Rule Based Chunker for Croatian

Kristina Vučković*, Marko Tadić, Zdravko Dovedan***

* Department of Information Sciences
** Department of Linguistics
Faculty of Humanities and Social Sciences
University of Zagreb
Ivana Lučića 3, HR-10000 Zagreb, Croatia
{kvuckovi, marko.tadic, zdravko.dovedan}@ffzg.hr

Abstract

In this paper we discuss a rule-based approach to chunking sentences in Croatian, implemented using local regular grammars within the NooJ development environment. We describe the rules and their implementation by regular grammars and at the same time show that in NooJ environment it is extremely easy to fine tune their different sub-rules. Since Croatian has strong morphosyntactic features that are shared between most or all elements of a chunk, the rules are built by taking these features into account and strongly relying on them. For the evaluation of our chunker we used a extracted set of manually annotated sentences from 100 kw MSD/tagged and disambiguated Croatian corpus. Our chunker performed the best on VP-chunks (F: 97.01), while NP-chunks (F: 92.31) and PP-chunks (F: 83.08) were of lower quality. The results are comparable to chunker performance of CoNLL-2000 shared task of chunking.

1. Introduction

Chunking is today considered to be the preprocessing stage that may facilitate the full parsing of sentences of a certain language. This task has already been proven using different methods: rule-based (Ramshaw & Marcus 1995, Abney 1991, Villain & Day 2000, Johansson 2000, Déjean 2000), memory-based (Veenstra & Van Den Bosch 2000), statistical (Pla et al. 2000, Osborne 2000, Koeling 2000, Zhou et al. 2000) and combined systems (Tjong Kim Sang 2000, Van Halteren 2000, Kudoh & Matsumoto 2000). However, when developing a chunker for a new language, statistical and machine learning methods require an already preprocessed corpus which is not always available. In our case, such a corpus for Croatian does not exist at all so we had to opt for the rule-based approach that would, in the prospect, end up with a data-set usable for future testing of similar systems for Croatian.

In this paper we describe the attempt to build the first rule-based chunker for Croatian and evaluate its performance. Section 2 of the paper describes the resources and tools used for producing and testing our chunker, while section 3 describes various rules for different types of chunks. Section 4 then defines the evaluation framework that would finally provide us with results. The discussion is presented in section 5 and conclusions along with future research plans in section 6.

2. Language resources and tools

In this section, we give detailed insight on tools and resources used in the experiment, along with other facts of interest – namely, basic characteristics of annotated corpus and NooJ development environment.

2.1. The annotated corpus

The *Croatia Weekly* 100 Kw newspaper corpus (the CW100 corpus further in the text) consists of articles extracted from seven issues of the *Croatia Weekly* newspaper, which has been published from 1998 to 2000 by the Croatian Institute for Information and Culture (HIKZ). This 100 Kw corpus is a part of Croatian side of the Croatian-English Parallel Corpus (CW corpus) described in detail in (Tadić, 2000). The CW100 corpus was pre-tagged using the MULTEXT-East version 3 (MTE v3) morphosyntactic specifications on the top of XCES corpus encoding standard (Erjavec, 2004):

```
<w lemma="ipak" ana="Rn">ipak</w>
<w lemma="početi" ana="Vmps-sfa">počela</w>
<w lemma="Hrvatska" ana="Npfsd">Hrvatskoj</w>
```

Figure 1: Excerpt from the XML encoding of CW100 corpus

The whole CW corpus was in fact built in two separate processing stages, as described in (Tadić, 2000): firstly, the raw text data was automatically converted into XML format and afterwards tokenized in order to be semi-automatically tagged using the full MTE v3 MSD tagset by matching the CW100 corpus and the Croatian Morphological Lexicon at unigram level via the Croatian Lemmatization Server (Tadić, 2006 and at http://hml.ffzg.hr).

Croatian language in general implements 12 out of 14 different PoS categories defined in the MTE v3 specification: Adjective (A), Conjunction (C), Interjection (I), Numeral (M), Noun (N), Pronoun (P), Particle (Q), Adverb (R), Adposition (S), Verb (V), Residual (X) and Abbreviation (Y). However, 11 of them actually do appear in the CW100 corpus (Residual missing out), the fact once again suggesting that it's a rather small resource to operate with, both in quantity and quality, especially when compared to resources available for some other highly inflectional Central and Eastern European languages.

