Web-based Acquisition of Subcategorization Frames for Turkish

Erdinç Uzun*, Yılmaz Kılıçaslan, Volkan Agun, and Erdem Uçar

Department of Computer Engineering, University of Namık Kemal,
Çorlu Mühendislik Fakültesi, 59860 Tekirdağ, Turkey*
Department of Computer Engineering, University of Trakya,
Ahmet Karadeniz Yerleşkesi, 22030 Edirne, Turkey
{erdincuzun@corlu.edu.tr},{yilmazkilicaslan,volkanagun,
erdemucar}@trakya.edu.tr
http://tbbt.trakya.edu.tr/

Abstract. The aim of this paper is two-fold. Firstly, it presents a system that is intended to perform three main functions: 1) to collect grammatical sentences with automated search queries, 2) to annotate these sentences with case information on nominal constituents, and 3) to automatically acquire subcategorization frames (SFs) of Turkish verbs using the naïve Bayesian classifier. Secondly, the paper reports on the experiments performed to acquire Turkish SFs. The experimental results show that the web can be effectively used for a task of machine learning like automatic SF acquisition.

Key words: Web as Corpus, Subcategorization Frames, Supervised Methods.

1 Introduction

The World Wide Web is an invaluable source of data for linguistic research, because it contains abundance of texts in various domains for large number of languages. These texts are indexed by search engines. Researchers show that corpuses derived from the web are very useful to improve various natural language based studies such as word sense disambiguation, machine translation and automatic question answering [7]. However, search engines have some limitations when dealing with morpho-syntactic peculiarities of natural languages. Further, the more a language is typologically distant from English, the less a search engine is capable of handling linguistic properties.

In this study, we describe how a Turkish corpus can be constructed by automated queries. Then, we focus on automatic acquisition of SFs of Turkish verbs from the corpus constructed in this way.

A Subcategorization Frame is an expression of what kind of and how many syntactic arguments a word takes. Subcategorization information is important especially for parsing and grammar development as it provides the parser with

syntactic and/or semantic constraints on a word. This information is coded manually in much work in natural language processing or computational linguistics. However, coding subcategorization information manually is slow, labor-intensive and error-prone. As grammars or parsers get bigger, a manual approach to the collection of subcategorization information becomes even more unattractive and impractical [1], [9], [13], [10], [11]. Furthermore, this information is totally unavailable in digital form for many languages.

A promising solution to these problems can be achieved by extracting SFs from textual corpora using machine learning methods. Many techniques and results have been reported concerning this issue. However, to the best of our knowledge, a corpus automatically derived from the web has been used in no such work.

As for Turkish, the only work concerning SF acquisition in this language is one of ours [8]. In this study, we try to apply two well-known statistical methods on Turkish data, namely Likelihood and t-score, in order to automatically acquire SFs. This study shows that Turkish is a more challenging language than English with respect to the task at hand. Given the unsatisfactory results obtained with the two unsupervised methods (F-Measure: 46%), we propose here to approach the problem using the naïve Bayesian classifier in a supervised learning setting (F-Measure: 77%).

2 System Overview

The system is composed of four components[1]: a verbal generator, which generates all different conjugations of a Turkish verb; a web-based sentence collector (WBSC), which retrieves sentences headed by any of these verbal items; a case tagger, which annotates nominals based on their case information; and a learning layer, which performs automatic acquisition of SFs by using machine learning methods. Figure 1. shows the architectural design of the system.

In what follows is a more detailed account of each of system components constituting the system.

2.1 Verbal suffix generator

Turkish is a highly agglutinative language, which has, therefore, a large vocabulary of words. Also, it is an exclusively suffixing language. Inflectional suffixes may be divided into two groups, a noun paradigm and a verb paradigm. What is needed when implementing the verbal suffix generator is the verb paradigm, the elements of which are as follows:

[1] The downloadable codes of the components are accessible via http://tbbt.trakya.edu.tr/download.htm

