International Journal of Computational Intelligence Systems, Vol.3, No. 5 (October, 2010), 632-645

DOCUMENT CATEGORIZATION WITH MODIFIED STATISTICAL LANGUAGE MODELS FOR AGGLUTINATIVE LANGUAGES

Ahmet Cüneyd TANTUĞ
Computer Engineering Department, Istanbul Technical University
İstanbul Teknik Üniversitesi Ayazaga Yerleskesi Elektrik-Elektronik Fakültesi, Maslak
Istanbul, 34469, Turkey
E-mail : tantug@itu.edu.tr
www.itu.edu.tr

Received: 16-10-2009
Accepted: 25-06-2010

Abstract

In this paper, we investigate the document categorization task with statistical language models. Our study mainly focuses on categorization of documents in agglutinative languages. Due to the productive morphology of agglutinative languages, the number of word forms encountered in naturally occurring text is very large. From the language modeling perspective, a large vocabulary results in serious data sparseness problems. In order to cope with this drawback, previous studies in various application areas suggest modified language models based on different morphological units. It is reported that performance improvements can be achieved with these modified language models. In our document categorization experiments, we use standard word form based language models as well as other modified language models based on root words, root words and part-of-speech information, truncated word forms and character sequences. Additionally, to find an optimum parameter set, multiple tests are carried out with different language model orders and smoothing methods. Similar to previous studies on other tasks, our experimental results on categorization of Turkish documents reveal that applying linguistic preprocessing steps for language modeling provides improvements over standard language models to some extent. However, it is also observed that similar level of performance improvements can also be acquired by simpler character level or truncated word form models which are language independent.

Keywords: document categorization, statistical language modeling, n-gram, Turkish

1. Introduction

During past decades, the proliferation of documents accessible in digital form boosts the need for information retrieval related tasks. One of the challenging information retrieval problems is document categorization which is basically the task of assigning a document to one or more of the predefined set of categories. The document categorization term sometimes referred as text/document classification[a].

Document classification has found many application areas such as document filtering (e.g. spam filtering), document indexing based on a controlled vocabulary, document organization, identification of topic or language, and word sense disambiguation. Also some applications combine document classification and other methods. For example, speech can be categorized by

[a] We use document/text classification and document/text categorization terms interchangeably throughout this paper.

means of text categorization performed after speech recognition. Similarly, multi-media document categorization can be done by analyzing the textual captions in the document.

Although the history of document classification goes back to 1960s, the field gained its outstanding status in 80s and 90s. While automatic document classification systems of these years were generally rule based systems built by knowledge and domain experts, a shift occurred towards machine learning based approaches in the beginning of 90s. The state of the art document classification techniques rely on both machine learning and information retrieval paradigms. Generally, automatic document classification systems are implemented by supervised machine learning techniques where a set of hand-labeled documents must be supplied for teaching algorithms how to classify a new document. Some of the most popular techniques are Naïve Bayes classifiers, decision trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN), multi-variate regression models and k-nearest neighborhood classifiers (kNN). Sebastiani[1] gives a comprehensive review of automatic document categorization. Naïve Bayes technique is frequently applied in classification tasks because of its simplicity and relatively high accuracy. Despite of its oversimplified assumptions, it is shown that Naïve Bayes can achieve superior results among others[2-4]. Briefly, Naïve Bayes assumes that each term (word) is independent from others, which means the document is processed as a bag-of-words where the order of the words is not taken into account. If the contexts of the words are incorporated into the classification process, Naïve Bayes functions very similar to n-gram models. This similarity emerged the recent idea of using n-gram based statistical language models in text categorization[5].

Even though statistical language models was first used by the speech recognition community[6], a number of various applications like information retrieval[7], machine translation[8], part-of-speech tagging[9] and parsing[10] benefit from the advantages of statistical language models. Language models have the ability to assign a concrete probability to a sequence of utterances such as words, letters or syllables. This basic language modeling property appeal researchers from many fields, even from bioinformatics[11].

Choosing statistical language models as a classifier presents some advantages for document classification. As all machine learning algorithms need a formal representation of the samples (*documents* in this context), a feature selection phase is essential before classification. In this phase, according to their representational and discriminative properties, some terms are selected to build a feature space on which the classification algorithms can work. This stage is an important initial step which can affect not only the learning process but also the efficiency of classification. Unfortunately, feature selection process often depends on the task and the language. A number of pre-processing efforts like stop-word removal and stemming are language dependent. Furthermore, different applications require different choice of features; for example text genre identification tasks look for linguistic patterns whereas topic spotting tasks generally work with bag-of-words. Statistical language models eliminate the feature selection process since the features are selected implicitly. Additionally, a wide number of researches on advanced smoothing methods provide eligibility for language models over traditional methods in text classification.

The main motivation of our work is investigating and improving the performance of statistical language model based document categorization for agglutinative languages. Even though there exists an extensive literature on text classification, the exploitation of statistical language models in text classification is a recent research topic[5]. In this study, Peng et al. obtained the state of the art results with Naïve Bayes classifiers augmented with statistical language models. Owing to the vocabulary explosion in agglutinative languages, statistical language models for these languages suffer from data sparseness problem. In previous works on other application areas excluding text classification, various language model modifications are offered for better modeling of the agglutinative structure, and considerable improvements are achieved. We aim to evaluate and improve the performance of the text classification in agglutinative languages with modified language models proposed in prior studies. Also, the effect of the document length on classification success is evaluated in the scope of our work.