Although we could have used other types of annotating scheme featuring stand-off annotation (e.g. Buitelaar et al, 2003), we decided against it since the NooJ environment, that we selected for the sake of simplicity of developing and adapting local grammars, does not handle stand-off type of annotation well.

2.2. NooJ development environment

In this project we used NooJ (Silberztein, 2006 and at http://www.nooj4nlp.net) as a tool for natural language

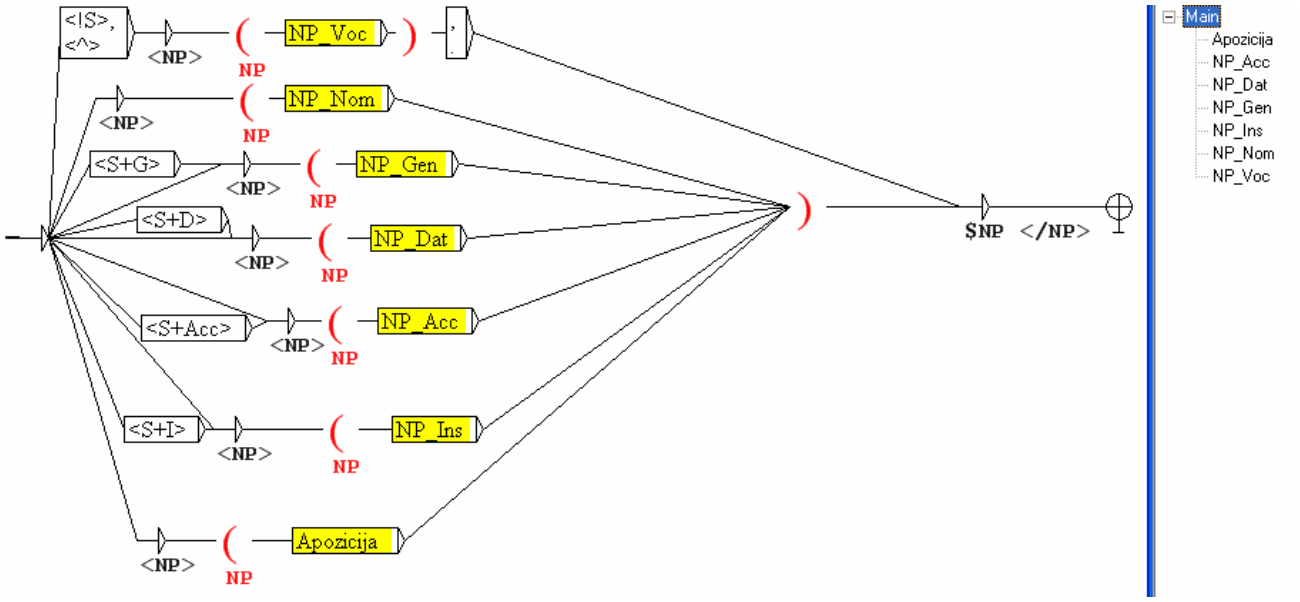

Figure 2. NP-chunks graph

processing using formalized descriptions of inflectional and derivational morphology, lexicon, regular grammars, and CF grammars. NooJ uses electronic lexicons and grammars represented by organized sets of graphs. It integrates morphology and syntax thus enabling morphological operations inside syntactic grammars. Using morphological and syntactic formal descriptions in NooJ, it is possible to add or remove additional annotations on different linguistic levels.

NooJ uses:

- FST for building graphs that recognize described strings of text;
- FSA for locating POS and/or MSDs;
- RTN grammars that contain more FST and/or FSA graphs;
- ERTN that contain variables for storing parts of matching sequences used to perform some operations before producing the output;
- CFG for morphological and syntactic grammars;
- regular expressions for simple queries.

