A RULE BASED NOUN PHRASE CHUNKER FOR TURKISH

Kübra Adalı
Dep. Of Computer Eng.
Istanbul Technical University
Istanbul, Turkey
kubraadali@itu.edu.tr;

Yrd. Doç.Dr.A. Cüneyd Tantuğ
Dep. Of Computer Eng.
Istanbul Technical University
Istanbul, Turkey
tantug@itu.edu.tr;

ABSTRACT

In this paper, we presented a noun phrase chunker for Turkish as an agglutinative language. For finding noun phrases in Turkish sentences, we propose a rule based model which includes preprocessing part and a unit that applies the local grammatical rules to the output of the dependency parser. To the best of our knowledge, our model gives the first results on noun phrase chunking of Turkish sentences that is expected to find not only the basic noun phrase sentences but also the complex noun phrases including the relative clauses. We believe that on that sense, our model will be a good reference for future studies in this domain. We tested our model both on manually annotated data and the output version of the dependency parser. Our model gives the results with annotated data for full match 66.15\% and the partial match 76.79\% (for F1 results). Using output of the dependency parser, the results are 47.91\% and 60.75\% for F1 results accordingly (for F1 results).

1 Introduction

As the conclusion of the wide usage of the internet and social media and the data on the web which is getting bigger day by day, the applications that are used to summarize huge amount of data such as information retrieval, text summarization, text categorization, information extraction become more popular and considerable for analysis of the web data. These applications needs meaningful groups of words so that they can analyze huge amounts of data without wasting effort for unnecessary details. Chunking is accepted as shallow parsing that parses a sentence into meaningful word groups (Ramshaw and Marcus, 1995). Additionally, it is used as preprocessing stage of dependency parsing and does not deal with as many details as full parsers. For this reason, chunkers become an important part of the applications that are motivated to summarize huge amount of data because they gives the word groups that gives the considerable information about the meaning of the sentence or gives the limited search spaces for a specific word or word group which represents a topic or an issue.

For machine translation, chunking of sentences gives the chance of making alignments in smaller search space in a sentence, increases the percentage of the word groups for alignment of the words from the source language to target language by finding phrases. For that reason, chunkers are used as a preprocessing stage in a machine translation system. In named entity recognition, especially noun phrase chunkers can be a very useful and substantial stage for named entity recognition.

After noun phrase chunker find the noun phrases in a sentence, it would be much more easier to find the named entities in noun phrases rather than in the whole sentence and in most of cases, the noun phrase becomes directly a named entity itself. (Sassano and Utsuro, 2000)

Although chunking is basicly accepted as dividing sentences into meaningful groups of words, the types of chunks are determining factor for the purpose of use of the chunker.

Noun phrase chunkers are used very commonly because they find chunks which contain words that organize around a noun and nouns are main types of words that gives the information about the topic or the meaning of the sentence. Noun phrases are defined as the basic word groups that qualifies a main word whose type is a noun in the literature, but we enlarge the area of the definiton of noun phrases because the scope of description of the noun phrases needs to be include more complex structures of word groups according to Turkish grammar. To give an example for Turkish, "büyük kırmızı bir elma" (a big red apple) is a standard basic noun phrase, but "ağaçtan düşen elma" (the apple which fell from the tree) is accepted by our study as a noun phrase although it is a relative clause and can not be a basic noun phrase.

In this paper, we propose a model that finds noun phrases in a Turkish sentence. Our model uses two stages: first part that is used preprocessing unit that gives the relations from the dependency parser and in the second stage, grammatical rules that uses the relations which are given by the Turkish dependency parser.

Although it is a rule based model and has language dependence, it is the first study for Turkish according to the scope of definition or complexity of noun phrases that are expected to find by our model. The first results of noun phrase chunker for Turkish 47.91% (for full match) and 60.75% (for partial match) as F1 results. We hope that our model and these results will be a reference for next studies for more complex noun phrase chunkers.