The rest of the paper is organized as follows. Section 2 gives brief information about related works. Section 3 describes the basics of statistical language models while Section 4 gives the details of statistical language model based text categorization. Section 5

introduces problematic issues of language modeling for agglutinative languages. The modifications to language models are explained in Section 6. The experiments and related results are presented in Section 7. Finally, we evaluate the results and draw some conclusions in Section 8.

2. Related Work

For text classification, n-grams are used in many ways. The most common approach is considering n-gram phrases as terms during feature selection process prior to classification[12-14]. On the other hand, several studies can be found dealing with the text classification based on prediction by partial matching (PPM) based language models[15, 16] andtopic models[17]. Text categorization with letter based PPM method performs better than word based Naïve Bayes classifiers, and even close results to linear support vector machine text classifiers[18]. But, the implementation of statistical language models in its traditional fashion, which is assigning a probability to a sequence of words, is first applied by augmenting the Naïve Bayes classifier with statistical language models[19]. Their statistical language model based on chain augmented Naïve Bayes classifiers (CAN) are applied on three different text classification problems (topic detection, authorship determination and text genre classification) in four different languages (English, Greek, Chinese and Japanese). Although some data sets are limited to draw a generalization, an overall assessment states that they get state-of-the-art results and even better.

The sophisticated morphosyntactic structures of agglutinative languages led researchers to take some preprocessing steps in document classification tasks. The most common step for dimensionality reduction is word form normalization. It is shown that lemmatization or stemming can result in considerable performance increments in information retrieval tasks for many agglutinative languages such as Finnish[20], Hungarian[21, 22] and Basque[23]. Also being an agglutinative language, numerous similar studies on Turkish has been done. For different applications, various methods are used to perform document classification on Turkish texts. For example, Naïve Bayes, support vector machines, decision trees are used to identify the authors of the Turkish documents[24]. Another work concentrates on spam filtering task on Turkish e-mails employing artificial neural networks[25, 26]. There exists two studies with the focus on classification of Turkish news specifically[27, 28]. In the former work, kNN and a time-efficient improvement of kNN, feature projection text classification, techniques are used in the classification of a dataset consisting of 20K news articles. The latter research involves a comparative performance evaluation of Naïve Bayes and artificial neural network classifiers on a dataset which consists only 50 news articles. Cataltepe et al.[29] analyze the effect of stemming as a pre-processing step when centroid classification algorithm and support vector classifier are applied in Turkish document classification tasks. They conclude that roots consisting only consonants can achieve the highest performance for the cases where the representational term vector cardinality should be small.

3. Statistical Language Models

Statistical language models define probability distributions on word sequences. By using a language model, one can compute the probability of a sentence *S* $(w_1 w_2 ... w_k = w_1^k)$ by the following formula:

$$\begin{aligned} P(S) &= P(w_1)P(w_2|w_1)...P(w_k|w_1...w_{k-1}) \\ &= \prod_{i=1}^{K} P(w_i|w_1^{i-1}). \end{aligned} \quad (1)$$

This means that the probability of any word sequence can be calculated by decomposition using the chain rule, but usually due to sparseness, most terms above would be zero, therefore n-gram approximations are used. N-gram models predict the probability of a word from the previous *N-1* words by using Markov assumption.

$$P(S) = \prod_{i=1}^{K} P(w_i|w_{i-N+1}^{i-1}). \quad (2)$$

The most intuitive way to estimate the n-gram probabilities is maximum likelihood estimation which is simply counting word occurrences in the corpus and calculating the relative frequency:

$$P(w_i|w_{i-N+1}^{i-1}) = \frac{\text{Count}(w_{i-N+1}^{i-1} w_i)}{\text{Count}(w_{i-N+1}^{i-1})} = \frac{\text{Count}(w_{i-N+1}^{i})}{\text{Count}(w_{i-N+1}^{i-1})}. \quad (3)$$

However, some of the word sequences have zero counts because every corpus is limited and cannot contain all legal word sequences in the language. Also,

zero probabilities assigned to unseen word sequences in the corpus cause any sentence *S* containing even one of those unseen word sequences get *P(S)=0* for the whole sentence. For these reasons, some of the probability mass from observed word sequences is distributed over zero counts, which is called *smoothing*. A good comparison of simple and advanced smoothing techniques can be found in[30, 31]. Commonly used smoothing techniques are absolute smoothing[32], Laplace smoothing[33-35], Witten-Bell smoothing[36], Good-Turing smoothing[37, 38] and Kneser-Ney smoothing[32].

The correct way of evaluating the performance of a language model is evaluating the total performance of the application after embedding it in the application[39]. Employing such an evaluation scheme can be burdensome for many applications, so perplexity is frequently used as a language model evaluation measure. However, the results in our work are presented by means of document categorization performance scores instead of representing the perplexities of the related language models.

