A Suffix Based Part-of-Speech Tagger for Turkish

Taner DINCER
Mugla University, Department of Statistics

Bahar KARAOGLAN
Ege University, International Computer Institute

Tarik KISLA
Ege University, International Computer Institute

Abstract

In this paper, we present a stochastic part-of-speech tagger for Turkish. The tagger is primarily developed for information retrieval purposes, but it can as well serve as a light-weight PoS tagger for other purposes. The tagger uses a well-established Hidden Markov model of the language with a closed lexicon that consists of fixed number of letters from the word endings. We have considered seven different lengths of word endings against 30 training corpus sizes. Best-case accuracy obtained is 90.2% with 5 characters. The main contribution of this paper is to present a way of constructing a closed vocabulary for part-of-speech tagging effort that can be useful for highly inflected languages like Turkish, Finnish, Hungarian, Estonian, and Czech.

1. Introduction

Information Retrieval (IR) systems are used to handle information gathered from a large amount of electronic documents. Information on a document is composed of words' semantics. Hence, an IR system actually deals with those words, which are the representatives of semantics that are truly the building blocks of intended information. Index term selection is a task of finding manually or automatically the words or collocations that are the representatives of the potential information on a particular document. Those terms are then used to represent the document in an IR system for further processing purposes. The potential information in a document is mostly represented by the noun part-of-speech (PoS) category of the word forms [3]. Thus, an index term is commonly a noun word or noun word collocation. Consequently, PoS tagging is a preliminary step for indexing collocations for linguistically motivated IR purposes [1, 9]. Vocabulary based PoS tagging effort for agglutinative languages as Turkish is problematic because of rich set of part of speech categories. The vocabulary that should be stored for PoS tagging can theoretically be an infinite set of word forms [24]. On the other hand, even for analytical languages like English, the same problem still exists in different manner resulting in the failure of closed vocabulary assumption in principal.

In this paper, we offer a closed vocabulary formation to be used in PoS tagging for Turkish which is sure to have an open vocabulary for being agglutinative and having productive generative morphology. We investigate vocabularies of different sizes constructed by taking different fixed length unity of characters from the word form ends as lexemes for the lexicon of HMM model. Actually, the usage of the word endings is not a new concept in the literature for PoS tagging effort in smoothing probabilities of unknown word-tag pairs [6, 10]. However, accomplishing all the work with just word endings, to our current knowledge, has never been attempted for agglutinative languages. On the other hand, this approach is theoretically reasonable for agglutinative languages like Turkish, since all the part of speeches are generated from a noun root or a verb root by the use of a finite set of suffixes or by no suffix at all. The results reveal that, the closed vocabulary assumption can be provided and a finite set of fixed number of characters can be used instead of the actual set of word forms for Turkish with an acceptable accuracy of 90.2%.

The paper is organized as follows: In section 2, we give a brief review of previous work, background information related to HMM language model and Turkish. In section 3, we present our research methodology, corpora and the results. Section 4 is the conclusion.

2. Part-Of-Speech Tagging

2.1 Previous Works

In computational linguistic literature, there are several different approaches to the problem of assigning grammatical function to a word form in a particular sentence including the pioneering works of Klein and Simpson [28] and Garside et al. [20, 21]. These divide the area into two main streams according to their theoretical context: Rule-based and Statistical. Klein's work [28] appears to be based on roughly the same idea as Salton's and Thorpe's [23]. Klein uses the

978-0-7695-3099-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ITNG.2008.103