The remaining of the paper is structured as follows: Section 2 presents the related work, Section 3 explains the definition of a noun phrase in literature and the definition of noun phrases and the scope of the noun phrases in Turkish. Section 4 discusses our proposed model and Section 5 explains our experiments and results. The conclusion is given in Section.

2 Related Work

Noun phrase chunking has been done for many different types of languages by using many different methods. To start with English, Church (1988) used a stochastic model for noun phrase chunking. To the best of our knowledge, this is the first study about noun phrase chunking for English. Chen et al. (1993) deals with the noun phrase chunking problem on English by applying bigram language model as a statistical method. Cardie and Pierce (Cardie and Pierce, 1998) works on a system based on machine learning of the patterns that are composed of specific and common used sequences of part of speech tags on English. This is a hybrid model that includes rule based and statistical approach.

Another study about fusional languages is on German which Kermes et al. (2002) has built a rule based system that is composed of rules according to German grammatical structure. Another noun phrase chunker system for German is done by Atterer and Schlangen (Atterer and Schlangen,2009). This study uses an incremental structure.

On the other hand, Singh et al. (2005) reports the results of a chunker system for not only noun phrases but also the other types of phrases. The system works on Hindu by using HMMs. Vuckovic et al. (2008) uses a rule based model for noun phrase chunking of Croatian sentences based on morphological and syntactic structure of Croatian as a Slavic language. This study also works on the other types of phrases addditional to noun phrases. Dhanalakshmi et al. (2009) is an important study for Turkish because it possess noun phrase chunking problem on Tamil which is agglutinative language like Turkish. Three machine learning techniques that are CRFs, SVMs and memory based learning are used and compared. The winner method is CRFs for 9 different types of phrases including noun phrases. Another chunking system which uses a machine learning technique is the system of Radziszewski and Piasecki (Radziszewski and Piasecki, 2010). They uses decision trees to deal with the noun phrase chunking problem on Polish which is a Slavic language. To train for the decision trees some tree patterns are used as features. Asahara et al. (2003) uses support vector machines for word segmentation using the groups of characters on Chinese. It can be accepted as a chunking study because each character can represent a word in Chinese.

The work done so far for Turkish are the work of Kutlu (2010) for noun phrase chunking and Akın and El-Kahlout (El-Kahlout and Akin, 2013) for chunking of constituents of Turkish sentences. The noun phrase scope of the study of Kutlu (2010) is different from the scope of our study (see Section 3 for further discussions), the application of the work is not accessible for public use, and Kutlu (2010) didn't give the details of the rules to apply the similar version of this. For the second work, also (El-Kahlout and Akin, 2013) only finds the constituents in a sentence and can not say if a chunk is a noun phrase or not. For these reasons, it is impossible to give the comparison between two later systems and our proposed model and the first study that we could find and which finds not only basic noun phrases but also complex noun phrases of a Turkish sentence is our study.

3 Noun Phrase Chunking

This section gives the details about the noun phrases in Turkish as an agglutinative language, the scope and complexity of the noun phrases that our work tries to find by supporting with examples.

3.1 Turkish

Noun phrases in Turkish has a main rule that the head word must be a noun. This rule is valid for all types of the noun phrases.

As an agglutinative language, the words in Turkish has the possibility of turn into an infinite number of different words by using the iterative sequences of inflectional and derivational suffixes. So as for all types of words, theoretically, it is a possibility that a noun is derivated from any types of words and this makes impossible to find the surface of the noun is derived from a noun or a verb. Morphological analyzer can not be enough in some cases to find this detail to decide the word can be a head of a noun phrase or not.

In Turkish sentences, the words that have a relation can have a long distance so this causes noun phrases become very long or a disambiguation problem about which noun is the head word of the noun phrase, which word defines which head word or belongs to which noun phrase. Additional to them, it is an another disambiguation problem to detect the phrases which seems to be a noun phrase but it composes a verb phrase.