4. Text Categorization with Statistical Language Models

Text categorization can be defined as assigning a document d_j to the category c_i from the set C, where $c_i \in C=\{c_1,c_2,\ldots, c_N\}$. In order to apply supervised machine learning techniques to build an automatic classifier, a set of documents must be provided with their correct categories. Documents in this *training set* are represented in the form (dj, c_i), where d_j represents the j-th document in the collection $D=\{d_1,d_2,\ldots,d_M\}$ and c_i is the category of the document d_j. Finally, a *model* $\boldsymbol{M:D \rightarrow C}$ that maps documents to classes must be generated by using a *training procedure*. It is very common to use probabilistic models focused on finding the class $\hat{c}$ that maximizes the probability $P(c_i|d_j)$, which can be re-written as in Eq. (4) by the help of Bayes rule:

$$P(c_i \mid d_j) = \frac{P(d_j \mid c_i) \times P(c_i)}{P(d_j)}. \qquad (4)$$

Since the probability $P(d_j)$ is constant for a specific document d_j, the probability of $P(d_j)$ can be ignored and the most probable class $\hat{c}$ for the document d_j can be calculated as in Eq. (5).

$$\begin{aligned} \hat{c} &= \arg\max_{c_i \in C} P(c_i \mid d_j) \\ &= \arg\max_{c_i \in C} \frac{P(d_j \mid c_i) \times P(c_i)}{P(d_j)} \\ &= \arg\max_{c_i \in C} P(d_j \mid c_i) \times P(c_i). \end{aligned} \qquad (5)$$

The prior class probabilities $P(c_i)$, where $i=1,2,\ldots,N$, are generally calculated straight forward by using maximum likelihood method. Different training procedures try to estimate $P(d_j|c_i)$, the likelihood of document d_j for category c_i, in various ways. Statistical language models can also be used in calculation of this conditional probability distribution. For each class c_i, a separate language model is generated by using training documents in category c_i. Each language model LM_i can compute the likelihood of a new document d_j for category c_i, and *argmax* function searches for the most probable class $\hat{c}$ according to formula in Eq. (6). In other words, each class c_i has its own language model which calculates how much the language of a new document resembles the language usage in this category.

$$\begin{aligned} \hat{c} &= \operatorname*{argmax}_{c_i \in C} P(d_j \mid c_i) \times P(c_i) \\ &= \operatorname*{argmax}_{c_i \in C} P(w_{j1} w_{j2} \ldots w_{j|d_j|} \mid LM_i) \times P(c_i). \end{aligned} \qquad (6)$$

The choice of statistical language models for estimating $P(d_j|c_i)$ presents a number of advantages against other ML techniques. The elimination of feature engineering is the most apparent one. Typically, vector space model is used as data representation where each document is represented by a high-dimensional vector of word counts or binary flags for the existence of the corresponding words in that document[40]. The selection of the words constituting the representation space is implemented by an external effort which may be subjective. On the other hand, statistical language models calculate the contribution of each unit (word or n-gram) to the model and perform selection according to the importance of the words implicitly.

Classification with unigram language models using Laplace smoothing functions almost the same as Naïve Bayes classification[5]. However, choosing the language model order higher than one increases the ability to

model longer contexts which capture the discriminative properties of word order. Additionally, statistical language models can benefit from more efficient smoothing methods for unseen words (or word sequences) as described in previous section. Chen and Goodman[31] points out that the performance of Laplace smoothing is outperformed by many advanced smoothing methods. In a recent work about Naïve Bayes based text classification, it is shown that improved smoothing techniques yields better and more stable performance than Laplace smoothing[41].

5. Issues with Statistical Language Modeling for Agglutinative Languages

In agglutinative languages such as Turkish, Finnish, Hungarian and Estonian, words are formed by concatenation of morphemes extensively. Due to the rich morphological structure, the number of word forms that can be encountered in natural text is very large than other languages. For instance, it is reported that in a 10M word corpora of English and Turkish, the number of distinct English word forms is 97,734 whereas the number of Turkish word forms is 417,775 [9]. The productive derivational and inflectional suffixations of Turkish allows generation of more than one million legitimate word forms from only one Turkish root word[42]. Even in some cases, just one word form may convey the equivalent semantic information of a whole English phrase, clause or sentence. Below is a popular (but exaggerated) example which demonstrates the complex morphological process of Turkish[43]:

uygarlaştıramayabileceklerimizdenmişsinizcesine

uygar+laş+tır+ama+yabil+ecek+ler+imiz+den+miş+siniz+cesine

"(behaving) as if you were one of those whom we might not be able to civilize"

As another example, the list in Table 1 shows the word forms of the root word "*futbol*" (football) encountered in the corpora, along with their unigram counts[b]. It is noted that a single Turkish verb root can have around 40,000 forms excluding the derivational suffixes[44]. Similarly, a Finnish verb may have 12,000 forms whereas a Finnish noun may have 2,000 forms because of the highly inflected Finnish morphology[20].

[b] Only first 10 of the total 42 word forms are included in the table because of space limitations.

Table 1. Some example data set observations of the word forms whose root are "*futbol*"

Word forms	Count	English Translation
futbol	59	football
futbola	2	to the football
futbolcu	105	football player
futbolcudan	1	from the football player
futbolcular	39	football players
futbolculara	5	to the football players
futbolculardа	1	at the football players
futbolculardan	10	from the football players
futbolcularla	1	with the football players
futbolcularımız	1	our football players

From the point of view of statistical language modeling, the large number of distinct word forms, i.e. vocabulary size, causes significant data sparseness problems. The large vocabulary size not only necessitates larger training sets for better statistical modeling but also lead to estimate much more parameters even for small order language models.