In NooJ, grammars could be defined in several ways: writing regular expressions or using graphical interface for drawing the grammar graphs. System then interprets the graphical representation and converts it into an automaton. Cascading grammars and invoking grammars from within each other are completely supported thus leading to a powerful and yet user-friendly development environment. The way in which NooJ's enhanced grammar uses internal variables for storing parts of the recognized sequences in order to use them for constraining the output, greatly increases the functionality of this tool. NooJ not only lets you use derivational and inflectional morphology engine for processing variables' content but also retrieves and extracts values of a variable's property associated with it's content.

For these reasons, we have chosen NooJ as our development platform for building local grammars that shoud function as a chunker for Croatian.

3. Description of rules

Three different types of chunks are covered by our rules: NP-chunks, PP-chunks and VP-chunks. Each of these types show their particular features which we had to consider while developing three sets of rules. They all share the same preconditions for input and output file processing, as described in the previous section. We now describe these rules.

3.1. NP-cunks

The first types of chunks are NP-chuks which could be defined as presented on Figure 2.

In this graph we have combined seven graphs, one for the apposition (see Figure 3) and six for six cases including Nominative, Genitive, Dative (see Figure 4), Accusative, Vocative and Instrumental. Notice that Locative case does not have a graph inside the NP chunk since it can only come with a preposition so it is only included into the PP chunk.

Also an additional information on the type of <np> chunk or case of its head is added in tag attribute.

This grammar recognizes the following examples:

```
<NP type="Acc">Američke ratne mornarice
<NP type="Acc">Apsolutni trenerski record
<NP type="Acc">austrijski medijski concern
<NP type="Acc">četiri posebne nagrade
<NP type="Acc">druge nadležne državne institucije
<NP type="Appo">violinist Anđelko Krpan
<NP type="Appo">Vijeću sigurnosti
<NP type="Appo">vratar Dragn Jerković
<NP type="Appo">web stranicu
<NP type="Appo">zamjenik premijera
<NP type="Appo">zemljama članicama
<NP type="Dat">afričkim zemljama
```