3.2 Noun Phrases in Turkish

In Turkish, the noun phrases can be categorized as four main groups that are:
sanctus est Lorem ipsum dolor sit amet

1. Some examples can be as "Fatma'nın kitabı" (Fatma's book), "Kapının anahtarı" (key of the door), "Araba tekerleği" (wheel of car) etc. In Turkish, there are four different subtypes of this type of noun phrases according to suffixes that the head word and the assisting words of the noun phrase have.(Hengirmen, 2002)
2. Basic noun phrases in which the head word is connected by assisting words with the modification relation. To give some examples, "beyaz elbise" (white dress),"kırmızı gül" (red rose),"küçük ve zeki bir çocuk" (a small and clever child),"eski, şirin, küçük, boş, ve mavi bir ev" (a blue, old, sweet, small, empty house) etc.

3. The hybrid of the last two types of noun phrases. In hybrid type, only head part or only assisting part of the phrase can be a word or another type of phrase that is latter first or second type, or both of them can be a word or another type of phrase that is latter first or second type. Both two parts can be in three states (a word, a first type phrase and a second type phrase) so there are totally 9 subtypes of the hybrid type.
   The examples that belongs to these subtypes are: "Sema'nın güzel kızı" (the beautiful daughter of Sema), "Küçük çocuğun bisikleti" (the bicycle of small child), "Kırmızı başlıklı kızın yaşlı anneannesi" (the old grandmother of the girl with red hat), "Büyük kapılı evin küçük yeşil bakımlı bahçesinin ağaçları" (the trees of the small, green, well-kept garden of the house with big door)
4. Noun phrases whose head word is related with relative clauses. In this type, the head word must be a noun and if it is a derived noun, it shouldn't be derived from a verb. The assisting part of the noun phrase is represented by a relative clause. This type is most complex and long type of the noun phrases in Turkish. The examples that are from real texts of news in Turkish are as follows:
   The examples that are from real texts of news in Turkish are as follows:
   - "Başbakan Kostas Simitis başkanlığında düzenlenen güvenlik toplantısı" (a security meeting that is headed by Prime Minister Costas Simitis)
   - "Dünyanın dört bir yanından televizyon izleyicilerinin tanık olduğu muhteşem gösteri" (a spectacular performance which is witnessed by television viewers around the world.)
   - "1996 yılında Atlanta' da altın madalya alan Yunan rüzgar sörfcüsü Nikos Kaklamanakis" (windsurfer Nikos Kaklamanakis who is a gold medallist in Atlanta in 1996.)

As can be observed from the examples, the assisting part of the noun phrase can be so long also a phrase or a big sentence so it causes the growing of the complexity of the noun phrase chunking. The main rule that the head word must be a noun is valid for the multi word expressions whose last word is a verb can not accepted as a noun phrase.

4 Proposed Model

The work which is done so far in order to chunk noun phrases is based on two main different approaches that are mainly rule based systems and the systems in which machine learning methods such as CRFs, SVM, TBL etc. are used.(Section 2). Our model belongs to first group of approaches. Turkish is an agglutinative language. with rich, highly inflectional and derivational morphology, and complex relations between words in a sentence. Additional to the basic types of noun phrases, the scope of our study contains more complex and longer noun phrases as seen on the examples in the previous section. For these reasons, a rule based system needs to get detailed morphological analysis and relations between words in order to develop and apply the rules that finds accurate bounds of noun phrases in a sentence.