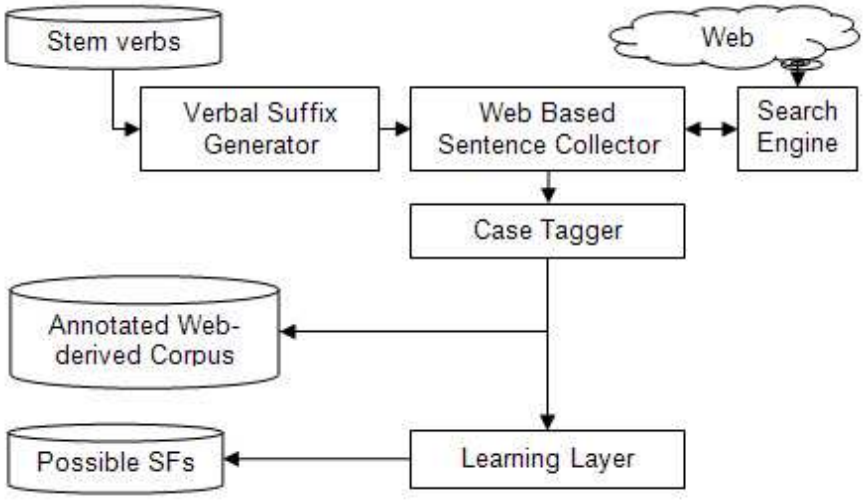

Fig. 1. System Overview

(1) 1. Verb Stem
2. Derivation: reflexive *-In*, reciprocal *-In*, causative *-DIr*, passive *-Il*
3. Negative: inability *-(y)E* and negative *-mE*
4. Tense: aorist *-Ir*, progressive *-Iyor*, definite past *-DI*, narrative past *-mI*, future *-(y)EcEk*, optative *-(y)E*, necessitative *-mElI*, conditional *sE*
5. Auxiliary: past *-(y)mI*, conditional *-(y)sE*, adverbial *-(y)ken*
6. Person

Turkish suffixes alter in form due to some morphophonemic processes, such as vowel harmony. In this dissertation, capital letters will be used as cover symbols for morphemes alternating between differing forms. (2) shows the symbols used for that purpose along with the Turkish characters that they stand for. (The IPA equivalents of the non-English orthographic symbols are given in square brackets.)

(2) *Symbol Surface Form*
I *i, ü [y], ı [ɯ], u*
E *e, a*
D *d, t*
C *c, ç [tʃ]*
G *g, ğ [ɣ], k*

Another alternation results from the fact that some suffixes are preceded by a buffer consonant, *y* or *n*, according to whether they follow a vowel or a consonant. We write the buffer consonants in brackets.

With all these suffixes and variations a Turkish verbs can have up to 40.000 different forms [6]. When it comes to English, the number of verbal forms can be at most 6. Furthermore, search engines support word stemming (dropping the 'ed', 's' and 'ing' from different verb forms) for English but not for Turkish. Therefore, a verbal suffix generator that generates all different conjugations of a Turkish verb is of vital importance for any sort of NLP application on the verb concerning Turkish.

2.2 Web-based sentence collector

The web based sentence collector, firstly, prepares a query string for an HTTP request. The query string starts with the host name or IP address, and is followed by the string "/search?" and one or more name-value pairs separated with an ampersand (&) character. The following is an example query string for Google:

http://www.google.com.tr/search?hl=tr&q=gidecek&num=100&lr=tr

When the query string has been composed, the component sends the search engine an HTTP request. The search engine returns the results in HTML format. Having retrieved the HTML documents, WBSC starts a filtering process to obtain legitimate Turkish sentences from the title and short description. The filtering process comprises two sequentially fulfilled tasks:

- Remove HTML tags (such as , , ,) and illegal characters (such as ' , => , &quot), and if required replace illegal characters with the appropriate ones.
- Eliminate similar sentences and sentences containing only one word.

Briefly, given a Turkish verb, this component returns a set of sentences free of HTML tags and illegal characters[2].

2.3 Case tagger

Elements of SFs of Turkish verbs are discriminated from each other through case marking. Turkish nominals can occur with four different cases:

Case	*Abbreviation*	*Suffixes*
nominative	(NOM)	*(-)*
accusatice	(ACC)	*(y)(-i, -ı, -u, -ü)*
dative	(DAT)	*(y)(-e, -a)*
ablative	(ABL)	*(-den, -dan)*

[2] See [14] for a more detailed account of the web-based sentence collector.

