Recognizing noun phrases in biomedical text: An evaluation of lab prototypes and commercial chunkers

Joachim Wermter [a] **Juliane Fluck** [b] **Jannik Stroetgen** [b] **Stefan Geißler** [c] **Udo Hahn** [a]

[a] Jena University Language and Information Engineering (JULIE) Lab
http://www.uni-jena.de/coling.html
[b] Fraunhofer Institute – SCAI Bioinformatics
http://www.scai.fraunhofer.de/bio.html
[c] TEMIS Deutschland GmbH
http://www.temis-group.com

Abstract

In the biomedical domain, many systems for text mining and information extraction rely on basic morphological and syntactic analysis such as part-of-speech tagging or noun phrase (NP) chunking. Due to the lack of sufficient in-domain resources these systems often make use of NLP tools trained and evaluated on newspaper-language training sets. Scientific texts in the life sciences, however, differ from general language in the structure and complexity of noun phrases. Therefore, we tested the effects this domain change has on the performance of these systems.

For this purpose, we compared three prototype chunking systems developed in research labs (all based on statistical learning methods) and one chunking system which is part of a commercial information extraction toolkit (based on manually supplied grammar specifications). Trained on PENN TREEBANK *tagging and chunking annotations for newspapers, we ran these systems on the* GENIA *treebank which contains such annotations for biological abstracts taken from* MEDLINE*. We, first, observed a significant over-all loss in performance (on the order of 4%) and, second, found (with the exception of the SVM-based system) no significant difference between the performance of lab prototypes and the commerical chunker on* GENIA *data. Fortunately, the performance loss can also be partly remedied by few biomedical domain-specific adaptations.*

Introduction

In the life sciences domain a large fraction of information is only available in form of unstructured free text. For molecular biology, genome-based clinical research and medicine, this comes in the form of technical reports and scientific articles. By now, the sheer volume of literature and medical narratives makes it almost impossible for biologists, clinical researchers and medical professionals to retrieve all relevant information on a specific topic and to keep up with current research. Fortunately, the field of human language technology makes available various tools for text mining in order to automatically extract relevant information contained in free text. Their benefits are to filter out relevant information, to extract structured knowledge from large text collections, or to support database curators and providers in choosing the most relevant data and extracting relevant data for filling up database entries faster.

Many NLP applications are composed of different levels of text analysis. A basic processing step consists of the assignment of part-of-speech tags to text tokens. A subsequent step after tagging focuses on the identification of basic structural relations between groups of words. This recognition task is usually referred to as noun phrase (NP) chunking. Because both of these techniques are particularly beneficial for named entity recognition,[1] they are (among others) widely used in applications of text mining in the life sciences domain (cf., e.g., [13], [15], [9]). It should be noted, however, that most of these studies make direct use of NLP tools for part-of-speech (POS) tagging and chunking that were developed for general-purpose newspaper language and whose performance on biomedical language has not been evaluated so far. The question thus arises whether such tools – and, if so, which ones – are portable to the biomedical domain without a drastic performance loss. Results on part-of-speech tagging indicate different methods vary as for performance loss [4] when domains and text genres are exchanged.

In this paper, we focus on the exemplary evaluation of noun phrase chunking in the biomedical domain and look at two very different types of chunking tools:

- We examine three general-purpose chunkers which rely on statistical machine learning techniques and are trained on a common newspaper-language corpus: YAMCHA [5] a kernel-based support vector machine system, BOSS, a statistical chunking tool developed at Jena University's Language and Information Engineering Lab, and TBL [14], a base NP

[1] For example, in the biomedical domain, virtually all named entites, such as protein, gene or cell names, are linguistically expressed as noun phrases.

chunking tool that learns transformation rules. Especially the performance of the latter one is of interest since a number of recent studies in need of a noun phrase chunker use it for the various biomedical text mining tasks (e.g., [6], [11]).

- We also looked at a commercially deployed system (TEMIS), which is used in different application domains in industrial contexts. The grammar rules it employs are hand-crafted. Text mining systems which employ manually supplied rules are easier to adapt to different domains because the in-domain training corpora required for machine-learning methods are often not available or only costly to establish. Still, such manually maintained systems tend to be incomplete and error-prone. Thus, a commercial system's usability for noun phrase recognition, alongside with research-only machine-learning methods, should be particularly illuminating. Moreover, to our knowledge, this is the first study to evaluate a commercially deployed system in the biomedical domain according to the scientific community's evaluation standards.