We propose a two stage model (Figure 1)

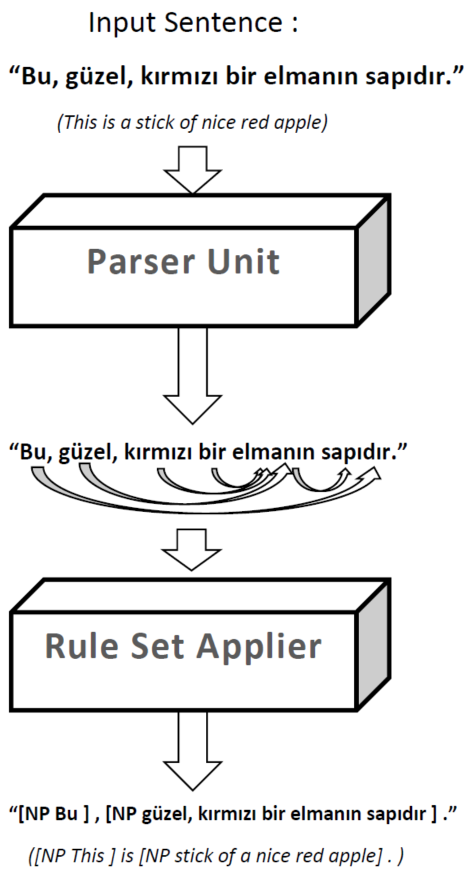

Figure-1. Proposed Model

which has basically two components.

1. A parser unit
2. A rule set applier

Our model uses four different labels for chunking noun phrases:
B : means that the word is the first word of the noun phrase.
I : means that the word is one of intermediate words of the noun phrase.
H : means that the word is head word of the noun phrase.
O : means that the word is not in a noun phrase.

4.1 Parser Unit

This stage is the part that we get the dependency relations of words. We employed dependency parser of Eryiğit et al. (2008) in order to obtain the relations to apply rules. The dependency parser needs morphological analysis of words with tags at the beginning and end of the sentence in a particular format so that this part contains two main substages:

1. Preprocessing
2. Parsing

In the preprocessing stage, there are two steps for a sentence as an input to the system. In the first step, we use the modified version (xxx, 2014) of two-level morphological analyzer (Şahin et al.,2013). After this step, we use morphological disambiguator by (xxx, 2014) and make the sentence in the available format for the dependency parser secondly. This preprocessing stage not only prepares input data for dependency parser but also serves the morphological information directly for rules. In the parsing stage, we use dependency parser of Eryiğit et al. (2008). It takes the input data which is prepared in the first stage, and gives the relation types and relation numbers in the same format.

It is shown by the Table 1 that an example sentence "Bu, güzel, kırmızı bir elmanın sapıdır." which has 9 tokens tokenized, morphologically analyzed and disambiguated in the preprocessing stage and parsed in the second stage. At the end of the parse unit, the output that we use as the parse information of the sentence is shown on the Table-1. As seen on the Table-1 each word is represented by a word number and each relation number states the word has the dependency relation to the word which is the owner of the row. For example, the first word "bu" is related to the word "sapıdır" whose number is 8 and the relation type is subject.

Word Num.	Surf. Form	Lemma Form	Coarse P.Tag	Fine P.Tag	Feats	Rel. Num.	Rel. Type
1	Bu	bu	Pron	DemonsP	A3sg‖*Pnon*‖*Nom*	8	SUBJECT
2	,	,	Punc	Punc	_	3	notconnected
3	güzel	güzel	Adj	Adj	_	7	MODIFIER
4	,	,	Punc	Punc	_	5	notconnected
5	kırmızı	kırmızı	Adj	Adj	_	7	MODIFIER
6	bir	bir	Adj	Num	_	7	DETERMINER
7	elmanın	elma	Noun	Noun	A3sg‖*Pnon*‖*Gen*	8	POSSESOR
8	sapıdır	sap	Verb	Zero		9	SENTENCE
9	.	.	Punc	Punc	Pres‖*A3sg*‖*Cop*	0	ROOT