terms 'tag' and 'tagging' which are in fact interchangeable with 'code' and 'codes' as in Salton's work, and it is the first rule-based tagging program based on a large set of hand-constructed rules and a small lexicon to handle the exceptions. In the same line, TAGGIT system, developed by Greene and Rubin [22], was used for initial tagging of well-known Brown corpus. TAGGIT uses the lexicon only to eliminate the impossible tags, and only applies the tagging rules to assign the grammatical function of a word form by the use of its morphology if the grammatical functions of the previous and the next word forms are unambiguous. The output of this tagger was then corrected by hand and it took many years. The other examples, which contribute to this area, include Brill [7, 8] and Church [13]. In the counter-part of the development, Stolz et al [38] have reported the earliest work for Statistical/Probabilistic context. This program initially assigns tags of known word forms by the use a lexicon and then assigns the tags of unknown words by the use of conditional probabilities calculated from the tag sequences. In the line of Statistics, majority of the work fall into the Markov model of the language introduced by Markov [31]. The first tagger based on Markov model was developed in the University of Lancaster as a part of the LOB (Lancaster-Oslo-Bergen) corpus tagging effort by Garside et al. [20] and Marshall [32]. This tagger was based on the use of probabilities of bigram tag sequences, with the limited use of the higher context, but the probabilities of having a different part of speech for a particular word was assigned by heuristic discounting method. The key usage of Markov model with both word and tag transition probabilities was done by Church [14] and DeRose [15]. These works of Markov model PoS tagging are the resurgence of the statistical methods in computational linguistics after Chomsky's criticisms on the inadequacies of this model [12] at the early sixties. The work on Markov model based tagging had actually begun much earlier including the studies of Bahl and Mercer [33], Baker [4], Jelenik [27], Derouault, and Merialdo [18]. Taggers are currently widespread and have been developed for a number of different languages such as Basque [2], Dutch [16], French [11], German [19], Greek [16], Hebrew [29] , Italian [16], Slovene [15], Spanish [37], and Swedish [5]. Moreover, extended comparisons of Taggers for seven languages can be found in Dermatas and Kokkinakis [16] work. The PoS tagging of Turkish texts, which had employed a rule-based approach is introduced by Oflazer et al. [35] and a statistical approach by Hakkani-Tür et al. [26], in the manner of morphological disambiguation. The PoS tagger made by Hakkani-Tür et al. performs with 93.95% accuracy by using words as a calculating unit.

2.2 Background Information

2.2.1 The HMM language model. A stochastic model of Part-of-Speech problem may be formulated as:

$$\operatorname{argmax}_{t_1 \cdots t_n} P(t_1 \cdots t_n | w_1 \cdots w_n)$$

This gives the possible PoS tag sequence $t_1 \cdots t_n$, which maximizes the conditional probability $P(t_1 \cdots t_n | w_1 \cdots w_n)$ given a sequence of words $w_1 \cdots w_n$, By applying Bayesian inversion, we can rewrite the formula given above as:

$$\operatorname{argmax}_{t_1 \cdots t_n} \frac{P(w_1 \cdots w_n | t_1 \cdots t_n) \cdot P(t_1 \cdots t_n)}{P(w_1 \cdots w_n)}$$

Since the probability of the given sentence is maximized against the tag sequence $t_1 \cdots t_n$, and the word sequence probability $P(w_1 \cdots w_n)$ is constant for a particular sentence, the model formula can further be rewritten in its final form as:

$$\operatorname{argmax}_{t_1 \cdots t_n} P(w_1 \cdots w_n | t_1 \cdots t_n) \cdot P(t_1 \cdots t_n)$$

A Markov model is a stochastic process of identically independent distributed set of random variables $\{X_i\}$ indexed over a suitable set of time values $i \in T$ having two characteristic properties [39]: Limited horizon,

$$P(X_n = k | X_0 = l_0 \cdots X_{n-1} = l_{n-1}) = P(X_n = k | X_{n-1} = l_{n-1})$$

and, Time Invariance,

$$P(X_n = k | X_{n-1} = l_{n-1}) = P(X_1 = k | X_0 = l_{n-1})$$

for all time points n and for all states $k, l_0, \ldots l_n$. A discrete time Markov model is defined over an index set $T = \{0,1,\ldots\}$ of discrete values of time, and a continuous time Markov model is defined over an index set $T = \{j | j \in \Re\}$ of continuous values of time. Consequently, a PoS tagging problem can be modeled with a discrete time Markov model. As a result, with some reinterpretation of the final stochastic formula of PoS tagging problem, new formulation can be rewritten by the use of Limited horizon property as [30]:

$$\operatorname{argmax}_{t_1 \cdots t_n} \prod_{i=1}^{n} P(w_i | t_i) \cdot P(t_i | t_{i-1})$$