Methods

In this section, we, first, describe the corpora which we used for training and testing. Second, we introduce the four types of chunking systems we assess.

The Training and Test Environments

All three lab prototypes were trained on the standard data set for base NP[2] chunking put forward by Ramshaw and Marcus [14], viz. sections 15-18 of the Wall Street Journal part of the PENN TREEBANK [7]. This benchmark set amounts to 211,727 tokens which were part-of-speech (POS-)tagged [2] with the PENN TREEBANK (PTB) tagset and chunk-annotated using the standard Inside/Outside (or IOB[3]) chunk representation, first introduced by Ramshaw and Marcus [14] and since then canonically applied to base NP chunking. Typically, ML-based chunking systems make use of the available types of linguistic information (i.e., word and POS information) in the training corpus in order to estimate their model parameters.

TEMIS, the commercial system, on the other hand, not only uses its own hand-crafted set of grammar rules (adapted to the same standard PTB training set – see the subsection on the TEMIS system below) but also its own internal XELDA [12] tagger and tagset. Thus, it is necessary to interpret the performance values obtained by the different systems accordingly.

The test set on which we evaluated the different systems was derived from the Beta version of the GENIA treebank[4], a subset of the GENIA corpus [10], which comprises 200 syntactically annotated MEDLINE abstracts from the molecular biology domain. Although GENIA is POS-tagged using the PTB tagset, its POS-annotation scheme had to be changed (and is thus different to the PTB scheme) to account for various properties specific to text from the molecular biology domain [19] Among these are (non-proper) names beginning with capital letters (e.g., "NFAT", "RelB"), chemical and numeric expressions including non-alphanumeric characters such as commas, parentheses, or hyphens (e.g., "beta-(1,3)-glucan"), participles of unfamiliar verbs describing domain-specific events, and fragments of words (e.g., "up- and down-regulate").

To conform to already established evaluation metrics [16], the GENIA treebank was automatically converted to the IOB-format described above (see also Table 1). We thus obtained a data set which runs 44,914 tokens in size. From this, we split off one quarter (∼11,246 tokens) as a development set to allow the TEMIS system output format to be formatted according to the IOB chunk tag notation. The remaining 33,668 tokens served as the actual test set.

a	DT	I
mechanism	NN	I
that	WDT	B
increases	VBZ	O
NF-kappa	NN	I
B/I	NN	I
kappa	NN	I
B	NN	I
dissociation	NN	I
without	IN	O
affecting	VBG	O
the	DT	I
NF-kappa	NN	I
B	NN	I
translocation	NN	I
step	NN	I

Table 1: The standard IOB chunk tag notation

[2] *Base NPs* are defined as non-recursive noun phrases ending after their nominal head and excluding any type of postmodification (e.g., prepositional phrases, attributes, appositions). Base NP recognition is an often used representative task to compare different NLP methods.

[3] I = current token is inside of a chunk, O = current token is outside of any chunk, B = current token is the beginning of a chunk immediately following another chunk.

[4] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/topics/Corpus/GTB.html`

Learning-based lab prototypes

We examined the following lab prototypes for chunking, all based on statistical methods of machine learning:

YAMCHA**.** YAMCHA [5] is an open source text chunker based on so-called Support Vector Machines (SVMs). Typically, SVMs are binary classifiers and thus must be extended to multi-class classfiers to classify three (as in the case for NP chunking with (I,O,B) or more classes – see [17] for the underlying statistical learning theory). Typically, they map their n-dimensional input space into a high-dimensional feature space in which a linear classifier is then constructed. Generally, this approach requires considerable computational resources. As a consequence, various methods are employed by YAMCHA [5] to reduce the training costs incurred by this approach.

TBL**.** Transformation-based error-driven learning [14] starts with a training corpus specifying the correct values for the linguistic features of interest, a baseline heuristic for predicting initial values for these features and a set of rule templates that determine a space of possible transformational rules. Model learning is achieved by iteratively testing and improving hypotheses using the rule templates. TBL turned out to be one of the standard systems used for base NP chunking.