Table-1. The output from the dependency parser of the sentence that Figure 1 shows.

```
Procedure:
Initialize chunkbin
Initialize chunk
for i = 1 to Number_of_Words_in_Sentence
    Set currentWord = i. Word
    Set rel = relation type of i. Word
    Set nRel = next relation of i. Word
    Set nWord = next related word of i. Word
    Set chunk = null
    if  i. word doesn't has any incoming relations then
         add i to chunk
        while (nRel is not (sentence or root)) or (nWord is noun)
           if nRel is (modifier or determiner or possessor)
               add word number of relatedWord to chunk
           else if rel is (modifier or determiner or possessor)
               add word number of relatedWord to chunk
           else if rel is not (subject or object)
                if relatedWord is noun
                   add word number of relatedWord to chunk
            end
            Set currentWord = relatedWord
            Set rel = relation type of currentWord
            Set nrelation = next relation of currentWord
            Set nWord = next related word of currentWord
        end while
        add chunk to chunkbin
    end
end
for each (chunk as c in chunkbin)
   if c is a subset of any chunk in chunkbin then
        Remove c from chunkbin
   end
end for each
```

Figure-2. The pseudo code for the algorithm of rule based system.

There are totally 26 types of relations that is defined for the dependency parser and the number of possible relation word for each word is equal to the number of words in a sentence. So the complexity for the rule set for each word in a sentence is 26xn (n is the number of words in a sentence.).

4.2 Set Of Rules

The morphological analysis and dependency relations of sentence that is taken from the Parse Unit are transferred to the Set of Rules stage. This stage which is based on a chunking algorithm gives the noun phrases annotated in the sentence as output. The pseudo code of the algorithm is given as in the Figure 2.

The algorithm visits each word in a sentence and firstly checks if the word has an incoming relation from any other words in the sentence in order to decide to start a chunk. If the word has an incoming link, the algorithm passes to the next word and start the rest of the procedure again for the next word, if the word does not, the algorithm starts to a new chunk and puts the word into the chunk. After that, the algorithm checks the related word recursively if it should be put into the chunk until the related word reaches the sentence relation. It puts the related word to the chunk and goes on at three states:

1. the next relation type is one of modifier, determiner, or possessor: this is for the noun phrases that contains relative clauses.
2. the relation type is one of modifier, determiner, or possessor.
3. the relation type is not one of (subject or object) and related word is noun.

After the algorithm reaches the sentence relation for each chunk, the chunk is put to the chunk bin. When the walk of the algorithm on the words of the sentence completed, the chunks that is subset of an another chunk in the sentence are filtered from chunk bin as the last step of the algorithm. In the example at Figure 1, the relations of the sentence that can be seen on the Table-1 is given to the algorithm and gets 2 noun phrases that are “Bu” (This) and “güzel, kırmızı bir elmanın sapı” (stick of a nice, red apple).

5 Experimental Setup And Results

We will focus on the results of experiments of our proposed model after we give the details about our datasets and evulation metrics that we used in experiments in this section.

5.1 Datasets and Evaluation

As it is shown in the model (Section 4), our model uses dependency parser so we need parsed data with manual annotation as gold standard data for the Set of Rules stage of our model or in order to identify the effect of the dependency parser to the score. For this reason, we collected a test and development set from Metu-Sabancı Treebank by (Eryigit et al., 2011),(Atalay et al., 2003) and (Oflazer et al., 2003) and we manually annotated the noun phrases of the set.

The set has 500 sentences, 1252 noun phrases, 5293 tokens, and 6333 relations in total. We divide the set into two parts that are equal number of sentences. We used the first part as development set which has 250 sentences, 557 noun phrases, 2250 tokens, 2745 relations and the second part as test set which has 250 sentences, 695 noun phrases, 3043 tokens, 3588 relations in total. We use two types of F1 scores first of which is F1 score over the noun phrases that matches fully all of the words of the gold standard noun phrases (F1Fullmatch Equation 1) and the other F1 score is calculated by the average(F1Pmatch Equation 2) of the F1 scores of partial match scores(pm Equation 3). The second metric is also used by Kutlu (2010).