With the aid of this formulation, we can easily calculate the probability of a given sentence by the use of probability of a word given a tag and the 2-gram tag transition probabilities without knowing what the empirical joint distribution of word and tag sequences is, as opposed in the case of the fundamental stochastic process model. In this way of calculating probability of a particular sentence by the use of one item back in the sequence of tags is called as first order Markov model (i.e. analogy to 2-gram language model) and using two items back in the tag sequence is called as a first order Markov model with one-step history in the state space Thus, by a first order Markov model with one-step history in the state space, a PoS tagging problem can be re-modeled as:

$$\operatorname{argmax}_{t_1 \cdots t_n} \prod_{i=1}^{n} P(w_i|t_i) \cdot P(t_i|t_{i-2},t_{i-1})$$

To estimate the two complementary probabilities of this formula, we use an annotated training corpus to observe tag pair frequencies $f(t_{i-2},t_{i-1},t_i)$ and $f(t_{i-2},t_{i-1})$, and tag-word pair frequencies $f(t_i,w_i)$, after that, MLE (Maximum Likelihood Estimation) can be directly calculate as:

$$\hat{P}(w_i|t_i) = \frac{f(t_i,w_i)}{f(t_i)}$$

$$\hat{P}(t_i|t_{i-2},t_{i-1}) = \frac{f(t_{i-2},t_{i-1},t_i)}{f(t_{i-2},t_{i-1})}$$

The tag-word pair probabilities and tag transition probabilities are commonly called as lexical probabilities, and transition probabilities, respectively. Lexical and transition probabilities must sum up to one to provide an axiomatic probability density function. We assume that each sentence in the corpus is affixed by a beginning-of-sentence symbol for the first order HMM and two beginning-of-sentence symbols for the first order HMM having one-step history in the state space. Next, in the calculation of MLEs, beginning-of-sentence symbol is treated as an ordinary word with the corresponding beginning-of-sentence tag. Afterward, sentence probabilities against all the possible tag sequences can be calculated by any suitable dynamic programming method. For instance, we used the widely accepted Viterbi algorithm [40].

The major problem of this model is assigning zero-probabilities to unknown words. This is a common situation on a new corpus tagging effort. If once, a zero probability is assigned for an unknown word-tag pair:

$$\hat{P}(w_i|t_i) = 0$$

All the sentence probabilities, which include this pair's tag, will have a zero sentence probability in the new corpus. Thus,

$$\hat{P}(w_i|t_i) \cdot \hat{P}(t_i|t_{i-2},t_{i-1}) = 0$$

This problem is valid for all language models based on a lexicon of closed vocabulary; several numbers of attacks have been made in the literature to solve this problem. Here, we give the original formulation of Kneser-Ney [34] smoothing method to preserve generality, although we have used a slightly modified version which is named as 'Kneser-Ney-Modified-fix' to smooth transitional probabilities.

2.2.2 Smoothing methods in our POS tagger. The transition probabilities may be formulated from the general form of the Kneser-Ney smoothing with interpolation as:

$$\hat{P}_{kn}(t_i|t_{i-2},t_{i-1}) = \frac{\max\{f(t_{i-2},t_{i-1},t_i) - D, 0\}}{\sum_{t_i} f(t_{i-2},t_{i-1},t_i)} + \frac{D}{\sum_{t_i} f(t_{i-2},t_{i-1},t_i)} \cdot N_{1+}(t_{i-2},t_{i-1},\bullet) \times \frac{N_{1+}(\bullet,t_{i-1},t_i)}{N_{1+}(\bullet,t_{i-1},\bullet)}$$

where

$$N_{1+}(t_{i-2},t_{i-1},\bullet) = \left|\{t_i \mid f(t_{i-2},t_{i-1},t_i) \geq 1\}\right|$$

$$N_{1+}(\bullet,t_{i-1},t_i) = \left|\{t_{i-2} \mid f(t_{i-2},t_{i-1},t_i) \geq 1\}\right|$$

$$N_{1+}(\bullet,t_{i-1},\bullet) = \left|\{(t_{i-2},t_i) \mid f(t_{i-2},t_{i-1},t_i) \geq 1\}\right|$$

and, the estimated value D is:

$$D = n_1/(n_1 + 2n_2)$$

where n_1 and n_2 are the total number of 3-grams with exactly one and two counts, respectively, in the training corpus.

