Automatic Subject-Object-Verb Relation Extraction

Merve Temizer
Software Development
Gobito Enterprise Solutions
İstanbul, Turkey
mervet2009@gmail.com

Banu Diri
Computer Engineering
Yıldız Technical University
İstanbul, Turkey
banu@ce.yildiz.edu.tr

Abstract— **Artificial Intelligence is one of the key concepts of today's technology. As it is known, AI's aim is to developing technology that can learn by itself. Also, Natural Language Processing is another key concept as a significant contributor to AI in the field of natural languages. Considering the AI and NLP together brings us to teach computers to learn on their own about the natural languages and human derived words with their relationship. This paper aims to transfer a considerable amount of information to computers' world by presenting a way to extract Subject-Object-Verb relation extraction from Turkish documents automatically. Through three main steps the goal is achieved: (1) morphological analysis, (2) dependency analysis, (3) triplet extraction. As a result, an independent triplets graph can be generated for each text input, and verbs- nouns relation can be viewed.**

Keywords-dependency analysis; subject-object-verb relation extraction

I. Introduction

AI presents the least time consuming way by redirecting the researchers to find ways in those computers gets the information on their own rather than requiring people to tell everything to computers Through this paper, an application that extracts the Subject-Object-Verb triplets (SOVt) of a document and transfers more information to digital world is aimed to be developed. These triplets can be considered as a resource of relation information about all the verbs and nouns in Turkish. Also it may be used to contribute to Turkish WordNet.

There are some other papers in this field. For example, [1] consider the triplet extraction for English based on parse trees of sentences. An other paper [2] is about triplet extraction in Korean texts. Also there is another research [3] about triplet extraction using a different method, Support Vector Machines. A machine learning approach to extract SOVts from English sentences is described in [3]. SVM is used to train a model on human annotated triplets, and the features are computed from three parsers. Beside this, a hybrid approach is used in [4]. Approach employs knowledge-based and (supervised and unsupervised) corpus-based techniques.

In this paper, a common diagram of flow that is followed as the work list in order to develop the method, is defined in section II. In the following sections, the steps; text preprocessing, morphological analysis of words, dependency analysis of sentences and finally triplet extraction are described in detail. In section IV, an output generated by the described section for an input takes place as result experiment.

978-1-4673-1448-0/12/$31.00 ©2012 IEEE

II. Common Flow

The main work flow consists of the steps; preprocessing the input text by using Java Standard Edition's [5] string processing facilities, morphologically analyze the words using Zemberek [6], dependency analysis of sentences with a rule based algorithm that is described in [7], triplet extraction from the relations between the dependent words. Fig. 1, illustrates the common flow diagram.

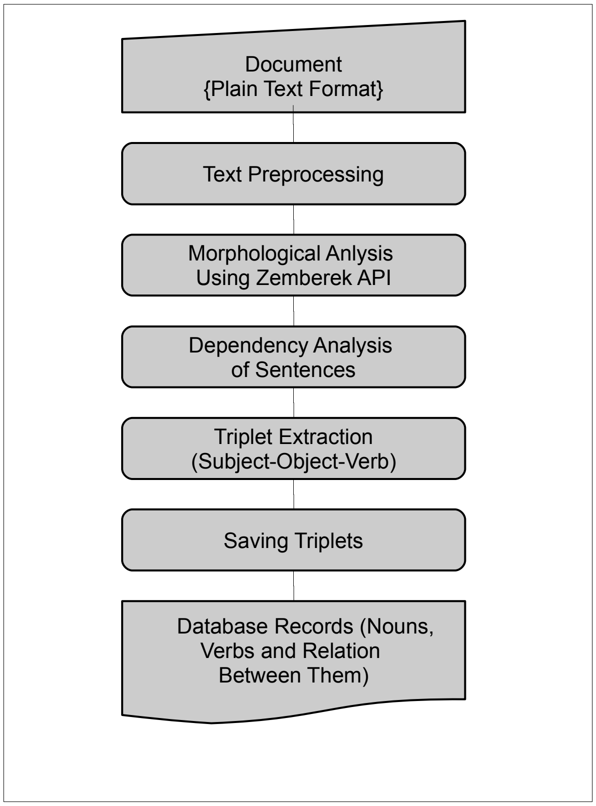


Figure 1. SOVt extraction diagram

III. Text Preprocessing

Input as plain text that consists of sentences is tokenized with the “String” object of Java Standard Edition's [5] “lang” package. In this procession, an undetailed sentence end recognizing mechanism is used. A Sentence Java class is defined and after defining the sentences' ends, all the sentences is defined as a Sentence object. Also the text is defined as an array of sentences. Also only by considering the space character the words are separated from each other. A Word class is described. Then the Word objects are defined for each word in the text and the words which belong to one sentence are put into an array of Word objects associated to its Sentence object.