BOSS**.** The BOSS chunking system developed at Jena University's Language and Information Engineering (JULIE) Lab predicts borders of noun phrases (beginning and end points) based on statistical criteria.[5] These predictions are estimated by combining the observed probabilities of NP borders and NP POS patterns in a training corpus. The challenge then is to pair the predicted borders in an 'optimal' way into non-overlapping phrases. BOSS, in analogy to [8], finds the pairing with the maximal value using a shortest-path algorithm. At its current development stage, BOSS is comparatively knowledge-poor since it only uses POS information from the training corpus, whereas both YAMCHA and TBL, in addition, integrate lexical and word feature information.

TEMIS **– The hand-coded commercial system**

Architecture of the TEMIS **system.** The experimental setup for the TEMIS system initially ignored the GENIA POS tags, using only the plain text part of the corpus. This was fed into a processing chain that uses the XELDA toolkit [12] to compute morphological information using finite-state transducers to implement a two-level morphology [1]. The resulting potentially ambiguous chain of POS tags is then disambiguated using a HMM POS tagger. Finally, transducers are again employed to apply a finite-state grammar to construct larger phrases such as NPs from the tagged input.
This processing chain is implemented in the commercial TEMIS system, which is applied to information extraction tasks in a wide range of different application settings. Because the system is designed to facilitate grammar development in several languages, the underlying tagset is common to all languages and applications. This is a compromise satisfying as good as possible the requirements in each of the languages addressed. Still, this tagset does not stand in a well-defined relation to commonly used tagsets such as the PTB tagset.

Adjustment to the Evaluation. A first quick comparison of the NPs returned by the TEMIS system and the NPs given on the development set showed that the TEMIS developers had to modify the NP grammar in order to account for the differing interpretations of what should be an NP. In particular, the complex NPs (NPs with embedded PPs or genitives, etc.) accounted for many cases where the TEMIS results differed from the intended base NP target representation (see Table 1). For example, in *[[the synthesis]$_{NP_{base}}$ of [long enhancer transcripts]$_{NP_{base}}$]$_{NP_{complex}}$* or in *[[Hashimoto]$_{NP_{base}}$ ['s thyroiditis]$_{NP_{base}}$]$_{NP_{complex}}$* two or more base NPs connect to one complex NP in the TEMIS grammar, because for commercial-type information extraction there is no need to analyze such individual base NPs separately. To be in line with the base NP target representation and to be comparable to the ML-based systems, however, such complex NPs were split up into their base components using the patterns provided by the PENN TREEBANK .

Translation of the TEMIS**-internal tagset.** Because the TEMIS system uses its own XELDA tagset and thus ignores the POS information for chunking provided in the GENIA test set, we wanted to test whether a manual translation of those 28 XELDA tags, which are relevant for NP chunking, to the ones used by GENIA would be beneficial for the commercial software. Due to their different underlying design principles, a simple one-to-one mapping between the relevant tags in both tagsets is not always possible. Hence, we had to distinguish three types of mapping:

- In 28.5% (6/28) of the cases, there is a one-to-one correspondence, e.g., the XELDA tag for coordinations (COORD) corresponds to the GENIA tag (CC).

[5] Viewing noun phrase recognition as a border finding problem was first introduced by Church [3].

Default	PTB corpus			GENIA corpus		
	Recall	Precision	F-score	Recall	Precision	F-score
YAMCHA	94.29	94.15	94.22	89.00	89.30	89.15
BOSS	89.92	90.10	90.01	86.46	86.84	86.65
TBL	92.27	91.80	92.03	86.31	85.49	85.90
TEMIS$_{XeLDA}$	86.94	86.29	86.61	87.14	85.34	86.23

Table 2: Benchmark results of the different systems as default.

Domain-specific Adaptations	GENIA corpus		
	Recall	Precision	F-score
TEMIS$_{Genia}$	91.24	90.59	90.91
BOSS$_{Par}$	87.25	89.19	88.21

Table 3: The TEMIS-internal tagset was translated into the tagset used by GENIA. BOSS uses a pattern which recognizes NP-internal parentheses.

- In 25% (7/28) of the cases, a XELDA tag corresponds to more than one GENIA tag. For example, the word "there" is always tagged as an adverb in XELDA although it actually can have a nominal function such as in existential constructions ("There is quite a bit ..."). In GENIA, "there" receives an RB tag in its adverbial function, while in its existential one it gets an EX tag.

- In 46.5% (13/28) of the cases, more than one XELDA tag corresponds to one GENIA tag. For example, XELDA distinguishes between personal (PronPers) and reflexive pronouns (PronRefl), whereas GENIA only uses one tag (PRP) for both.