Zero-probabilities problem regarding the unknown words is solved by using tag ratios obtained from training corpus that has 45,000 sentences as shown in table 1. In other words, we add all tag ratios to HMM process as word probabilities ($\hat{P}(w_i|t_i)$), whenever we see an unknown word.

Table1. Tag ratios obtained from training corpus.

Tag	Ratio	Tag	Ratio	Tag	Ratio
Noun	46,63%	Num	2,68%	NTime	0,04%
Punc	18,00%	Postp	1,73%	Dup	0,01%
Adj	9,63%	Pron	1,62%	NDet	0,004%
Verb	7,89%	Nnum	0,44%	NVerb	0,003%
Conj	3,96%	Ques	0,22%	NAdv	0,0001%
Adv	3,52%	NDot	0,16%	NAdj	0,0001%
Det	3,38%	Interj	0,05%		

2.2.3 Turkish. Turkish is an agglutinative and a free constituent order language, with high productive morphology, which has a rich set of derivational and inflectional suffixes. All part of speeches and derived morphological forms of words are constructed by suffixation(s) from a noun root or a verb root. Hankamer [25] states that the number of different word forms, one can derive from just a particular Turkish lexeme, may theoretically be in the millions. Although, this is not the ordinary case in practice, it is true that a lexicon may rapidly exceed the limit out of a manageable size, if all different forms of words are simply added to the list without any pre-processing.

The number of minor part of speech categories for Turkish are much richer than any analytical language likes English. For the computational linguistics' perspective, in order to disambiguate a word form's final part of speech, it is crucially needed that all possibilities of every intermediate derivational and inflectional production to be marked. In effect, intermediate temporary productions eventually determine the set of appropriate suffixes, which in turn, determine the final part of speech category. For example, for the Turkish word *Türkçeleştirme* has two different possible morphological parses as seen in Figure 1. In the first parse, there are two intermediate derivations, where the first one drives the active stem, *Türkçe,* (*/Turkish language*) from the root *Türk* (*/Turkish (people)*) with suffix *çe*, and a final derivation, labeled as two in the figure, to the major part of speech *Verb*. Other labels mark the minor parts of the corresponding major part of speech. In the second parse, the final derivation runs the stem to the major part of speech *Noun.* Readers may refer to work of Oflazer [36] for further morphological considerations of Turkish.

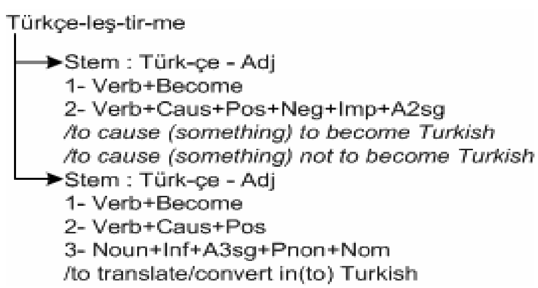

Figure 1. Morphological parses of *Türkçeleştirme***.**

3. A Suffix Based POS Tagger

3.1 Methodology

The computational complexity of the HMM model increases exponentially with higher history depths. The first order Markov model with one-step history in the state space is common in practice. Therefore, we considered only the two forms of HMM model: first order HMM without History, and first order HMM with one-step history in state space.

In our formalism, we ensure the closed vocabulary property by constructing our lexicon from the unity of word endings of selected fixed length characters as our new lexemes (Afterward, we will use the term *l*-lexeme to denote the lexemes constructed by taking *l* character from the word form ends, and term *l*-lexicon to denote lexicon of *l*-lexemes). Therefore, we only need to replace the actual word forms in the original Markov model for PoS tagging with the specific word endings. We have considered seven different number of character lengths, $l = 1,2,..,7$ in this study which are all tested on 30 different sizes of training corpus measured in number of sentences.