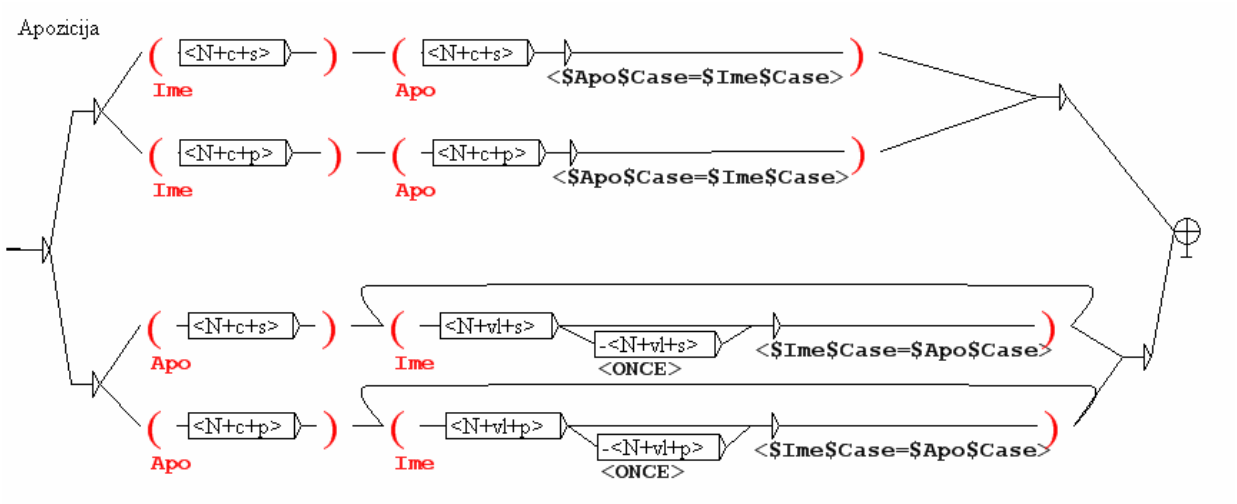

Figure 3. Graphical rendering of App-NP-chunk local grammar

```
<NP type="Dat">aktualnoj gospodarskoj situaciji
<NP type="Dat">četvrtoj Uskršnjoj regati
<NP type="Dat">demagoškim predizbornim obećanjima
<NP type="Dat">drugim većim hrvatskim gradovima
<NP type="Gen">Arheološkog muzeja
<NP type="Gen">Baletne trupe
<NP type="Gen">balkanske regije
<NP type="Gen">bogatog umjetničkog opusa
<NP type="Ins">aktualnim događanjima
<NP type="Ins">američkom državnom tajnicom
<NP type="Ins">atraktivnom turističkom ponudom
<NP type="Ins">Sistematskim liječničkim pregledom
<NP type="Ins">svim diplomatskim predstavnicima
<NP type="Ins">svojim dugim monolozima
<NP type="Nom">aktivno sudjelovanje
<NP type="Nom">apsolutni trenerski record
<NP type="Nom">besplatan vojnički grah
<NP type="Nom">bivša vladajuća stranka
<NP type="Nom">blažena Elizateba Mađarska
<NP type="Nom">čuvena šibenska katedrala
<NP type="Nom">davno nestalo Panonsko more
<NP type="Nom">DC-ov politički program
```

Rules for apposition NP-chunk are defined in Figure 3. This grammar checks whether two nouns agree in case and number but not necessary in gender since this feature is irrelevant for NP of apposition type. This grammar recognizes the following examples:

```
<NP type="Appo">milijun kuna
<NP type="Appo"predsjednika Vlade Račana
<NP type="Appo"ispraćaj predsjednika
<NP type="Appo"Potpredsjednik Vlade Goran Granić
<NP type="Appo"direktora Galića
<NP type="Appo"Vijeće HRT-a
```

Two NP-chunks that share the same case and are connected with 'i' or 'ili' belong to the same NP.

This grammar recognizes the following examples:

```
<NP type="Nom">mnogo veće i siromašnije zemlje
<NP type="Dat">civilizacijskoj i europskoj obnovi
<NP type="Nom">europski ciljevi i zahtjevi
<NP type="Nom">kontinuitet i obnova
<NP type="Nom">neka ministarstva i djelatnosti
```

3.2. PP-chunks

The second types of chunks are PP-chunks and they could be defined as a combination of a preposition and an already recognized NP-chunk as illustrated in Figure 6.

