This is to avoid unnecessarily long dependencies that are difficult for parsers to recover.

3.2.4 Omissions

In ICU Finnish, a frequent syntactic feature that has a notable effect on parsing the language is the omission of different sentence elements. One example of this is the omission of copulas and auxiliaries, which have little effect on sentence semantics. Consider, for example, *The patient is awake* vs. *The patient awake*.

In some cases, it is even possible to omit the main verb of a sentence. For instance, the structure

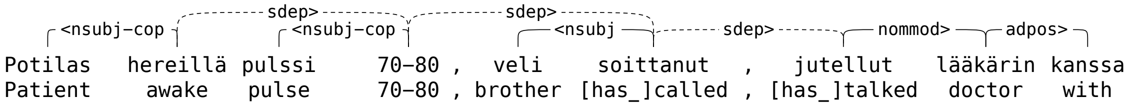

Figure 4: The purpose of the *sdep* dependencies is to combine the independent sentences under one surface sentence into a single analysis. Without the dashed *sdep* dependencies, the analysis would contain separate islands. This sentence can be roughly translated as *Patient awake pulse 70-80, brother called, talked with the doctor*.

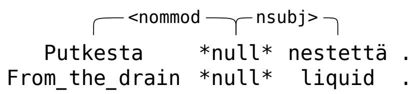

Figure 5: Missing main verbs are represented by a null verb, in order to construct a dependency analysis for sentences such as this. The sentence can be roughly translated as *Liquid from the drain*.

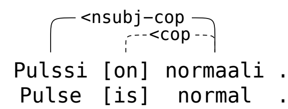

Figure 6: The new dependency type *nsubj-cop*, used instead of *nsubj* in copula clauses. Note that the analysis stays essentially the same, regardless of the presence or absence of the copula.

Putkesta nestettä (*Liquid from the drain*) is common in ICU Finnish, though it would be judged fragmentary in standard Finnish. Here, the case of the noun *putkesta* (*from the drain*) expresses the direction of the liquid, and the actual verb (*to come*) can therefore be omitted, as its meaning is clear in the context. This poses a problem for most dependency schemes, as the main verb of a clause is also its head word. To be able to analyze the sentences with a missing main verb (21% of the sentences in the corpus), we have manually introduced a *null verb* in those sentences to represent the missing verb. See Figure 5 for an illustration of this solution.

Because the purpose of the null verb is to represent a word that is absolutely necessary for the construction of an SD analysis, null verbs are introduced only when the main verb is omitted. Copulas and auxiliaries never act as governors in the SD scheme and thus do not require a null verb to be inserted.

Finally, the frequent omissions of copulas require another minor modification to the SD scheme, the introduction of the dependency type *nsubj-cop*. The *nsubj* type used in the original SD scheme for both standard and copula subjects is in our version of the scheme replaced by *nsubj-cop* in copula clauses. This is to differentiate the special case of copula subjects, where, in the SD scheme, the governor of the dependency is not a verb but, for example, an adjective. For an illustration of the use of *nsubj-cop*, see Figure 6.

4 Performance measures

When evaluating the quality of our corpus, as well as the performance of the parsers in the experiments described below, we use the following measures.

Precision (P) is defined as the proportion of dependencies in the parser output that are also present in the gold standard. *Recall (R)*, in turn, is the proportion of dependencies in the gold standard that are also present in the parser output. These two are combined into an F*-score*, defined as $F = \frac{2PR}{P+R}$.

Labelled attachment score (A_L) is the proportion of tokens that are assigned the correct head and dependency label according to the gold standard, and *unlabelled attachment score* (A_U) is the proportion of tokens that are assigned the correct head, regardless of the dependency label (Nivre, 2008a). Note that A_L and A_U are defined for tree structures where each token has exactly one head. As noted previously, analyses in the SD scheme are not necessarily trees, and thus the two measures are not directly applicable to it.

5 Corpus annotation and statistics

As one of the primary contributions of this work, we have annotated a corpus of 1019 ICU Finnish sentences with 7614 tokens of which 6082 are non-punctuation. The text of the corpus consists of notes written by nurses about the condition of a patient, often with respect to standard topics such as breathing, hemodynamics, diuresis and relatives.

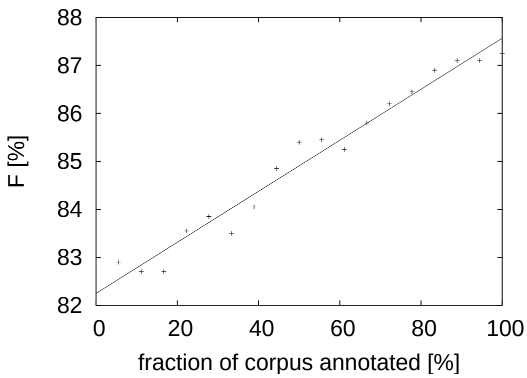

Figure 7: Inter-annotator agreement in F-score at various stages of the corpus annotation with a trend line. Note that the A_L and A_U measures are not reported, as the SD analyses are not necessarily trees.