Results

Two Series of Experiments

In evaluating the performance on the GENIA test set, we ran two series of experiments. The first one used all systems in their default configuration, except for the adaptation of the TEMIS software (see the description in the subsection describing this adjustment). For the ML-based systems, their parameters from their PENN TREEBANK training were left unchanged. In the second series of experiments, we made some in-domain adaptations on two systems. For TEMIS, we used the GENIA-translated tagset. For BOSS, a simple additional bio-domain-specific pattern was introduced, which recognizes noun phrases with internal parentheses (such as in *interleukin 2 (IL-2) activation*).

Evaluation of the Different Systems

The three lab prototype systems based on machine learning techniques were all trained and tested on the same PENN TREEBANK (PTB) general-language newspaper corpus data set.[6] The adjusted NP grammar for TEMIS was also based on the PTB corpus. Table 2 contains the performance figures of these four systems on the GENIA corpus. The results for the two adapted system (TEMIS$_{Genia}$ and BOSS$_{Par}$) are reported in Table 3.

As far as the default systems are concerned, the YAMCHA kernel-based support vector machine performs best on both corpora, but looses approximately 4 percentage points of performance (from an F-score of 94.22% to 89.15%) for the GENIA corpus. The TBL method, which performs second best on the Penn TreeBank corpus (F-score: 92.03) performs worst on the biomedical corpus (with a F-score of 85.9%). Of all ML-based systems, the BOSS system has the lowest performance on the PENN TREEBANK corpus but faces the least loss (only 3.35 percentage points) on the GENIA corpus, on which it performs second best. Its comparatively low performance on PTB can be explained by the fact that it only utilizes POS information for chunking but no lexical information like the two other ML-based systems do. In comparison to the ML-based methods, the performance of the grammar-based TEMIS$_{XeLDA}$ system on GENIA lies between the BOSS and the TBL method. Due to its low performance on the PENN TREEBANK (F-score: 86.61), to which this base NP grammar has been adapted, the loss is only 0.38 percentage points. A detailed error analysis is given in the next section.

Overall Error Analysis for the Default Systems

For error analysis, the false negative hits (i.e., tokens that were not recognized as part of a noun phrase) as well as the false positive hits (i.e., tokens that were er-

[6] The results for YAMCHA and TBL are reported in [5] and [14], respectively.

roneously identified as part of noun phrase) are sorted with the help of the positional IOB chunk tag information. The hits were then compared in a pair- and n-wise fashion between the different systems and thus allowed them to be examined as to whether they assign the same erroneous IOB chunk tag to the same token, i.e., their common mistakes could be identified (see Figures 1 and 2).

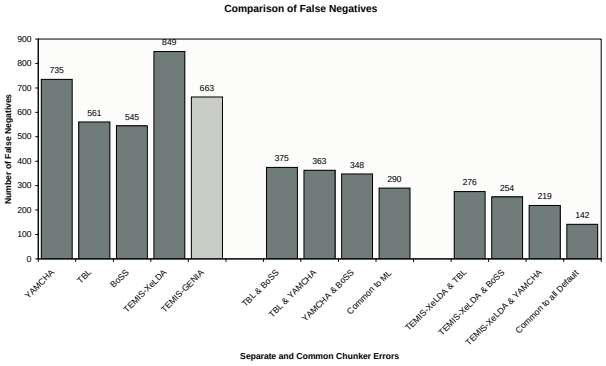

Figure 1: False Negative (FN) errors based on postitional IOB chunk tag information

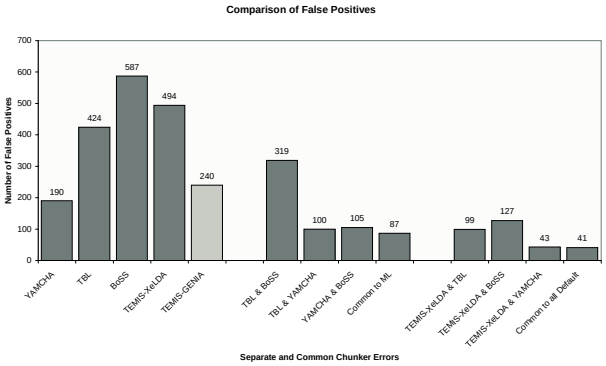

Figure 2: False Positive (FN) errors based on postitional IOB chunk tag information