We used a set of 19 tags: 11 major parts of speeches (*Noun*, *Verb*, *Det*, *Adj*, *Adv*, *Conj*, *Pron*, etc.); 7 extra parts of speeches for the derivations from numeric tokens (*Num* for number, *NNum* for noun from number, *NVerb* for verb from number, *NDet* for determinant from number, *NAdj* for adjective from number, *NAdv* for adverb from number, *NTime* for noun from time); 1 (*NDot*)for abbreviations.

We evaluate our test runs with the word success ratios. The word success ratio is calculated as:

$$W_S = \frac{\#\ \text{of correctly assigned PoS}}{\#\ \text{of tokens in test corpus}}$$

3.2 Corpora

Bilkent corpus which is a collection of Turkish news texts having about 650,000 tokens and 48,000 sentences is used as the training corpus. It is morphologically analyzed and disambiguated automatically by Hakkani-Tür et al. [26] in Bilkent University. Test corpus is selected from METU-Sabanci Turkish treebank named as OSTAD. OSTAD corpus is a subset of METU(Middle East Technical University) corpus that has been released for the academic purposes. OSTAD is morphologically analyzed by hand which ensures its high percentage of correctness. It consist total of 51,209 tokens (including just letter(s) of alphabet) and about 7,200 sentences. Properties of selected training and test corpus are given in Table 2.

Table 2. Properties of Turkish corpora.

Corpora	Tokens	Vocabulary	Sentence Count
Training[1]	655,720	90,655	45,000
Test	1078	598	100

The corresponding reduced vocabulary sizes of the seven *l-lexeme* for the 45000 sentences training corpus are 37, 589, 12993, 26629, 41489, and 56031, respectively.

3.3 Results

We have tested our corpora for the two models of Markov Model: first order Markov Model and first order Markov Model with one-step history in the state space. We have run our experiments for each seven *l*-lexemes from word endings for 30 different training sizes for each of two Markov models. The results of the first case where each *l*-lexicon is tested on 30 different training sizes on first order Markov Model are given in Figure 2.

[1] All statistics for training corpus are given over the largest size of 45,000 sentences.

The best case of the first experiment is obtained with 5-lexeme on the training size of 45,000 sentences at the accuracy of 88.9%.

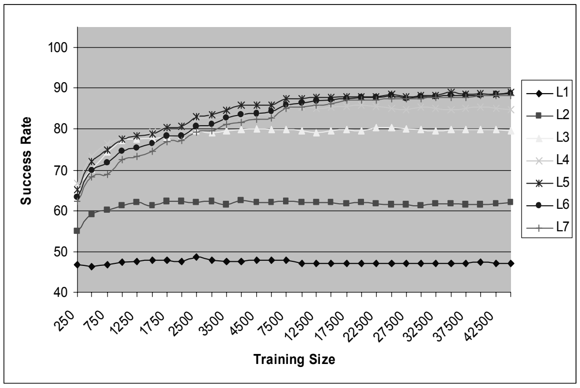

Figure 2. Results of 1st order HMM for *l*-lexicon vs training sizes.

The results of the second experiment where again each *l*-lexicon is tested on 30 different training sizes but on the first order Markov Model with one-step history in the state space are given in Figure 3.

The best case of the second experiment is obtained with 5-lexeme on training size of 45000 sentences at the accuracy of 90.2%.

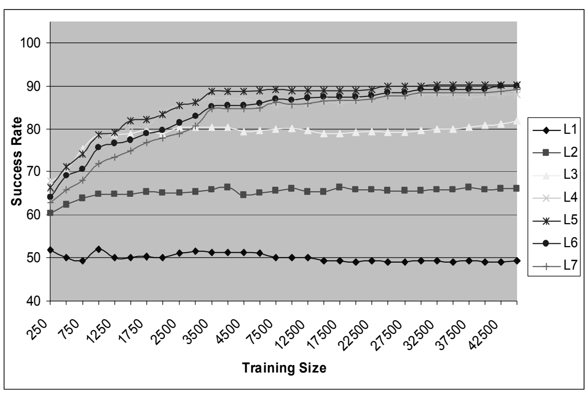

Figure 3. Results of 1st order HMM with one-step history for *l*-lexicon vs training sizes.