In addition to these four cases, Turkish also has a clitic particle that has a case-like function:

Case	*Abbreviation*	*Suffixes*
instrumental	(INST)	*(y)(-le, -la)(ile)*

In Turkish, there can be $2^4 = 16$ possible different SFs with these case suffixes.

The case tagger firstly searches a given word in the dictionary. If the word is in the dictionary, the case tagger annotates it with nominative case. Otherwise, the word is checked as to whether it ends with a case suffix. If it carries a case suffix, it is annotated with the encoded case. Consider the following example:

(3) Adam-∅ kadın-ı öptü.
man-NOM kadın-ACC kissed

"The man kissed the woman"

Adam "man"will be found in the dictionary and, hence, it will be marked with the nominative case. However, kadın-ı "woman-ACC"cannot be included in a dictionary as it bears case morphology, which makes the tagger annotate it with a case.

As a last possible case, the word may not be in the dictionary and may not carry case morphology. In that case, it would be a nominative case-marked nominal, which is either a proper name or a common noun bearing some morphology whose content is not relevant for the work at hand. Consider the example below:

(4) Çocuk-lar koştu
child-PL ran

"The children ran"

As the subject of the sentence in (4) carries the plural suffix, *-lEr*, the case tagger will not find it in the dictionary. This suffix will not be detached to get to stem but the whole word will be annotated with the nominative case. We need not worry about the number information lost in this way because it does not bear any relevance to the task at hand.

2.4 Learning layer

In our earlier work on SF acquisition, we saw that purely statistical methods such as likelihood and t-score are unsatisfactory for automatic acquisition of SFs of verbs in Turkish [8]. In this present work, we apply the naïve Bayesian classifier to Turkish data extracted from the web as described above.

The naïve Bayesian classifier is a simple probabilistic classifier based on applying the Bayesian theorem with naïve independence assumptions. Given a set of training instances with SFs and a test instance SF represented by the SF values $< a_1, a_2, ...a_n >$, a Bayesian classifier uses the following equation to classify

SF:

$$c(SF) = \arg\max_{ceC} P(c)P(a_1, a_2, ...a_n|c)$$

where

$P(a_n)$= Count of SF for a given verb / count of sentences for a given verb

Despite this assumption is almost always violated in practice, recent studies have shown that this classifier is remarkably effective in practice [5]. The class probability $P(c)$ can be estimated from training data. To make this estimation, we use the Expectation Maximization (EM) algorithm [4], which is a general framework for estimating the parameters of a probability model when the data has missing values.

3 Evaluation

3.1 Data Collection and Evaluation Method

200 verbs were randomly chosen from the dictionary. Then, 16,841 sentential instances headed by these verbs were extracted from the data fetched by the Google search engine. The instances were fed to the case tagger for annotation. Having prepared the annotated Web-derived corpus, each SF was trained and cross-validated for 10-times with a 10-fold random sampling by the Naïve Bayesian classifier.

The results of the experiments were evaluated using *precision* (percentage of actually correct SFs to the SFs suggested as correct) and *recall* (percentage of SFs suggested as corrected to all possible SFs). We also combined precision and recall into a single measure of overall performance using the *F-measure*[15]:

$$F = \frac{2.precision.recall}{precision+recall}$$

3.2 Results and Discussions

In this work, firstly we investigate the effect of the size of the training data on the learning performance of the system. We started our experiments with a pattern set containing three sentences for each verb. In the subsequent experiments, we used pattern sets automatically enlarged to contain 3, 5, 10, 15, 20 and more sentences for each verb. The results obtained in this way are shown below:

# of sentences	Precision	Recall	F-Measure
3	66%	84%	74%
5	66%	84%	74%
10	67%	84%	75%
15	72%	82%	77%
20	73%	81%	77%
$\succ$20	74%	81%	77%

Table I. Effects of data size on the learning performance

We observe that the performance gets better as the size of the data goes up. The highest performance scores are as follows: 74% of precision, 81% of recall and 77% of F-measure. It is worth noting that these results are much better than those we obtained with two unsupervised methods, log likelihood and t-score: 31% of precision, 91% of recall and 46% of F-measure (for both methods) [8]. It is also noteworthy that these two latter methods are reported to perform quite well for languages like Czech [13], Greek [10] and Bulgarian [11]: around 84% of F-Measure. Thus, Turkish appears to be more challenging for the task of SF acquisition compared to many other languages. [3] A modest way to escape from this challenge is to have recourse to supervised methods. Our late experiments show that with the naïve Bayesian classifier the performance results come close to those obtained for other languages.