6. Language Model Modifications

In order to alleviate the data sparseness drawback posed by morphology, a number of studies propose several modifications over language modeling. Although the application areas of these studies are varying, they all achieve to have remarkable levels of improvements when they incorporate the underlying morphology in the model. Arısoy et al.[45] investigate alternative language modeling units like "stems and endings", "stems and morphemes", and "syllables", instead of "words" in speech recognition tasks. A recent work which splits words into their stems and suffix components results in a significant perplexity reduction in Turkish language modeling[46]. Since some morphemes carry long distance dependencies in Turkish sentences, language models based on units comprising word stem and its last inflectional morpheme group[9, 47], yield better results than word form based models in machine translation[48].

Similarly, information retrieval and indexing tasks also suffer from the unlimited vocabulary properties of agglutinative languages. The problem is generally tackled by stemming approach[49-51].

One of the targets in this work is the classification performance evaluation of the modified language models proposed for different tasks in previous studies. The basic model is classifying texts with statistical

language models based on word forms where standard language models are trained without any language dependent information. This model will construct a baseline for future evaluations of classification performance with different language model types.

Our next language model type utilizes only root words in training, similar to previous efforts[45, 46]. In most of the languages, roots of the word forms are obtained by basic suffix stripping stemmers such as Porter Stemmer for English[52]. However, having root words from word forms is a non-trivial task for Turkish. A morphological analyzer should be used to get root words as well as other morphological features. Moreover, morphological disambiguation must be applied as the next step since roughly 50% of the morphological analyzer output is ambiguous. We have used a two-level morphological analyzer[53] and a statistical morphological disambiguation tool based on[9]. In our dataset, the total number of distinct word forms is 68,420, whereas the total number of distinct root words is 23,739 after morphological analysis and disambiguation. This means approximately 65% reduction in the vocabulary size of the language models.

The part-of-speech tag may help discriminating some synonymous word roots (e.g. `ara+Verb` (*to search*) and `ara+Noun` (*distance/space*)) in some cases [54]. So, another type of language model, which is based on part-of-speech (POS) tags being attached to root words, is included in our evaluation scheme.

Language model modifications mentioned above require language dependent tools. Thus, language independency advantage, being one of the most useful properties of language model based document classification method, vanishes. To mitigate this disadvantage, language independent methods are investigated for getting root words from word forms. So, another type of language model is suggested based on the fact that average root word length is 4.03 letters in Turkish[55]. This approach is basically inspired from an information retrieval oriented work[51]. According to this *FirstFLetters* model, first *F* (*F*=3,..,7) letters are truncated from the beginning of every word form so that a convergence can be maintained to the root word based models by evading any language specific process. Recently, it has been shown that using truncated words in indexing rather than the actual root words produced by a sophisticated stemmer simply improves the system effectiveness of search engines[51]. A similar truncating approach is also suggested for rapid and feasible Turkish information retrieval system[50].

Lastly, character level language models are incorporated into our study. This type of language model uses character sequences as the base unit instead of words. Although no study on Turkish language modeling applies character based fashion, it has been shown that character level language model classifiers are able to acquire high levels of accuracy in English, Japanese and Chinese texts[5]. Particularly, this approach is well suited for the languages suffer from word segmentation problems, such as Chinese and Japanese. Nevertheless, it has been shown that even for Western languages like English and Greek, character level language model classification can also perform higher classification performance than word level models[19]. With this motivation, character level language model types are also participated in our tests. It is noteworthy to point out that our previous model which just truncates the first F character of the word form is completely different with the character level models. For example, the first 3 character trigram language model run on first 3 character truncated version of the word forms, but it still operates on word level. However, a character level trigram language model should be thought as a 3 character width sliding window moving on all characters constituting the word forms (also whitespaces).

Table 2 shows base unit examples of a trigram sequence for the language models mentioned in this section.

7. Experiments

7.1. *Data set*

We have generated a new data set from scratch by processing the news broadcasted by Anadolu Agency, the national news agency of Turkey[c]. A similar dataset from the same source was used in a previous work[27], but the dataset is not publicly available. The agency provides news in eight different categories listed in Table 3.

The dataset is composed of 20,000 downloaded documents that are evenly distributed among categories. These files are cleaned up from HTML tags and parsed to get useful information on the page. The final data set

[c] http://www.aa.com.tr

Table 2. Base unit examples for language models

Type	Base Unit Example
Word form	… uçağında yaptığı açıklamada …
Root	… uçak yap açıkla …
Root+POS	… uçak+Noun yap+Verb açıkla+Verb …
First3Letter	… uça yap açı …
First4Letter	… uçağ yapı açık …
First5Letter	… uçağı yapıl açıkl …
First6Letter	… uçağın yapıla açıkla …
Character level	…u ç a ğ ı n d a # y a p t ı ğ ı # a ç ı k l a m a d a …

contains 2,500 news documents per category, each stored in separate XML files[d]. Sample file content is given in Fig. 1.