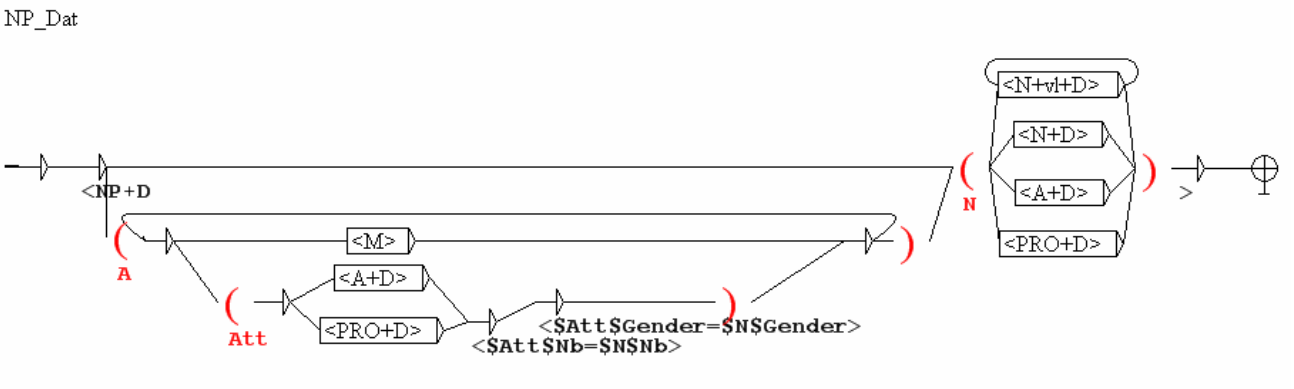

Figure 4. Graphical rendering of Dative-NP-chunk local grammar

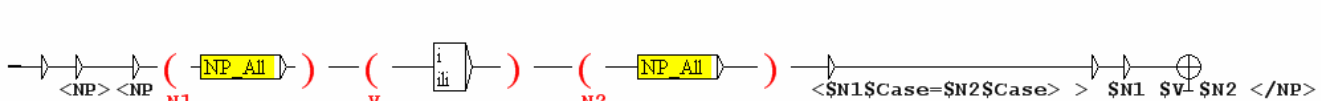

Figure 5. Graphical rendering of coordination

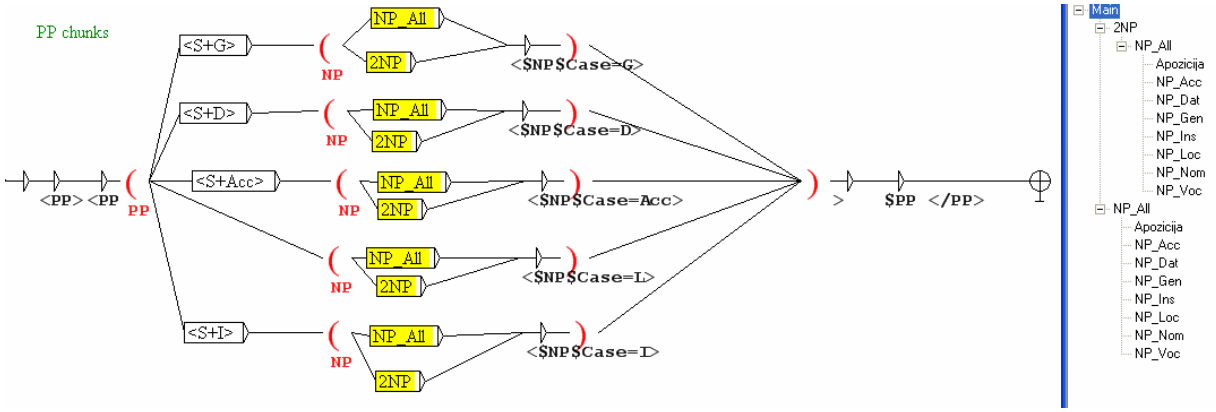

Figure 6. Graphical rendering of PP-chunk local grammar

The PP-chunks local grammar takes into account the agreement in case between the nominal part of NP and preposition that opens a place for this case. The nominal part of the PP-chunk may either be one NP in Genitive, Dative, Accusative, Locative or Instrumental or two NPs that share the same case. Since Nominative and Vocative cases come without the preposition, they are not included into this grammar.

This grammar recognizes the following examples:

```
<PP>iz naše prve banke
<PP>između šarenih nogu
<PP>za dodatno europsko prosjačenje
<PP>u tim okvirima
<PP>o nemoćnoj paralizi
<PP>u prvom redu
<PP>osim Predsjednika Republike
<PP>za obnovu i razvoj
```

3.3. VP-cunks

The third types of chunks are VP-chunks that, in our case, cover the compound verbal tenses and moods. They are defined as a combination of a participle and auxiliary which agree in number, person and gender.

Since the gender is inflectionally marked in the participle the cases of agreement between chunk boundaries (i.e. subject NP and predicate VP) are left to higher syntactic level of processing.

This grammar is presented by Figure 7. and it recognizes the following examples:

```
<VP>bi trebali biti
<VP>će se protiviti
<VP>definirat će
<VP>ne može govoriti
<VP>ne mogu čekati
<VP>nije htio pozdraviti
<VP>nije otvoreno
<VP>bude išla
<VP>bude mogla financirati
<VP>bi mogao pokrenuti
<VP>bismo htjeli biti
<VP>će nastojati podići
<VP>je morala djelovati
<VP>ne bi mogla slijediti
<VP>ne bismo zanemarili
<VP>se ne smiju događati
<VP>se ne vraća
```