We repeated the above experiments for different formation of closed lexicon. To see if it would make any difference if we get the lexemes from the word beginnings rather than the endings, seven *l*-lexeme from word beginnings on 30 different training corpora are tested. The best case is obtained on 1st order HMM and 1st order HMM with one-step history with 5-lexeme from word beginnings at the accuracy 85.4% and 86.9% respectively. In another experiment, we have used last syllable from word endings. This experiment resulted in 72.9% and 78.2% success rates for 1st order HMM and 1st order HMM with one-step history respectively.

4. Conclusion

Our results show that a PoS tagging task with lexicon of closed vocabulary is possible for an agglutinative language as in Turkish by the use of simply suffixes, thus far, more research is required to find the more generalized construction form of the word endings as the special unities. In this direction, one of the important future works is to cluster word endings into the linguistic suffix groups defined for Turkish. For instance, *dık ,dik duk, dük, tık, tik, tuk, and tük* are actually the realized form of a single suffix and must be considered as a unity by the means of a meta-suffix. This will probably accelerate convergence of output probabilities (i.e. $\hat{P}(w_i|t_i)$) at moderate training sizes, and also contribute to the model accuracy.

5. References

[1] Adriani, M. van Rijsbergen , C. J. *Phrase identification in cross-language information retrieval.* In RIAO, 2000.

[2] Aduriz, I. Alegria, J.M. Arriola, X. Artola, A. Diaz de Illarraza, N. Ezeize, K, Gojenola, and M. Maritxalar. Different issues in design of a lemmatizer/tagger for Basque. In *SIGDAT-95 (EACL-95 workshop),* 1995.

[3] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. 1st ed. Addison-Wesley, England, 1999.

[4] Baker, J. K. Stochastic modeling for automatic speech understanding. In D. Raj Reddy (eds.), *Speech Recognition: Invited papers presented at the 1974 IEEE symposium,* New York: Academic Press. 1975, 297-307.

[5] Brants, T. Samuelsson, C. Tagging the Teleman Corpus. In *Proceedings of the tenth Nordic Conference of Computational Linguistics,* 1995.

[6] Brants, T., TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference*, ANLP-2000, Seattle, WA, 2000 (April 29-May 3).

[7] Brill, E. Transformation-based error driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics, 21(4)*, 1995 (December), 543-566.

[8] Brill, E. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the Third Workshop on Very Large Corpora,* Cambridge, MA, 1995 (June).

[9] Broglio J, Callan JP, Croft WB. INQUERY system overview. In *Proceedings of the TIPSTER Text Program (Phase I)*. San Francisco, CA Morgan Kaufmann, 1994, pp. 47-67.

[10] Carlberger, J. and Kann, V. Implementing an efficient part-of-speech tagger. *Software Practice and Experience,* 1999.

[11] Chanod, J.P. Tapanainen, P. Tagging French – comparing a statistical and constrained-based method. In *EACL-95,* 1995.