IV.Morphological Analysis

Thus far, a list of words have been got already. After having the words individually, words are analyzed using Zemberek API [6] morphologically. For example, it is convenient to detail how Zemberek [6] analyzes the word “kitabını”. As it is known, the word “kitabını” can be analyzed in three different ways:

- *Root: kitap, ISIM, Formatives: ISIM_TAMLAMA_IN + ISIM_BELIRTME_I*
- *Root: kitap, ISIM, Formatives: ISIM_SAHIPLIK_O_I + ISIM_BELIRTME_I*
- *Root: kitap, ISIM, Formatives:ISIM_SAHIPLIK_SEN_IN + ISIM_BELIRTME_I*

To detail the morphological analysis, it is appropriate to describe word “morphological”. Morphologic means “related to form”. Thus, it can be said that morphological analyze is analyze words by using only their forms, not meanings. As it is known, Turkish is an agglutinative language and Zemberek can split words into their roots and formatives. Zemberek API is a kit of Java class packages that provide everything to parse Turkish words morphologically. In an Object Oriented way, a SentenceWord class has been defined. After a loop that run the morphological analysis of every word in the text, morphological units of a word has been assigned to sentence's word objects those are derived from SentenceWord class. The morphological units can be called as Inflectional Groups. The Inflectional Groups of all of words helps during dependency analysis of sentences which is going to be detailed in the next sub-topic. To perform the parsing algorithm that is concerned in the next sub-topic, a Sentence class is described in Java. This Sentence class is responsible for hold the word sequence of a sentence in the text. Also, the Sentence class has some other fields for triplet extraction.

V.Dependency Analysis Of Sentences

For SOVt relation extraction, it is needed to parse a sentence to its components. To parse a sentence there is several methods. Some of these methods use Lexical Functional Grammar formalism, some other use an entity based technique and some other use rule based dependency parsing technique. In this paper, rule based dependency parsing[7] technique is used to parse the sentences to extract triplets those have SOVt relation. The most valuable resource for this paper is [7] which is the first rule-based dependency analyzing paper in Turkish.

In this technique every words in the sentences have to be separated into Inflectional Groups. As mentioned in early topics, the Inflectional Groups are extracted using Zemberek API with an Object Oriented approach.

After having all the words as an object, the dependency analysis can be started. The rule-based dependency analysis describe a way to understand this kind of analysis. In this way, every word is dependent to another. For Turkish sentences it can be said that every word must be dependent to a word and can be dependent only one word. Also a word can have none, one or more dependents. Fig. 2 demonstrates dependency relations between words in a Turkish sentences[8].

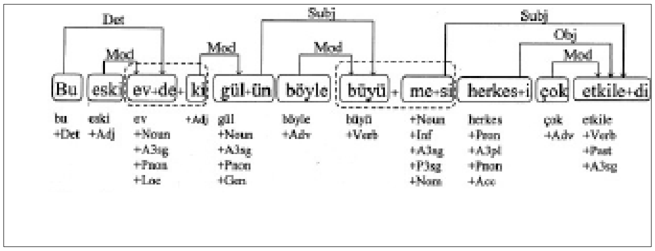


Figure 2. Sentence dependency diagram

The method followed in this paper is slightly different from the method in [7]. Eryigit [7], advises to define to structures those can be named as a queue and a stack and move the Inflectional Groups to the stack from queue or vice versa. In this paper, a queue and a stack is used in exactly the same manner. The little difference of this paper is that it use words as the units of stack and queue and move the words between stack and queue. This is not against of the nature of the main rule based dependency analysis method because the words must be dependent to a word and can be dependent to only one word. The words with a dependent word can be called as head of the dependent word. Also the connections between heads and dependents can be named as links. To generate all the necessary links, there is an algorithm and a parsing model in [7] that uses a stack and a queue for Inflectional Groups of words. In this paper, words are going to be moved instead of Inflectional Groups. But the algorithm rules are going to be used to decide to move or remove or where to move the words according to Inflectional Groups that is defined in morphological analysis step.

VI.Triplet Extraction

Triplet extraction is getting the SOVt relation from the sentences. In the previous section, using dependency analysis method, it is represented how sentences can be parsed. To extract triplets, all the verbs in a sentence are found. After finding verbs, the dependents of the verbs are determined. After determining the dependents of verbs, the dependents of dependents is considered. Also, it is determined that which dependent is subject and which dependent is object. After these operations, for every verb in the sentence a Set is described which has the SubjectOfSet, ObjectOfSet and VerbOfSet. The algorithm can be viewed in Fig. 3 is describing the triplet extraction from the sentences those their words' dependency relations are defined in the previous section.

```
for each verb in verbs_of_a_sentence do
    for each dependent in dependents_of_verb do
        if dependent is a noun
            if dependent has an addition
                describe dependent as object of verb
            else
                if dependent is immediately preceding of verb
                    describe dependent as related to verb, ambiguous_if_subject_or_object
                else dependent is subject

for each noun in nouns_of_a_sentence do
    for each dependent in dependent_gerundial_of_noun
        if dependent has gerundial formative FIIL_BELIRTME_DIK and formative ISIM_TAMLAMA_I and formative ISIM_SAHIPLIK_*
            describe dependent as object of verb
        else
            describe dependent as subject of verb
```

Figure 3. SOVt extraction algorithm

VII. Experimental Result

In this section, to demonstrate the process, an input Turkish sentence is considered. For every step the output of that step is presented. For the input sentence “Pazara gidip, bir top aldı.”