$$F1_{Fullmatch} = \frac{\#\ of\ fullmatched\ noun\ phrases}{\#\ of\ noun\ phrases} \quad (1)$$

$$F1_{P_match} = \frac{\#\ of\ total\ pm\ of\ noun\ phrases}{\#\ of\ noun\ phrases} \quad (2)$$

$$pm = \frac{\#\ of\ corr\ match\ words\ of\ noun\ phrases}{\#\ of\ words} \quad (3)$$

5.2 Experiments

Since we work on a rule-based system, we need to the rule for start point of a chunk. For this aim, we calculate the distribution of numbers of types of relations onto the types of combination that consists of the chunk label of current word, and the chunk label of related word. For example, for the phrase “beyaz çiçek” (white flower), the chunk label of “beyaz” (white) is B, and the other chunk label is H and the type of the combination of chunk labels which belongs to the relation from the word “beyaz” (white) to the second word is ”BH”. From the chunk labels B,I,H,O , 16 different types of combinations can exist(BB,BI,BH,BO,IB,II,IH,IO etc...). The numbers of distribution of 26 types of relations to the 16 types of chunk label combinations tells that the start point of a chunk is any word that does not has an incoming relation in a sentence can be a chunk because the total number of combinations BB,IB,HB,OB is 0. Additional to it, more than 90% of the numbers of the relation types (modifier, determiner, possessor) states inside a chunk (has numbers BI, IH, II, BH chunk label groups).The third main idea which comes from the analysis is that more than 90% of the numbers of the relation types (subject, object) states outside a chunk or goes out of a chunk (has numbers HH, BO, IO, HO, OO chunk label groups). The last result that 100% of the numbers of the relation types (sentence, root) 6 tells that the end of the constituent should be the end of the opened chunk.

After we complete the algorithm by writing rules that are constructed by using the analysis of numbers of the chunk label combinations according to relation types, we evaluated our model both with the data which is parsed by manual annotation and the same data but parsed by automatically by the dependency parser.

System	F1 Score F. Match	F1 Score P.Match
Model+Parse By Manual An.	66.15	76.79
Model+Parse By Dep.Parser	47.91	60.75

Table-2 Results of The Proposed Model

As seen on the (Table 2), the quality of parsing directly effects the scores of our model. Our scores are 47.91% for full match and 60.75% for partial match. That means that our noun phrase chunker can find almost half of the noun phrases or the cuts the scope of the related topic in half in an NLP system and increases the performance of the system.

6 Conclusion And Future Work

Our study was a rule based model for chunking noun phrases in Turkish sentences which can assist many types of applications such as information extraction, text summarization, machine translation... etc.

We obtained F1 scores 47.91% for full match and 60.75% for partial match. This is the first study that does the chunking of noun phrases that are as complex as they have relative clauses to the best of our knowledge although a study of noun phrase chunker for Turkish has been done as an agglutinative language so it would be irrational to compare our model and the previous study. Since our model was a rule based model, the development of the model becomes impossible at the level of the state that the system start to develop rules that are for rare, single, and specific conditions. So, as the first future plan, we want to apply a machine learning method for noun phrase chunking or different machine learning methods to detect the machine learning method that gives the best results for Turkish.

The second future plan about noun phrase chunking is to use language models instead of using dependency relations because using language models would make the chunking time less and gave good results for fusional language such as English. The other four future plans for this study is to do the experiments how our study effects the scores of mainly NLP tools that does named entity recognition, dependency parsing, detecting of multi word expressions, sentiment analysis.

7 References

[1] Masayuki Asahara, Chooi Ling Goh, Xiaojie Wang, and Yuji Matsumoto. 2003. Combining segmenter and chunker for chinese word segmentation. In Pro- ceedings of the Second SIGHAN Workshop on Chinese Language Processing - Volume 17, SIGHAN'03, pages 144–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

[2] Nart B. Atalay, Kemal Oflazer, Bilge Say, and Informatics Inst. 2003. The annotation process in the turkish treebank. In Proc. of the 4th Intern. EACL Workshop on Linguistically Interpreteted Corpora (LINC).