For the false negative (FN) hits, the TEMIS$_{XeLDA}$ software made the most mistakes (849), followed by the YAMCHA system (735). The other two systems have very similar error rates. The overlap of mistakes is the highest between the ML-based systems. The proportion of common mistakes between all three systems is 53.2% according to the system with the lowest error rate (BOSS), and varies between 63.9% and 68.8% on pairwise comparisons. According to the ML system with the highest error rate (YAMCHA), 39.5% of all mistakes are common to the three ML-based systems, whereas the overlap on a two-system comparison basis ranges from 47.3% to 66.8%. Scaled by the system with the lowest error rate, the error overlap between the TEMIS$_{XeLDA}$ software and each ML-based systems ranges between 29.8% (YAMCHA) and 49.2% (TBL); between all systems the error overlap is 26%. Scaled by the TEMIS$_{XeLDA}$ system, the overlap only ranges from 25.8% (YAMCHA) to 32.5% (TBL) for pairwise comparisons and only reaches 16.7% comparing all systems.

For the false positive (FP) rates, the BOSS system made the most errors (587) followed by the TEMIS$_{XeLDA}$ system (494). The YAMCHA SVM performs by far the best with the lowest error rate (190). Under the FP condition, the pairwise overlaps between the ML-based methods are also higher than between the TEMIS$_{XeLDA}$ software and each ML system. In particular, the overlap between BOSS and TBL is very high (75%) in comparison to 53% and 55% from these systems to YAMCHA.

Error Type Analysis for Default Systems

Although the false negative/positive error rates shed some light on the *overall* performance of each system, they alone do not *explain* the performance on the GENIA corpus. Therefore, we tried to identify the most common *error types* across the different systems by looking at the part of speech and the context of each false negative/positive hit (see Tables 4 and 5).

There were certain linguistic constructions around which error types could be established for FNs, i.e. tokens that were not recognized as part of a noun phrase with such a linguistic property, and for FPs, i.e. tokens that were erroneously identified as part of noun phrase with such a linguistic property. The following list enumerates the most salient ones:[7]

- NPs with coordinated/enumerated elements (Coord), e.g.,
  FN: *new DNA binding proteins of 85, 75 and* 54 kDa*
  FP: *Cyclosporin A and FK506 inhibit T- and B-cell activation and* other processes*

- NPs with internal parenthesized/bracketed elements (Par), e.g.,
  FN: *chloramphenicol acetyl-transferase (CAT)* gene expression*
  FP: *human immunodeficiency virus type 1 (HIV-1)**

- NPs with verbal forms in prenominal adjective function, (Verbal), e.g.,
  FN: *from resting* and induced* ML-1 cells*
  FP: *a specific target termed* TAR*

[7]The underscored items marked with an asterisk (*) are misclassified by some or all systems as FNs or FPs with respect to their correct IOB chunk tag.

Method	Coord	Par	Verbal	Adv	Adj	Noun	Det
YAMCHA	52.0 (382)	21.6 (159)	8.3 (61)	3.3 (24)	3.3 (24)	3.5 (26)	0 (6)
BOSS	30.5 (166)	34.7 (189)	12.1 (66)	2.2 (12)	5.0 (27)	1.7 (9)	0 (1)
TBL	37.4 (210)	23.4 (131)	14.8 (83)	3.7 (21)	7.8 (44)	1.1 (4)	0 (6)
TEMIS$_{XeLDA}$	26.0 (221)	5.8 (49)	8.2 (70)	11.2 (95)	8.6 (73)	22.5 (191)	7.5 (64)
TEMIS$_{GENIA}$	41.5 (275)	8.0 (53)	8.6 (57)	3.6 (24)	13.6 (90)	0 (0)	12.3 (85)

Table 4: Distribution of error types (in %, with absolute numbers in parentheses) for false negatives, i.e. tokens that were not recognized as part of a noun phrase chunk.

Method	Coord	Par	Verbal	Adv	Adj
YAMCHA	40.5 (77)	9.5 (18)	4.7 (9)	6.3 (12)	20.5 (39)
BOSS	52.3 (307)	1.2 (7)	8.5 (50)	6.3 (37)	11.8 (69)
TBL	60.1 (255)	6.8 (29)	9.2 (39)	3.8 (16)	9.2 (39)
TEMIS$_{XeLDA}$	23.1 (114)	13.0 (64)	32.6 (161)	3.0 (15)	14.2 (70)
TEMIS$_{GENIA}$	41.3 (99)	6.7 (16)	11.6 (28)	15.8 (38)	13.3 (32)

Table 5: Distribution of error types (in %, with absolute numbers in parentheses) for false positives, i.e., tokens that were erroneously recognized as part of a noun phrase chunk (error types listed in Table but not here are irrelevant).