Table 3. News categories in data set

Code	Category
1	Turkey News
2	World News
3	Politics
4	Economics
5	Sports
6	Education and Science
7	Culture and Art
8	Environment and Health

Although some meta data such as broadcast date and time are available for our documents, the classification is accomplished only on the basis of endogenous knowledge which means the knowledge extracted from the documents.

```
<id>100043</id>
<topic>07-CultureArt</topic>
<date>31.03.2007</date>
<time>20:48:00</time>
<title>Başkentte "İz Resim Ve Heykel
Sergisi"</title>
<text>Başkentte, 15 sanatçının 200 eserinin
yer aldığı, "İZ resim ve heykel sergisi"
açıldı.</text>
```

Fig. 1. Sample File Content

Table 4 presents some statistical properties of the data set.

[d] The compiled corpus can be downloaded from http://ddi.ce.itu.edu.tr

Table 4. Some statistical properties of the data set

Total Number of	
documents	20,000
documents per category	2,500
sentences	31,381
sentences per document	1.56
tokens	671,819
tokens per document	33.59
tokens per sentence	21.40

7.2. *Experimental results*

In order to have a fair evaluation, 10-fold cross-validation technique is used where the complete dataset is randomly divided in 10 parts. The usual train and test processes run 10 times; at each step, 9 parts are used for training and the testing is done on the remaining fold. SRILM toolkit[56] is used for training language models. We have conducted experiments with different language model orders (n=1,…,5) and smoothing methods to find an optimum parameter set. To maintain the comparability of our results with previous studies on classification with statistical language models, the classification results are given in terms of F_1 score, which is the harmonic mean of precision and recall. However, please note that F_1 is not a golden metric for all kinds of applications since the importance of recall and precision rates are not equal for various applications. For instance, spam filtering is a text classification task in which mistakenly classifying a legitimate mail as spam is a much more severe error than classifying a spam mail as legitimate. In this case, it would not be an appropriate solution to optimize an algorithm by using F_1 metric where recall and precision rates have equal weights.

The multi-class classification performance is represented by macro F_1 measure computed by averaging the F_1 score over individual confusion

Table 5. Results for word form based classification

	Laplace (LP)		Kneser-Ney (KN)		Witten-Bell (WB)		Absolute (AB)		Good-Turing (GT)	
n	**macroF_1**	**σ**	**macroF_1**	**σ**	**macroF_1**	**σ**	**macroF_1**	**σ**	**macroF_1**	**σ**
1	**0.7268**	**0.0060**	0.5821	0.0066	0.7160	0.0164	0.7452	0.0189	0.7719	0.0066
2	0.7243	0.0054	0.6827	0.0074	0.7275	0.0122	0.7451	0.0139	0.7855	0.0074
3	0.7161	0.0065	0.6963	0.0076	0.7284	0.0131	0.7434	0.0140	0.7852	0.0076
4	0.7117	0.0068	0.7006	0.0077	0.7284	0.0140	0.7433	0.0140	0.7853	0.0077
5	0.7099	0.0067	0.7015	0.0078	0.7288	0.0136	0.7432	0.0137	0.7855	0.0078

matrices from each category[1]. Table 5 shows results for the baseline (word form) system. The classification performances with each smoothing method are represented in two columns; average macro F_1 scores are given in the first column and standard deviations are shown in the next column.

Please note that, the performance of the unigram (*n=1*) language model using Laplace smoothing (denoted by the bold characters in the table) is approximately[e] the Naïve Bayes classifier performance. For a better visual presentation, results are depicted in Fig. 2. Although the best classification performance is obtained by the language models with Good-Turing smoothing and *n=3,4,5*; it will be reasonable to use trigram (*n=3*) models instead of higher order language models for the sake of simplicity. From the figure, apart from Kneser-Ney smoothing where *n=1*, the Laplace smoothing is outperformed by all other smoothing methods considered in this work. Furthermore, having larger context (*n>1*) slightly contributes to the performance.

Even with this baseline system, it can be said that n-gram based classification is able to achieve significant progress over Naïve Bayes method with the advantage of better smoothing methods and ability to process longer regularities in the text.

In a previous news classification effort on a very small Turkish news data set including only 25 documents, it is reported that Naïve Bayes achieves 0.76 accuracy[28]. However, in our study, which is carried out on a relatively larger data set, the Naïve Bayes performance is computed as 0.7268.

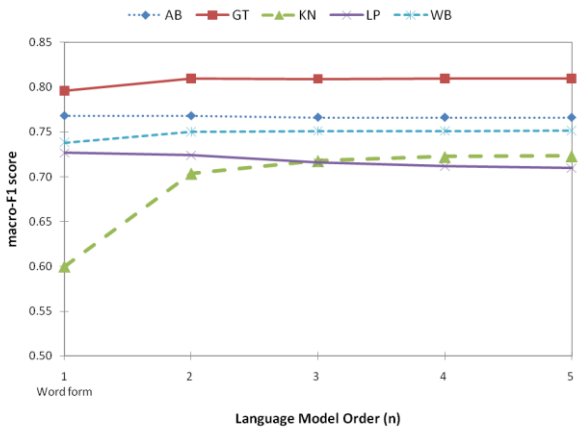

Fig. 2. Performance of word form based language models with different smoothing methods along with language model order

On the other hand, from the point of view of performance comparison between Naïve Bayes classifiers in Turkish and other languages like English, results on Turkish word form based text categorization shows that it fails to reach the accuracy on English. Peng et al.[5] states that their unigram n-gram classifier with laplace smoothing has 0.8493 accuracy on classification of English 20 newsgroup data set containing 19,974 documents. This important performance difference (approximately 14%) reflects the negative effects of the large vocabulary size in Turkish. However, the gap between Turkish and English classification performance shrinks to some extent when high order n-grams with advanced smoothing methods are employed. The best performing word-level F_1-score on English is 0.8822 [5] whereas it is calculated as 0.7852 on Turkish side.