[12] Chomsky, N. *Syntactic Structures.* The Hague: Mouton, 1957, chapter-3.
[13] Church, K.W. A stochastic parts program and a noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing,* Austin, Texas, 1988.
[14] Church, K. A stochastic parts program and noun phrase parser for un-restricted texts. In *Proceedings of the Second Conference on Applied Natural Processing*, Austin, Texas, USA (1988).
[15] Cussens, J., Dzeroski, S., & Erjavec, T. Morphosyntactic tagging of Slovene using Progol. *Inductive Logic Programming: Proc. of the 9th International Workshop (ILP-99)* Bled, Slovenia. Springer-Verlag, 1999.
[16] Dermatas, E. Kokkinakis, G. Automatic stochastic tagging of natural language texts. *Computational Linguistics,* 21(2), 1995, 137-164.
[17] DeRose, S. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, (1988).
[18] Derouault, Anne-Marie., Merialdo, B. Natural language modeling for phoneme-to-text transcriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (8)*, 1986, 649-742.
[19] Feldweg, H. Implementation and evaluation of a German HMM for POS disambiguation. In *EACL SIGDAT Workshop,* 1995.
[20] Garside, R. and Leech, F. The UCREL probabilistic parsing System. *The Computational Analysis of English: A Corpus-Based Approach*, Longman, London, 1987, 66-81.
[21] Garside, R. The CLAWS word-tagging system. In Garside R., Leech, F., and Sampson, G., editors, *The Computational Analysis of English.* Longman, (1987).
[22] Greene, B. B. Rubin, M. G. *Automated grammatical tagging of English*. Department of Linguistics, Brown University, (1971).
[23] Gerard, S. and Thorpe, R. W. An approach to the segmentation problem in speech analysis and language translation. In *Proceedings of the International Conference on Machine Translation of Languages and Applied Language Analysis,* volume 2, London, 1962, 703-724.
[24] Hankamer, J. Turkish generative morphology and morphological parsing. *In: Second International Conference on Turkish Linguistics*. Istanbul, Turkey, 1984.
[25] Hankamer, J. *Lexical representations and Process*, Morphological Parsing and the Lexicon (chapter). The MIT press, 1989.
[26] Hakkani-Tür, D.Z., Oflazer, K. and Tür, G. Statistical Morphological Disambiguation for Agglutinative Languages. *Computers and the Humanities*, 2002.
[27] Jelenik, F. Markov source modeling of text generation. *The Impact of Processing Techniques on Communications,* volume E91 of NATO ASI series, Dordrecht: M. Nijhoff, 1985, 569-598.
[28] Klein, S. and Simpson, R. A computational approach to grammatical coding of English words. *JACM (10)*, 1963, 344-337.
[29] Levinger, M. Ornan U. Itai, A. Learning morpho-lexical probabilities from an untagged corpus with and application to Hebrew.*Computational Linguistics, 21(3),* 1995, 383-404.
[30] Manning, C.D. and Schütze, H. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, London, England, 2003.
[31] Markov, Andrei, A. An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of 'tests' in chains. In *Proceedings of the Academy of Sciences, St. Petersburg,* volume 7 of VI, 1913, 153-162.
[32] Marshall, I. Tag selection using probabilistic methods. *The Computational Analysis of English: a copus-based approach,* Longman, London, 1987, 42-65.
[33] Mercer, R., L. Inflectional morphology needs to be authenticated by hand. In *Working Notes of AAAI Spring Symposium on Building Lexicons for Machine Translation.* Stanford, CA, AAAI Press, 1993, 99-99.
[34] Ney, H. Essen, U. and Kneser, R. On structuring probabilistic dependencies in stochastic language modeling. *Computer, Speech and Language,* 8, 1994, pp.1-38.
[35] Oflazer, K. Tür, G. Morphological disambiguation by voting constraints. In *Proceedings of the thirty-fifth Annual Meeting of the Association for Computational Linguistics (ACL'97/EACL'97),* Madrid, Spain, 1997 (July).
[36] Oflazer, K. Two-level description of Turkish morphology. *Literary and Linguistic Computing, 9(2)*, 1994, pp.137-148.
[37] Sánchez León, F. Nieto Serrano, A.F. Development of a Spanish version of the Xerox tagger. CRATER/WP6/FR1, 1995.
[38] Stolz, W. S., Tannenbaum P. H., and Carstensen, F. V. A stochastic approach to the grammatical coding of English. *Communications of ACM (8),* 1965, 399-405.
[39] Taylor, H.M. Karlin, S. *An Introduction to Stochastic Modeling.*Academic Press, Inc., London NW1, 7DX, 1984.
[40] Viterbi, A.J. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Transaction on Information Theory*, 1967, pp. 260-269.