For the preprocessing step, sentence is separated into words by space characters, as an output, “pazara”, “gidip”, “bir”, “top”, “aldi” words have been got.

For the morphological analysis, the following words are defined using Zemberek [6]:

- *Root: pazar, ISIM, Formatives: ISIM_YONELME_E*
- *Root:gidip, FIIL, Formatives: FIIL_IMSI_IP*
- *Root:bir, SAYI*
- *Root:top, ISIM*
- *Root:al, FIIL Formatives: FIIL_GECMISZAMAN_DI*

Zemberek [6] generates more than one option, in application's dependency analysis part all of the combination of options are tried to analyze sentence. According to the combination which allow to parse the sentence, the words are updated with information of who are the dependent of it and it depends to whom. For the above example sentence, Fig. 4 which illustrates the word relations.

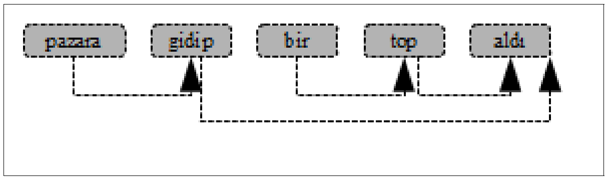
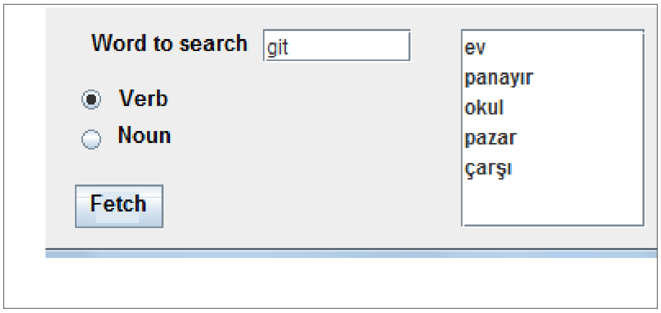


Figure 4. Dependency diagram for example sentence

After dependency analysis, SOVt extraction process is run. There are two triplets:

- *Subject: o (hidden subject), Object: pazar, Verb: git*
- *Subject: o (hidden subject), Object: top, Verb: al*

After extraction relations are saved into a relational database. And they can be queried. In Fig. 5, a screen of querying can be viewed. The list consists of the nouns those are related to verb “git”.



Word to search: git

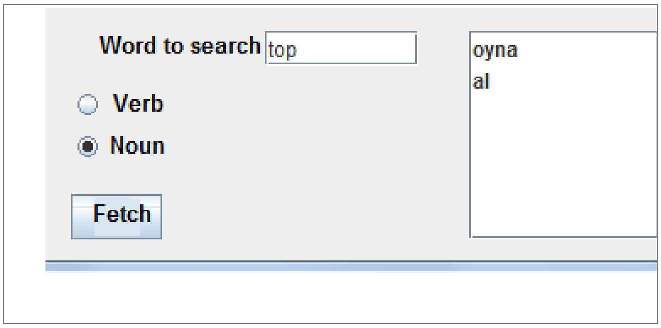
● Verb

○ Noun

Fetch

ev
panayır
okul
pazar
çarşı

Figure 5. Query for verb “git”



Word to search: top

○ Verb

● Noun

Fetch

oyna
al

Figure 6. Query for noun “top”

VIII. Conclusion

In this paper, an algorithm for extraction SOVts of sentences of texts is described. Also a top-down way is explained by sampling the usage of rule based dependency analysis [7]. The automatic noun-verb relations extraction may be used to browse remarkable text source of web and to contribute to Turkish WordNet.

References

[1] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, “Triplet extraction from sentences”, Proceedings of the 10th International Multiconference, Information Society – I 2007, vol. A, pp. 218-222, 2007.

[2] D. Choi and K. Choi, “Automatic relation triple extraction by dependency parse tree traversing”, EKAW 2008.

[3] L. Dali and B. Fortuna, “triplet extraction from sentences using SVM”, SiKDD 2008, Ljubljana, Slovenia, October 2009.

[4] L. Specia and E. Motta, “A hybrid approach for extracting semantic relations from texts”, COLING-ACL 2nd Workshop on Ontology Learning and Population (OLP2-2006), Sydney, Australia, pp. 57-64, 2006.

[5] www.java.com

[6] code.google.com/p/zemberek/

[7] G. Eryiğit, E. Adalı, and K. Oflazer, “Türkçe cümlelerin kural tabanlı bağlılık analizi”, Proceedings of the 15th Turkish Symposium on Artificial Intelligence and Neural Networks, Mugla, Turkey, 21-24 June 2006, pp. 17-24.

[8] K. Oflazer, “Dependency parsing with an extended finite-state approach”, Computational Linguistics, vol. 29, no. 4, pp. 515-544, 2003.