[e] Because the feature selection process is not automatic and subject to change

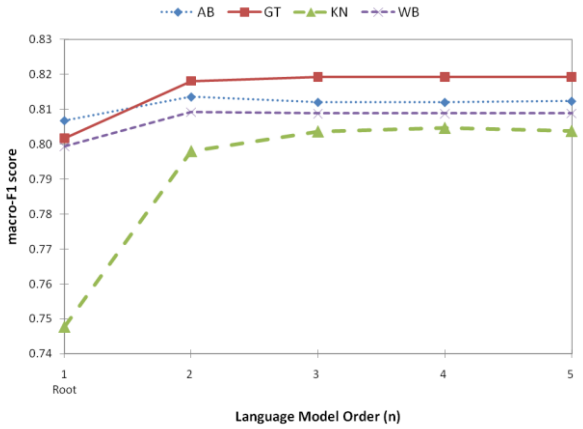

Fig. 3. Performance of root based language models with different smoothing methods with respect to language model order

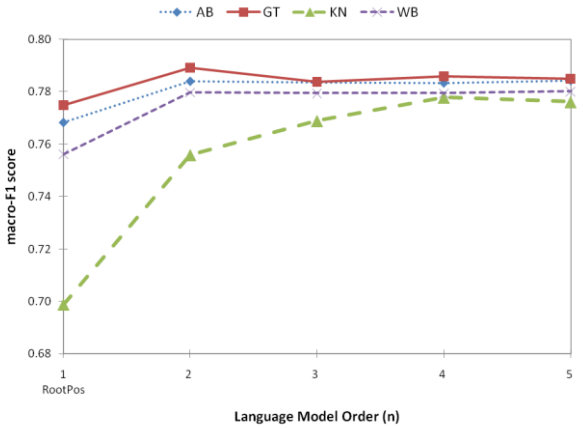

Fig. 4. Performance of root + POS tag based language models with different smoothing methods with respect to language model order

Root based language model classification results demonstrate an improvement on baseline model in Fig. 3. Except for unigrams, Good-Turing smoothing again yields superior results for all other cases.

Classification with trigram language models trained on root word units with Good-Turing smoothing can acquire 0.8198 F_1-score which is slightly better than 0.8140, the SVM performance with stemming on 1000 Turkish news documents splitted in 5 categories[29].

In root+POS model, it was expected that POS information could contribute to the overall performance by the help of the discriminative property of POS information for homograph root words. On the contrary, experimental results show no progress (even some reductions) for all smoothing methods (Fig. 4).

In our work, we have carried out some tests with the *FirstFLetter* models which make use of the truncated first *F* letters of the words as base units. Fig. 5 shows how different smoothing techniques affect the classification performance. For a further investigation on F parameter, we have conducted additional tests where $F \in \{3,\ldots,7\}$ and the results are presented in Fig. 6. The best performing smoothing method for $F=4$ is Good-Turing, so it is fixed for all other tests.

As the results indicate, *FirstFLetter* truncation models are able to achieve a classification performance close to root based models. The only exception of that finding is the case where $F= 3$, which is probably too short for discrimination. Consistently with the previous experimental results, no specific improvement is observed where $n > 3$.

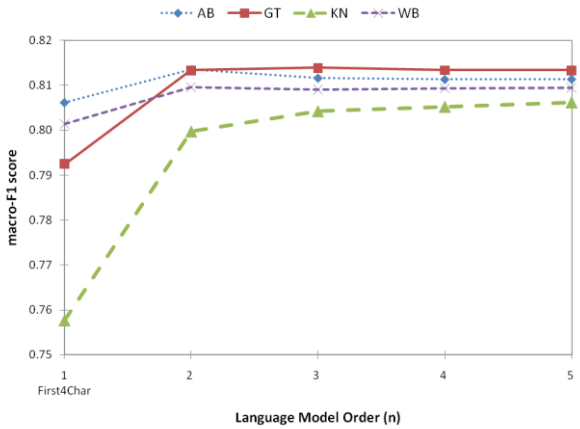

Fig. 5. Performance of the *First4Letter* based language models with smoothing methods with respect to language model order

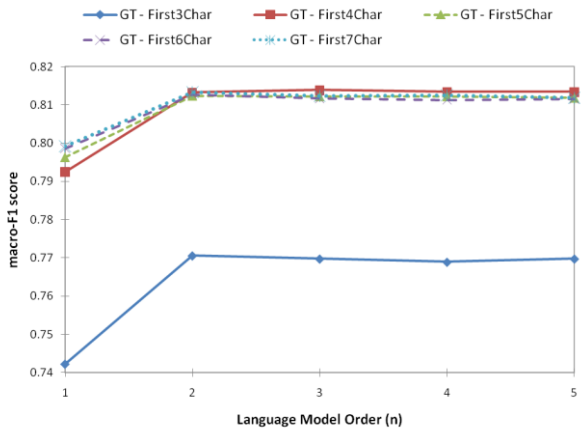

Fig. 6. Performance of the *FirstFLetter* ($F = 3,\ldots,7$) based language models with Good-Turing smoothing method along with language model order