- NPs with adverbs modifying prenominal elements (Adv), e.g.,
  FN: *abnormally* low plasma cysteine levels*
  FP: *Together* these results constitute...*

- Adjectives (Adj) in various functions, e.g.,
  FN: *the expression of endogenous AP-1 regulated* genes*
  FP: *16 patients, aged* 16-27 years,...*

- Nouns/nominal elements (Nom), e.g.,
  FN: *lymphocyte glucocorticoid receptor binding* parameters*

- Determiners (Det), e.g.,
  FN: *the* human and murine TNF genes*

In terms of the error type distribution, the most frequent type for FN and FP errors is the recognition of coordination elements. This is a dominant error source for the ML-based systems (YAMCHA: 52% FN and 40.5% FP; BOSS: 52.3% FP), except for BOSS, whose most common FN error type are parenthesized elements (34.7%), and the commercial TEMIS$_{XeLDA}$ system, which does a better job at not erroneously FP-recognizing coordinative elements as part of a noun phrase (26.0%). The BOSS system, in particular, very often erroneously recognizes coordinative elements[8] as part of an NP, which must be attributed to the fact that it does not utilize any lexical information. As for FNs, noun phrases with parenthesized elements, as well as verbal forms in prenominal adjective functions, are other common error sources. NPs with such internal parenthesized/bracketed elements are special to the biomedical domain and, thus, their higher amount of false negative errors can be explained. It seems, however, that the grammar-based TEMIS$_{XeLDA}$ system does a better job in recognizing these elements as part of a noun phrase (5.8% FN) than the ML systems (YAMCHA: 21.6% FN; BOSS: 34.7% FN; TBL: 23.4% FN). On the other hand, it also FP-recognizes them more often erroneously (13.0%).

[8] This mistake is also responsible for its overall high number of FP errors.

Although FN coordination is also a prominent error source for the commercial TEMIS$_{XeLDA}$ software (26%), its FN error type distribution exhibits other error sources which are virtually absent from the ML systems. Most strikingly, these are nominal elements (21.8%) and determiner elements (7.7%).[9]
Another noteworthy difference between the ML-based system and TEMIS$_{XeLDA}$ is the high amount of FP verbal elements (32.6%) for the latter one.

In-domain Adaptations

In the second round of experiments we tested whether some in-domain heuristic adaptations both on the commercial TEMIS software and on the ML-based BOSS system would lead to any performance increase.
First, for TEMIS we translated its internal XELDA tagset to GENIA[10] (see the subsection above). The results in Table 3 above show a clear boost for TEMIS$_{GENIA}$ by 4.7 percentage points to 90.91% F-score. The FN error rate dropped by 186 to 663, and FP errors decreased by more than 50% (down to 240). Another run through our analysis of error types showed that for the TEMIS system the FN error rate for nominal elements dropped to 0%. This is not unexpected because various biomedicine-specific nouns are unknwon to the XELDA default tagger. In partiular, the noun *binding*, which accounted for almost half of the noun FN errors, was frequently mistagged by XELDA as a verbal progressive form and thus not recognized as part of a base NP. Furthermore, as shown in Table 5 the number of FP verbal errors drops from 161 to 28 using the GENIA tags. Thus, in parallel to the class of nouns wrongly tagged as verbs there is another class of verbs wrongly tagged as nouns by the TEMIS-internal XELDA tagger.
Second, our error analysis showed that NPs with internal parenthesized/bracketed elements peculiar to the biomedical domain are a major source of errors. Such elements can be recognized in a straightforward way by checking whether the opening parenthesis is directly preceded and the closing one directly followed by an NP (i.e., by a chunk I-tag). We thus examined in an exemplary way whether such a heuristic adaptation facilitating the recognition of these types of NPs would lead to any performance increase on the BOSS system. As can be seen in our results in Table 3, BOSS$_{Par}$ increased its performance by 1.6% F-score. In particular, this heuristic performed a boost on its accuracy value by 2.3%. This shows that more noun phrases peculiar to the biomedical domain are recognized correctly.