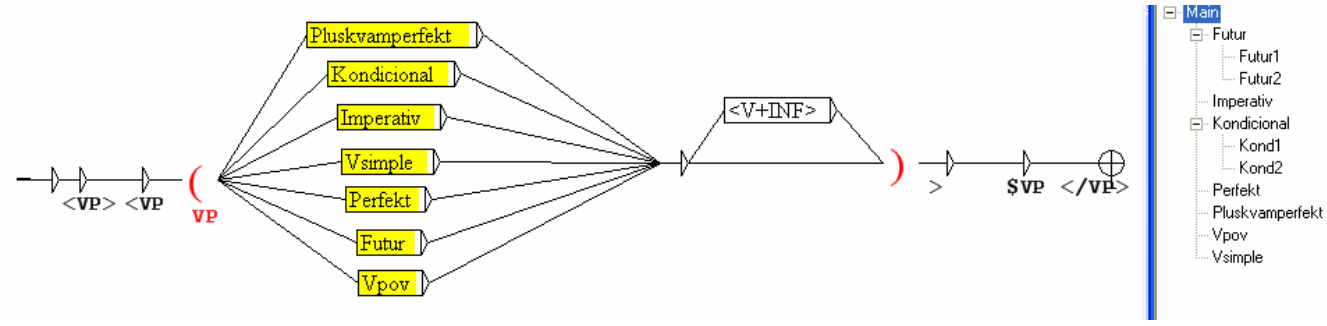

Figure 7. Graphical rendering of VP-chunk local grammar

Given Figures 2 to 7, it is obvious that our chunker may have problems with overlapping chunks. We tried to avoid this problem by introducing the longest-match priority. The longest possible match, defined by local grammar, recognizes the first amongst all possible solutions and that part of text is excluded from further processing.

4. Evaluation methods

Since there is no Golden Standard available for evaluation of chunkers in Croatian, we had to build our own set of sentences for that purpose.

For this first evaluation of our chunker output we used a subset of only 137 sentences selected from the CW100 corpus comprising 371 clauses. This test set was manually processed by inserting XML tags marking the chunk boundaries and types (such as <NP>, <VP>, <PP>). The basic statistics show that in this test set of sentences <NP> appeared 840 times, <PP> 342 times and <VP> 345 times giving an overall number of 1527 chunks.

The local grammars built for recognizing chunk boundaries and marking their types were applied to the same set of sentences with tags stripped off. The output was then compared to the original set of sentences with chunks annotated.

5. Results and Discussion

The first results were surprising because the number of recognized chunks was much higher than number of chunks manually annotated. The first explanation was that the grammars were probably not defined well. After careful checking of grammars, we found that the reason for such a large number of recognized chunks was in the fact that NooJ environment offered all possible inter-pretations for every homographical word. The reason for this is that NooJ deals with tokens at the unigram level and, if otherwise not described by grammar, does not take into account its co-text. The cascading order of application of grammars led to emmbeding of chunk tags within the tags for chunks of the same type i.e. it led to a recursion which should not be allowed by the very definition of a chunk.

	Test set	1st NooJ output
NP-chunks	1150	1837
PP-chunks	348	248
VP-chunks	447	416
Total	1945	2501

Table 1: The first application of grammars

In order to overcome this faulty results, we had to delete all tags that were embeded within the tags of the same type. This led to transformation of <NP type="Acc"> <NP type="Gen"> <NP type="Nom">države</NP> </NP> </NP> into <NP type="Gen">države</NP>. There were also cases where NooJ offered embedding of different types of chunks (e.g. <PP>u<NP type="Loc"> <VP>vezi</VP></NP></PP>) but they were filtered out since their distribution was not allowed (e.g. no VP-chunk is allowed inside NP-chunk).

After clearing out this problems the second application of grammars yielded the final results (Table 2).

	Test set	2nd NooJ output
NP-chunks	1150	1099
PP-chunks	348	249
VP-chunks	447	456
Total	1945	1804

Table 2: The second application of grammars

At the first sight it may look as a surprise that the number of VP-chunks NooJ detected is higher than the number of VP-chunks in the test set. This is explained by the overgeneration of VP-chunks due to the high frequency of certain conjunctions and nouns that are homographs with verbs as typical parts of VP-chunk. For the same reason some NP-chunks have also been wrongly detected and all of these cases have been treated as wrong chunk tag assignment.

After careful examination of such problematic cases, we also calculated the standard measures: precision, recall and F-measure as they are defined in (Tjong Kim Sang and Bucholz, 2000).

	Precision	Recall	F-measure
NP-chunks	94.50	90.26	92.31
PP-chunks	99.60	71.26	83.08
VP-chunks	96.05	97.99	97.01
Total	96.72	86.50	90.80

Table 3: The precision, recall and F-measure

From the Table 3 it can be seen that the F-measure for VP-chunks performed the best while NP-chunks and PP-chunks were of lower recall and hence also lower F-measure.