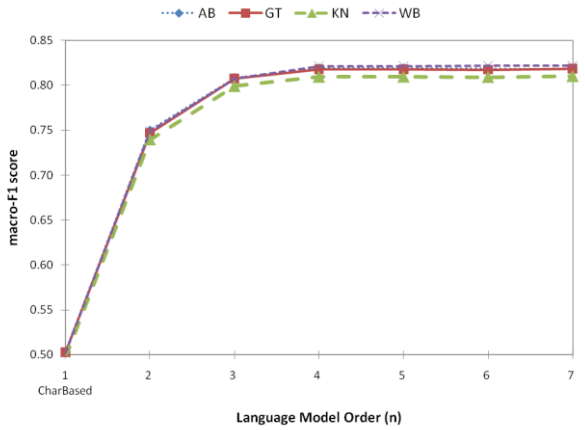

Fig. 7. Performance of character based language models with smoothing methods along with language model order

Finally, the results of the classification tests performed with character level language models are depicted in Fig. 7. A comparison between figures demonstrating classification performances of character level and word level models shows that character level language models can get classification accuracies comparable to word level models. Moreover, as character level language models are trained on word forms, language independency can be preserved without any loss in performance. From the point of view of smoothing impact on character level language model efficiency, no significant gain is observed with any method. This situation may be appropriately explained by the appearance of all possible regularities in the training data due to the extremely small vocabulary size (the number of distinct characters). In that case, all smoothing methods tend to calculate similar probability distributions for unseen sequences, so no clear disparity can be measured among them.

7.3. *Performance comparisons of different language model types*

In this section, we discuss an overall overview of experimental results and a comparative assessment of modified language models. We have used five different language model types for Turkish text classification: word form based, root word based, root word and POS tag based, first *F* character truncated word form based and character based language models. Each experiment aims to optimize at least two parameters: smoothing method and language model order. Almost all of the experiment outcomes share some common characteristics. For example, except for the character level models, no substantial progress can be observed where language model order is greater than three (*n>3)*. Thus, similar to other applications like speech recognition and machine translation, the selection of the language model order parameter *n* as 3 is shown to be suitable for text classification purposes. Another generalized outcome of the test results can be stated as the success of Good-Turing method over other smoothing methods. Language models using Good-Turing smoothing technique outperformed others whereas, in the character level case, Witten-Bell smoothing mechanism achieves slightly better accuracy than the Good-Turing and others.

The classification performances of all language model types considered in this study are consolidated in Table 6 and visualized in Fig. 8, with their best performing (optimized) parameters. For character based language model, the highest classification performance is measured where *n* is 4.

Table 6. Best performances of different language model types

n	Word form (GT)	Root (GT)	Root+POS (GT)	First4Char (GT)	CharBased (WB)
1	0.7721	0.8017	0.7749	0.7924	0.5023
2	0.7767	0.8181	0.7891	0.8133	0.7488
3	0.7772	0.8192	0.7838	0.8139	0.8078
4	0.7772	0.8192	0.7859	0.8134	0.8212
5	0.7769	0.8192	0.7850	0.8134	0.8209

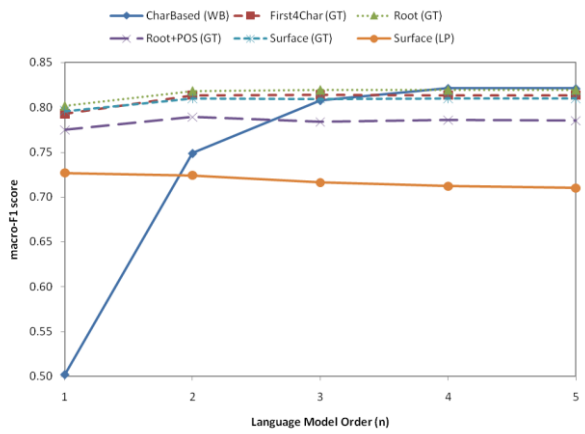

Fig. 8. Performance comparison of proposed language model types with best performing smoothing method

As it can be drawn from the figure, highest performance is attained by the root and character based methods for $n>3$. In any case, all of the language model types outperform our baseline Naïve Bayes method for $n > 1$.

7.4. *Effect of document length in classification performance*

The effect of document length on the classification performance is also studied as a part of this work. It can be claimed that texts containing long and/or many sentences may possess more information which can guide classification algorithms work better. In that case, classifying longer documents should get superior performance than of shorter documents. Fig. 9 presents the document length histogram of the data set. Here, the length of the documents are measured in words (tokens).