[9] These error classes, however, did not play any role as for FPs.

[10] A more far-reaching adaptation beyond the scope of this paper would be training the XELDA tagger on a domain-specific biomedical corpus (e.g., the GENIA corpus).

Discussion and Conclusions

We evaluated the performance of four different systems which perform noun phrase recognition for a biomedical text corpus (GENIA). Three of the systems are machine-learning-based systems which were trained on the PENN TREEBANK newspaper corpus. The F-score performance on the newspaper corpus is between 90.01% and 94.22% and drops down to between 85.9 and 89.15% for the biomedical corpus. Porting default chunkers to the life sciences domain, therefore, implies a substantial loss of performance. Furthermore, the drop of performance is system-dependent. The kernel-based support vector machine system, YAMCHA, performs best on both corpora but still looses 5% on the GENIA corpus. The performance loss for the TBL tool is even higher (over 6%). By contrast, the statistical chunking tool BOSS only looses 3.35%.
In a parallel evaluation step, we also examined a commercial general grammar-rule-based system (TEMIS) which employs hand-crafted grammar rules for fast system adaptations to different domains. It could be shown that the performance of the TEMIS system is comparable to the machine learning systems TBL and BOSS. Still, the settings of the two different methods (machine learning vs. hand coding) are not directly comparable. For the machine learning based systems the biomedical corpus is too small for using it as a training corpus. On the other hand, the standard TEMIS software uses a different tagger. Hence, a considerable amount of the TEMIS FN errors (approximately 21%, cf. Table 2) can be attributed to the fact that nouns in the GENIA corpus were not correctly recognized as such by the XELDA tagger. The error rate of false negatives could be cut in half through an adaptation of the TEMIS software using the GENIA tag set instead. With this adaptation, the performance is boosted to an F-score of 90.91%. This result shows the importance of accurate POS tagging for NP chunking. Despite of these differences, we stipulate that standard ML approaches (trained on a newspaper corpora) and a standard commercial domain-unspecific rule-based system (based on its own tagger) yield comparable performance results. This holds true unless support vector machines come into play. Though they grant a considerable performance boost, their application in large-scale systems is hard to envisage given their resource consumption requirements. This is a crucial counter-argument for their usability in the biomedical domain, which requires cheap computations on very large data sets.[11]

[11] This is particularly important considering the fact that we here deal with the rather basic pre-processing step of NP chunking, and have not even touched upon subsequent in-

For error analysis, false negative and false positive matches were compared. Although the individual systems' false negative and false positive hits do not directly correspond to their final performance, they do have an effect on it. For the TEMIS software, it is the high number of false negative hits, for the YAMCHA system, its very low number of false positive hits. With the help of the error rates we could identify common mistakes for the different systems. Twice as many false negative and positive errors are common between the ML-based methods as between all systems.

An analysis of error types showed that coordinated elements were identified as the most common error class. This comes as no surprise, since coordination was not only reported to be problematic for NP chunking tasks (see [14]), but also for more expressive higher-level formalisms such a as full-sentence parsing. Another prominent error class also reported in the literature is the recognition of verbal elements inside NPs. A more domain-specific error source came form NP-internal parentheses, which is a feature specific to the biomical domain. Certain error classes (nominal elements, cardinal numbers) do only appear in the rule-based TEMIS system and can be attributed to the fact that both its XELDA tagger and its XELDA dictionaries were set up as general and domain-unspecific as possible (whereas the ML systems also made use of the POS information given by the GENIA test set).

In follow-up experiments, however, we were able to boost the TEMIS software's performance by using the POS tags given in the GENIA test set. For this purpose, we manually translated the XELDA tagset to the GENIA/PTB one. A further domain-specific adaption concerned the recognition of noun phrases with internal parenthesized/bracketed elements. A straightforward heuristic solution for the BOSS system lead to noticeable performance increase.[12] Thus, our results are crucial with respect to the fast re-usability of such systems for different biomedical text mining tasks, such as named entity recognition or information extraction, especially in the light of insufficient in-domain (i.e., biomedical) training resources.

Acknowledgements:
Jena University is a member of the EU Network of Excellence *Semantic Mining* (Semantic Interoperability and Data Mining in Biomedicine – NoE 507505).

Address for Correspondence:
joachim.wermter@uni-jena.de

depth text processing and mining tasks, which tend to grow in their computational load.