The corpus currently consists of sentences from four different patient reports, as we decided to annotate full reports rather than randomly selected individual sentences, to enable further research, for example in report summarization.

The dependency annotation has in total 5194 dependencies. Only 2.9% of all sentences and 0.5% of all tokens are non-projective. The effect of non-projectivity on parsing ICU Finnish is thus negligible.

We used full double annotation, that is, each sentence was independently annotated by two annotators, and disagreements were jointly resolved. To evaluate the quality of the corpus, we measured inter-annotator agreement, defined as the average of the agreements of the two annotators against the final annotation. The average inter-annotator agreement on the whole corpus was 87.25% F-score. Figure 7 illustrates the growth of the inter-annotator agreement as the annotators become familiar with the task and the scheme.

We estimate that the current corpus has taken 70 man-hours of annotation work to develop, including both the independent annotation work by individual annotators and the joint resolving of disagreements. The disagreement resolving took in total approximately 30 man-hours. We used a custom software for annotation and disagreement resolution.

6 Experiments on the corpus

In this section, we discuss the experiments that the newly built corpus has enabled us to perform. We first describe our experiments on the rule-based approach, including the conversion rules required for the evaluation of the parser. We then present results of another experiment, which uses a statistical approach.

In order to be able to use the A_L and A_U performance measures described in Section 4, as well as to maintain comparability of results with MaltParser which produces tree analyses, the treeness of all analyses in all experiments was assured by breaking the possible cycles present in the gold standard. Punctuation tokens were excluded from all performance measurements and the null verbs representing omitted verbs were preserved in the parser input.

6.1 Parsing experiments with a rule-based parser

As the first part of our experiments, we apply the rule-based parser of Laippala et al. (2009) whose reported coverage is up to 75% of ICU Finnish sentences with an oracle best parse performance of above 90% in terms of the PARSEVAL metric (Black et al., 1991).

6.1.1 The dependency conversion

The parser natively produces constituency output. Thus, in order to evaluate the parser on the ICU Finnish corpus as well as to improve its applicability in the domain, we produce a conversion from this constituency scheme to the SD scheme. Note that, as illustrated in Figure 8, using a constituency scheme for ICU Finnish often results in complex representations which do not contain information about syntactic roles of the constituents. Inferring these roles is one of the aims of our conversion.

The conversion is implemented using handwritten rules. The parser assigns a head word for each phrase, and these heads are then used to produce the structure of the dependency graph by placing dependencies from the head word of each constituent to the head words of its subconstituents. The conversion rules are generally only needed to assign types to these dependencies. There are few exceptions, such as the *sdep* dependencies (see Section 3.2.3) and certain auxiliary structures, where the structure in the SD scheme does not correspond to that induced from the head words. The rules can restrict on the structure of a subtree, that is, a rule can require a phrase as well as its sub-phrases, at any depth, to be of specific types. Our conversion approach closely fol-

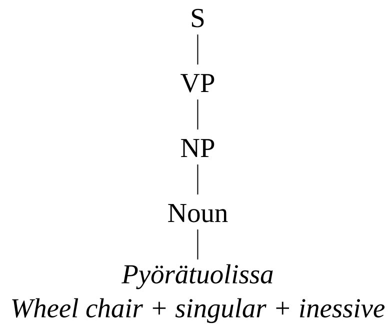

Figure 8: The constituency output of the parser of Laippala et al. (2009). The example sentence can be roughly translated as *In wheel chair*. The direct derivation of the VP from the NP is explained by the missing main verb that would in a corresponding SD analysis be represented by a null verb. Note the size of the tree, despite the fact that the sentence only consists of one word.

lows that of the Stanford tools (de Marneffe et al., 2006), as both utilize heads of phrases and subtree search to produce the structure and labels of the dependency parse.

The conversion rules were developed using the 80-sentence development set previously used by Laippala et al. (2009). We have annotated these sentences in the SD scheme to complement their existing constituency annotation.

6.1.2 Performance of the parser and conversion rules

When interpreting the results it is crucial to note that the rule-based parser does not have a ranking component that would select a single preferred analysis among the generated parses. The parser generates, on average, 33 parses per sentence and the figures reported are measured using the best parse with respect to the labelled attachment score (*oracle performance*). Further, the coverage of the parser in terms of the proportion of sentences that receive at least one analysis is 75% on our corpus and the performance values reported are calculated on these sentences, disregarding sentences that receive no analysis. The results are thus rather an upper limit of the performance to be expected in a real-world setting.