4 Conclusion

The paper has presented a system intended to overcome the difficulties peculiar to Turkish in the process of extracting a corpus from the Web. The first and most important difficulty in this process has to with the agglutinative nature of Turkish. When collecting instances, each instance is associated with a verbal stem. Nonetheless, a Turkish verb seldom appears in its base (citation) form in the Web documents (or in any other text). This obstacle is circumvented by a suffix generator which generates all possible suffixed forms of a verbal stem. Another difficulty arises from Turkish characters not supported by some applications and noisy and sparse data on the web. It is a web-based sentence collector that deals with such problems. A tagger is responsible for annotating the case value of nominals occurring in the documents. The naïve Bayesian classifier is applied to this annotated data in order to predict the SF of each verb. The components of the system can be customized and/or integrated into various other applications.

The performance results obtained in the experiments conducted with the developed system suggest that the Web can serve as a good source of data for learning tasks.

[3] See [8] for a concise discussion of the dimensions along which Turkish exhibit difficulties for SF acquisition.

References

1. Basili R., Pazienza M.T. & Vindigni M.: Corpus-Driven Unsupervised Learning of Verb Subcategorization Frames. Proceedings of the 5th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence, Number 1321 in LNAI, Springer-Verlag, Heidelberg, Germany, (1997) 159–170
2. Baroni, M., & Bernardini, S.: Bootcat: Bootstrapping corpora and terms from the web. Fourth Language Resources and Evaluation Conference (LREC2004), Lisbon (2004)
3. Chesley, P. & Salmon-Alt, S., Automatic Extraction of Subcategorization Frames for French, Language Resources and Evaluation Conference (LREC 2006), Genua, Italy (2006)
4. Dempster, A., Laird, N., & Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, (1977) 1–38
5. Domingos, P., & Pazzani, M.: Beyond Independence: Conditions for the optimality of the simple bayesian classifier. Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, (1996) 105-112
6. Jurafsky, D. and Martin J. H.: Speech and Language Processing. Prentice-Hall, New Jersey (2000)
7. Kilgarriff, A., & Grefenstette, G.: Introduction to the special issue on the Web as Corpus. Computational Linguistics , vol 29(3), (2003) 333–347
8. Kılıçaslan, Y., Uzun, E., Agun, V., & Uçar, E.: Automatic Acquisition of Subcategorization Frames for Turkish with Purely Statistical Methods. International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2007), İstanbul (2007) 11–15
9. Manning D. C.: Automatic Acquisition of a Large Subcategorization Dictionary from Corpora. Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio (1993) 235–242
10. Maragoudakis, M., Kermanidis K. & Kokkinakis G.: Learning Subcategorization Frames from Corpora: A Case Study for Modern Greek. Proceedings of COMLEX 2000, Workshop on Computational Lexicography and Multimedia Dictionaries, Kato Achaia, Greece, (2000) 19–22
11. Marinov S.: Automatic Extraction of Subcategorization Frames for Bulgarian. Proceedings of 16th ESSLLI Student Session, Nancy, France (2004)
12. Resnik, P., & Smith, N. A.: The Web as a Parallel Corpus. Computational Linguistics , vol 29(3), (2003) 349-380.
13. Sarkar, A. & Zeman, D.: Automatic Extraction of Subcategorization Frames for Czech, Proceedings of the International Conference on Computational Linguistics, COLING-00, Saarbrucken, Germany (2000) 691–697
14. Uzun, E., Kılıçaslan, Y., & Uçar, E.: Web Based Sentence Collector. Ninth International Scientific Conference 2007 (SMOLYAN-2007). Smolyan, Bulgaria (2007)
15. van Rijsbergen, C. J.: Information Retrieval. London: Butterworths (1979)