6. Conclusions and future work

In this paper we have presented the first attempt to develop a chunker for Croatian. We opted for rule-based approach since there are no corpora annotated for chunks in Croatian and no machine-learning or statistical-based methods could be applied. Also, since Croatian is highly inflectionally rich language, many correspondences between specific values of MSDs could be used for detecting chunk boundaries and chunk types. This was achieved by applying local (regular) grammars to strings of tokens and their MSDs.

There are many directions for further enhancement of the results presented in this paper but we see one as the most important. Additional preprocessing could help us avoid the homography problem that was generated by cascading application of grammars. This preprocessing could be achieved by applying MSD-tagger to text that produces disambiguated output and therefore avoid the limitation of NooJ system that was observed since it treats the tokens at unigram level if not otherwise defined by local grammar. This can lead to multiple interpretations. The fully MSD-tagged input would constrain this problem.

Obtained results peaked at around 90% correctly assigned chunk boundaries and types leading us to the conclusion that rule-based paradigm would be a reasonable choice for chunking of larger Croatian corpora in the future.

7. Acknowledgments

This work has completed within the projects supported by the Ministry of Science, Education and Sports, Republic of Croatia, under the grants 130-1300646-1776, 130-1300646-0645 and 036-1300646-1986.

8. References

Abney, S. (1991). Parsing by chunks. In *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers, Boston, 1991, pp. 257-278.

Buitelaar, P., Declerck, T., Sacaleanu, B., Vintar, Š., Raileanu, D., Crispi, C. (2003). A Multi-Layered, XML-Based Approach to the Integration of Linguistic and Semantic Annotations. In *Proceedings of the EACL2003 Conference, Workshop on NLP and XML Language Technology and the Semantic Web*, EACL, Budapest, 2003, pp. 9-16.

Déjean, H. (2000). Learning syntactic structures with xml. In *Proceedings o/ CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp 127-132.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. ELRA, Paris-Lisbon 2004, pp. 1535-1538.

Van Halteren, H. (2000). Chunking with wpdv models. In *Proceedings o/ CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 154-156.

Johansson, C. (2000). A context sensitive maximum likelihood approach to chunking. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 136-138.

Koeling, R. (2000). Chunking with maximum entropy models. In *Proceedings of CoNLL-2000 and LLL-2000.* Lisbon, Portugal 2000, pp. 139-141.

Kudoh, T., Matsumoto, Y. (2000). Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 142-144.

Osborne, M. (2000). Shallow parsing as part-of speech tagging. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 145-147.

Pla, F., Molina, A., Prieto, N. (2000). Improving chunking by means of lexical-contextual information in statistical language models. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 148-150.

Ramshaw, L. A., Marcus, M. P. (1995). Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpor*a. ACL, pp. 82-94.

Silberztein, M. (2006). NooJ' Manual. . http://www.nooj4nlp.net/NooJ%20Manual.pdf.

Tjong Kim Sang, E. F. (2000). Text chunking by system combination. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 151-153.

Tjong Kim Sang, E. F., Buchholz, S. (2000) Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000, pp. 127-132.

Tadić, M. (2000). Building the Croatian-English Parallel Corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation*. ELRA, Paris-Athens 2000, pp. 523-530.

Tadić, M. (2003). *Jezične tehnologije i hrvatski jezik*. Exlibris, Zagreb 2003.

Tadić, M. (2006). The Croatian Lemmatization Server. In *Proceedings of the FASSBL5 Conference*. Sofia 2006, Bulgarian Academy of Sciences, pp. 140-146.

Veenstra, J., Bosch, A. (2000). Single-classifier memory-based phrase chunking. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 2000, pp. 157-159.

Vilain, M., Day, D. (2000). Phrase parsing with rule sequence processors: an application to the shared CoNLL task. In *Proceedings of the Conference on Natural Language Learning of CoNLL-2000*, Lisbon, Portugal, 2000, pp. 160-162.

Zhou, G., Su, J., Tey, T. (2000). Hybrid text chunking. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal 2000, pp. 163-165.