We find that the rule-based parser augmented with our conversion achieves an A_L of 75.2%, A_U of 84.5%, and F-score of 70.2%. Given the A_U of 84.5%, the parser itself assigns incorrect heads for 15.5% of tokens. This is the starting point for the conversion rules, which result in the overall A_L of 75.2%. The difference of 9.3 percentage points between A_U and A_L is divided between errors of the conversion rules and errors of the parser who may assign correct heads but incorrect nonterminal labels, thus preventing correct interpretation of the parse. To establish this division of errors, we have performed a limited manual analysis of 16 randomly selected sentences (75 dependencies) and found that the conversion rules are responsible for 5.3 percentage points and the parser and FinTWOL for the remaining 4 percentage points.

6.2 Statistical parsing experiments with MaltParser

To complement the rule-based dependency parsing experiments, we also apply a statistical parser induced from the ICU Finnish corpus using the MaltParser system[4] (Nivre et al., 2007). We use the arc-eager parsing algorithm characterized as a deterministic, linear-time algorithm that generates a single projective dependency tree in a left-to-right pass through the sentence. The choice of a projective parsing algorithm is justified by the negligible amount of non-projective tokens in the corpus. The algorithm is based on the well-known shift-reduce bottom-up parsing strategy that processes the sentence from a token queue and maintains a stack of partially-processed tokens. At each point in the parsing process, the next transition applied by the parser is decided by a support vector machine (SVM) classifier based on features extracted from the sentence tokens as well as the partially-built dependency tree.

In training the parser, we use the MaltParser default feature model for the arc-eager parsing algorithm. Broadly stated, this model considers morpholexical properties of the first four tokens in the queue and the first two tokens on the stack as well as partially-built dependency structure features of the top items on the stack and the queue. The corpus text is first morphologically disambiguated using FinCG, thus obtaining a single morpholexical reading for each token. A separate feature is then generated for each morpholexical property produced by FinCG[5] for a given token (e.g. the POS, number, and case). Whenever the token wordform does not carry a particular property (e.g. nouns do not have a tense and verbs do not have a case), the

[4] Version 1.2, http://www.maltparser.org

[5] See http://www2.lingsoft.fi/doc/fintwol/intro/tags.html for the full set of tags given by FinTWOL/FinCG

feature is set to *null*. Rather than wordforms, we use word lemmas in the feature model to reduce training data sparseness.

All results reported in this section are obtained using ten-fold cross-validation, where in each fold 80% of the data is used for training, 10% for parameter estimation, and 10% for testing. In preliminary experiments on a small portion of the data, we selected the second degree polynomial kernel for the parser SVM classifier. The values of the SVM regularization parameter C and the kernel parameter γ were selected for each fold separately, using a joint grid search on the parameter estimation set. The best-performing parameter combination in terms of A_L on the parameter estimation set was then used in parsing the test set, thus avoiding parameter over-fitting. All other parameters were left at their default values.

The results are shown in Table 1 for varying sizes of the training sets, in order to estimate the learning curve of the parser. The overall parser performance, 69.9% A_L, can be contrasted with the results of Nivre (2008a) who reports an average A_L of 79.77% across 13 languages. The results for individual languages, however, range from 64.7% for Turkish to 90.1% for Japanese. In that respect, the results for ICU Finnish are among the lower ones, but arguably well within the typical range to be expected. This is particularly encouraging given that the ICU Finnish corpus is currently relatively small, consisting of 1019 sentences and 6082 non-punctuation tokens. As a point of comparison, Nivre has used corpora of 5000 sentences with 58000 tokens, and 17000 sentences with 151000 tokens for Turkish and Japanese, respectively.

The statistical parser yields a lower absolute performance than the rule-based parser. However, the two results are not directly comparable. First, the oracle best-parse strategy had to be used for the rule-based parser. Second, the results of the rule-based parser include only those sentences for which the parser has given at least one analysis (75% of all sentences). Taking these measurement limitations into account, it would seem likely that with a larger corpus available for training and other further improvements, a statistical parsing approach based on MaltParser will be preferable over the rule-based parser of Laippala et al. It is worth noting that the parsing speed of the statistical parser is on the order of 10 sentences per second, whereas the rule-based parser parses one sentence in approximately 2 to 3 seconds.

sample [%]	A_L[%]	A_U [%]	F [%]
100	69.9±2.0	77.1±2.5	66.6±2.2
75	68.4±2.8	75.8±2.2	65.0±3.2
50	65.8±2.0	73.6±1.5	62.0±2.3
25	57.2±2.7	67.5±1.7	52.6±3.2

Table 1: MaltParser results with varying training set size. The *sample* column gives the size to which the training sets in the ten-fold cross-validation were downsampled. Performance figures are given together with their standard deviation on the ten folds.