[12] These results are also in line with previous studies ([18], [4]) which examined the portability of part-of-speech taggers to the biomedical domain.

References

[1] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. Chicago, IL: CSLI Publications, 2003.

[2] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[3] Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *ANLP 1988 – Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 136–143. Association for Computational Linguistics, 1988.

[4] Udo Hahn and Joachim Wermter. High-performance tagging on medical texts. In *COLING Geneva 2004 – Proceedings of the 20th International Conference on Computational Linguistics*, volume 2, pages 973–979. Geneva, Switzerland, August 23-27, 2004. Association for Computational Linguistics, 2004.

[5] Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *NAACL'01, Language Technologies 2001 – Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 192–199. Pittsburgh, PA, USA, June 2-7, 2001, 2001.

[6] José Carlos Clemente Litrán, Kenji Satou, and Kentaro Torisawa. Improving the identification of non-anaphoric it using support vector machines. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *JNLPBA – Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 58–61. Geneva, Switzerland, August 28-29, 2004., 2004.

[7] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The PENN TREEBANK. *Computational Linguistics*, 19(2):313–330, 1993.

[8] Marcia Muñoz, Vasin Punyakanok, Dan Roth, and Dav Zimak. A learning approach to shallow parsing. In Pascale Fung and Joe Zhou, editors, *EMNLP-VLC'99 – Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 169–178. College Park, Maryland, 1999. Association for Computational Linguistics, 1999.

[9] Meenakshi Narayanaswamy, K.E. Ravikumar, and Shanker E. Vijay. A biological named entity recognizer. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *PSB 2003 – Proceedings of the Pacific Symposium on Biocomputing 2003*, pages 427–438. Lihue, Hawaii, USA, January 3-7, 2003. Singapore: World Scientific Publishing, 2003.

[10] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In M. Marcus, editor, *HLT 2002 – Human Language Technology Conference. Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 82–86. San Diego, Cal., USA, March 24-27, 2002. San Francisco, CA: Morgan Kaufmann, 2002.

[11] Kyung-Mi Park, Seon-Ho Kim, , Ki-Joong Lee, Do-Gil Lee, and Hae-Chang Rim. Incorporating lexical knowledge into biomedical NE recognition. In *JNLPBA'04 – Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications at COLING'04*, pages 76–79, 2004.

[12] Hervé Poirier. The XeLDA framework, 1999. Presentation at Baslow workshop on Distributing and Accessing Linguistic Resources. [http://www.dcs.shef.ac.uk/~hamish/dalr/baslow/xelda.pdf].

[13] James Pustejovsky, Joseé Castano, Jason Zhang, M. Kotecki, and B. Cochran. Robust relational parsing over biomedical literature: Extracting inhibit relations. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauderdale, and Teri E. Klein, editors, *PSB 2002 – Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 362–373. Kauai, Hawaii, USA, January 3-7, 2002. Singapore: World Scientific Publishing, 2002.

[14] Lance Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL Workshop on Very Large Corpora*, pages 82–94. Cambridge, MA, USA, June 30, 1995. Association for Computational Linguistics, 1995.

[15] Jasmin Saric, Lars J. Jensen, Rossitza Ouzounova, Isabel Rojas, and Peer Bork. Extracting regulatory gene expressions expression networks from PubMed. In *ACL'04/EACL'04 – Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics & 10th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2004.

[16] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Association for Computational Linguistics, 2000.

[17] Vladimir N. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998.

[18] Joachim Wermter and Udo Hahn. Really, Is medical sublanguage that different? experimental counter-evidence from tagging medical and newspaper corpora. In Marius Fieschi, Enrico Coiera, and Yu-Chan Jack Li, editors, MEDINFO 2004 – *Proceedings of the 11th World Congress on Medical Informatics. Vol. 1*, number 107 in Studies in Health Technology and Informatics, pages 560–564. San Francisco, CA, USA, September 7-11, 2004. Amsterdam: IOS Press, 2004.

[19] Tateisi Yuka and Jun-ichi Tsujii. Part-of-speech annotation of biology research abstracts. In *LREC 2004 – Proceedings of the 4th International Conference on Language Resources and Evaluation. In Memory of Antonio Zampolli. Vol. 4*, pages 1267–1270. Lisbon, Portugal, 26-28 May 2004. Paris: European Language Resources Association (ELRA), 2004.