[3] Michaela Atterer and David Schlangen. 2009.Rubisc: A robust unification-based incremental semantic chunker. In Proceedings of the 2Nd Workshop on Semantic Representation of Spoken Language, SRSL'09, pages 66–73, Stroudsburg, PA, USA. Association for Computational Linguistics.

[4] Claire Cardie and David Pierce. 1998. Errordriven pruning of treebank grammars for base noun phrase identification. In Proceedings of the 17th International Conference on Computational Linguistics - Volume 1, COLING'98, pages 218–224, Strouds- burg, PA, USA. Association for Computational Linguistics.

[5] Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In Proceedings of the Second Conference on Applied Natural Language Processing, ANLC'88, pages 136–143, Stroudsburg, PA, USA. Association for Computational Linguistics.

[6] V. Dhanalakshmi, P. Padmavathy, M. Anand

Kumar, K. P. Soman, and S. Rajendran. 2009. Chunker for tamil. In ARTCom, pages 436–438. IEEE Computer Society.

[7] Ilknur Durgar El-Kahlout and Ahmet Afsin Akin.2013. Turkish constituent chunking with morphological and contextual features. In CICLing (1), pages 270–281.

[8] Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can.2011. Multiword expressions in statistical dependency parsing. In Proceedings of the Second Work- shop on Statistical Parsing of Morphologically Rich Languages (IWPT - 12th International Conference on Parsing Technologies), pages 45–55, Dublin, Ireland, October. Association for Computational Lin-guistics.

[9] Gülşen Eryiğit. 2014. ITU Turkish NLP web service.In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April. Association for Computational Linguistics.

[10] Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of turkish. Computational Linguistics, 34(3):357–389.

[11] Mehmet Hengirmen. 2002. Tamlamalar. In Türkçe Dilbilgisi, pages 118–142. Engin.

[12] Kuang hua Chen and Hsin-Hsi Chen. 1993. A probabilistic chunker. In In: Proceedings of ROCLING VI, pages 99–117.

[13] Hannah Kermes and Stefan Evert. 2002. Yac – a recursive chunker for unrestricted german text. In Rodriguez M G, Araujo C P (eds), Proceedings of the Third International Conference on Language Resources and Evaluation, Las, pages 1805–1812.

[14] Mücahit Kutlu. 2010. Noun phrase chunker for Turkish using dependency parser.

[15] Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani Tür , and Gökhan Tür. 2003. Building a turkish treebank.

[16] Adam Radziszewski and Maciej Piasecki. 2010. A preliminary Noun Phrase Chunker for Polish. In Intelligent Information Systems, pages 169–180. Springer.

[17] Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. CoRR, cmp-lg/9505040.

[18] Muhammet Şahin, Umut Sulubacak, and Gülşen Eryiğit. 2013. Redefinition of Turkish morphology using flag diacritics. In Proceedings of The Tenth Symposium on Natural Language Processing (SNLP-2013), Phuket, Thailand, October.

[19] Manabu Sassano and Takehito Utsuro. 2000. Named entity chunking techniques in supervised learning for japanese named entity recognition.

[20] Akshay Singh, Sushma Bendre, and Rajeev Sangal.2005. Hmm based chunker for hindi. In In the Proceedings of International Joint Conference on NLP.

[21] Kristina Vuckovic, Marko Tadic, and Zdravko Dovedan. 2008. Rule-based chunker for croatian.In LREC. European Language Resources Association.

[22] İnternet Sistemleri Konsorsiyomu, www.isc.org/solutions/survey, Erişim tarihi: Kasım 2011.

[23] xxx. 2014. Morphological processing of turkish. xxx, x(x), x.