7 Conclusions and discussion

In this paper, we have presented a new syntactically annotated corpus of ICU Finnish, the language used in daily nursing notes in an intensive care unit. The corpus is annotated in the Stanford dependency scheme which we find suitable for ICU Finnish with only minor modifications. We have performed parsing experiments on this corpus using two approaches: by converting the constituency output of an existing rule-based parser (Laippala et al., 2009) to a dependency scheme, and by inducing a statistical parser from the new corpus using MaltParser (Nivre et al., 2007).

The rule-based parser, together with the constituency-to-dependency conversion developed for the purposes of this work, achieved the oracle labelled attachment score of 75.2%. In a separate evaluation of the conversion rules, we find that the rules contribute roughly 5 percentage points to the overall error rate.

The statistical parser trained on the rather modestly sized corpus achieved a labelled attachment score of 69.9%, approaching the results presented by Nivre (2008a) for parsers trained on significantly larger corpora. The comparability of results of the rule-based and the statistical parsers is difficult to establish given that the rule-based parser does not provide a single preferred analysis.

Our results on the statistical parsing of ICU Finnish, particularly encouraging when taking into consideration the modest size of the corpus, might suggest that full parsing of the intensive care language is, perhaps somewhat counterintuitively, not a very difficult task, relative to the general language. For a more definitive conclusion, a considerably broader study, beyond the scope of this paper, would need to be performed. In particular, possible features allowing the parser to better capture the idiosyncrasies of the ICU sublanguage need to be explored more thoroughly.

The first obvious future work direction is to further increase the size of the corpus and find a legal way to release the corpus annotation while protecting patient privacy. One option could, for example, be to release an unlexicalized version of the corpus with morphological and syntactic annotation only. The second direction is to complement the preliminary experiments with MaltParser presented in this paper by carefully exploring the possible feature models, parsing algorithms and parser training parameters in order to maximize the performance of the induced parser. The final direction is to develop a method for inserting the null verbs necessary in the dependency analysis, either as a separate pre-processing step, or directly as part of parsing.

Acknowledgments

We would like to thank Lingsoft Ltd. for making FinTWOL and FinCG available to us. This work has been supported by the Academy of Finland and Tekes, the Finnish Funding Agency for Technology and Innovation.

References

E. Black et al. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, pages 306–312.

J. E. Carroll, E. Briscoe, and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of LREC'98*, pages 447–454.

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

C. Friedman and S. Johnson. 2006. Natural language and text processing in biomedicine. In *Biomedical Informatics*, pages 312–343. Springer.

Y. Huang and H. J. Lowe. 2007. A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American Medical Informatics Association*, 14(3):304–311.

Y. Huang, H. J. Lowe, D. Klein, and R. J. Cucina. 2005. Improved identification of noun phrases in clinical radiology reports using a high-performance statistical natural language parser augmented with the UMLS Specialist Lexicon. *Journal of the American Medical Informatics Association*, 12(3):275–285.

F. Karlsson. 1990. Constraint Grammar as a framework for parsing unrestricted text. In *Proceedings of COLING'90*, pages 168–173.

K. Koskenniemi. 1983. Two-level model for morphological analysis. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 683–685.

V. Laippala, F. Ginter, S. Pyysalo, and T. Salakoski. 2009. Towards automatic processing of clinical finnish: A sublanguage analysis and a rule-based parser. *International Journal of Medical Informatics, Special Issue on Mining of Clinical and Biomedical Text and Data*. In press.

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

M-C. de Marneffe and C. Manning. 2008. Stanford typed hierarchies representation. In *Proceedings of COLING'08, Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

M-C. de Marneffe, B. MacCartney, and C. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*, pages 449–454.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

J. Nivre. 2008a. Deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

J. Nivre. 2008b. Sorting out dependency parsing. In *Proceedings of GoTAL'08*, pages 16–27.

S. Pyysalo, S. Aubin, A. Nazarenko, and T. Salakoski. 2006. Lexical adaptation of Link Grammar to the biomedical sublanguage: a comparative evaluation of three approaches. In *Proceedings of SMBM'06*, pages 60–67.

G. Schneider, F. Rinaldi, and J. Dowdall. 2004. Fast, deep-linguistic statistical dependency parsing. In *Proceedings of COLING'04 Workshop on Recent Advances in Dependency Grammar*, pages 33–40.

P. Szolovits. 2003. Adding a medical lexicon to an English parser. In *Proceedings of AMIA'03*, pages 639–643.

ISSN 1736-6305 Vol. 4
http://hdl.handle.net/10062/9206