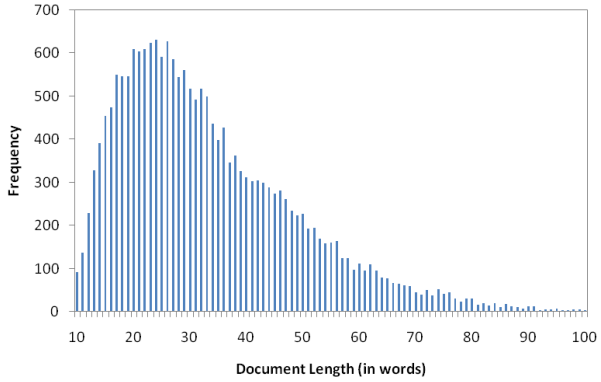

Fig. 9. Document length histogram in data set

The document length versus classification performance tests carried out by root based trigram language models with Good-Turing smoothing applied. As the graph illustrates in Fig. 10, the document length doesn't have a significant effect on performance. The sharp performance falls for both short (length<20) and long (length>80) documents are mainly caused by the insufficienct number of documents in those length ranges (see Fig. 9). It is also the main reason for the salient standard deviation jumps seen on the left and right sides of the figure.

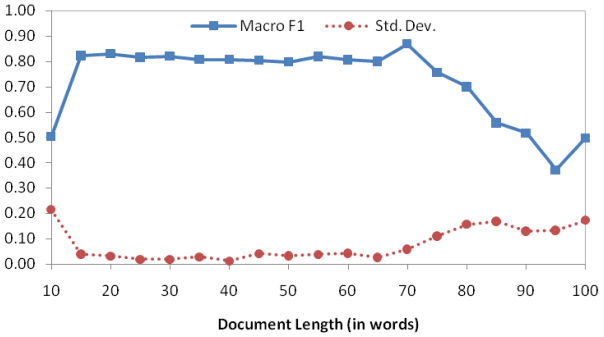

Fig. 10. Classification performance with respect to document length (in words)

8. Conclusions

In this work, we investigated the performance of document categorization with standard and modified language models for agglutinative languages. We evaluated the effects of five different language models on classification performance while three of them were suggested to overcome data sparseness problems in other application areas. Our tests are carried on a large Turkish data set; however the results can be extended to other agglutinative languages easily.

One of the results derived from our experiments reveals that statistical language model based text classifiers can outperform Naïve Bayes classifiers in Turkish document categorization task. Even word form based standard implementation of statistical language models can improve Naïve Bayes classification performance by approximately 8%, which is accomplished mainly because of advanced smoothing techniques and longer contexts. Moreover, our experimental outcomes show that root based n-gram classifiers can achieve 0.8192 F_1-score in the categorization accuracy, though they require sophisticated language dependent tools for stemming.

This means a 5.40% performance gain with respect to word form based language models. By a simple trick used in information retrieval tasks, almost same classification accuracy (F_1-score is 0.8139) is accomplished with n-grams based on truncation of first 4 letters of the word forms. We eliminate the language specific requirements by this way. Although previous studies report the success of character level n-grams, we were expecting a performance deterioration of character based models for agglutinative languages. Contrary to expectations, character level language models perform reasonably well and ranked at the top with 0.8218 F_1-score.

Similar to previous studies on other applications, having root words as base units improves the document classification performance. However, same level of accuracy can be achieved by simpler models which do not need complex language dependent tools. So, we suggest that using character level or *FirstFLetter* truncation models can perform well on statistical language model based document categorization tasks for agglutinative languages and unlike previous efforts, there is no need to apply language dependent preprocessing step.

Also, we concern with the influence of document length on classification performance. Although longer documents are supposed to help improving the classification performance by containing more discriminative words, experiments show that the accuracy of the classification remains almost steady for documents at every length, except some ignorable statistically insignificant fluctuations.

Our future work includes a comparative study on classification of the same data set with other machine learning algorithms such as support vector machines, kNN and others. Since character level n-gram models are shown to be effective in text categorization, we plan to investigate their efficiency in information retrieval tasks for agglutinative languages.

Acknowledgments

We want to thank Y. Yaslan and G. Eryiğit for reviewing this paper and providing valuable feedbacks. Also we would like to thank anonymous reviewers for their precious suggestions.

References

1. F. Sebastiani, Machine Learning in Automated Text Categorization, in *ACM Computing Surveys*, **34**(1) (2002) 1-47.
2. D. Lewis, Naive (Bayes) at forty: The independence assumption in information retrieval, in *Lecture Notes in Computer Science*, **1398** (1998) 4-18.
3. S. Robertson and K. Jones, Relevance weighting of search terms, in *Journal of the American Society for Information Science*, **27**(3) (1976).
4. Y. Li and A. Jain, Classification of text documents, in *The Computer Journal*, **41**(8) (1998) 537-546.
5. F. Peng, D. Schuurmans, and S. Wang, Augmenting naive bayes classifiers with statistical language models, in *Information Retrieval*, **7**(3) (2004) 317-345.
6. F. Jelinek, R. L. Mercer, L. Bahl, and J. K. Baker, Perplexity - A Measure of the Difficulty of Speech Recognition Tasks, in *Journal of the Acoustical Society of America*, **62**(S1) (1977) S63.
7. J. Ponte and W. Croft, A language modeling approach to information retrieval, in *Proc. 21st annual international ACM SIGIR conference on Research and development in information retrieval* (ACM New York, NY, USA, 1998).
8. P. F. Brown, et al., A Statistical Approach to Machine Translation, in *Computational Linguistics*, **16**(2) (1990) 79-85.
9. D. Z. Hakkani-Tür, K. Oflazer, and G. Tür, Statistical Morphological Disambiguation for Agglutinative Languages, in *Computers and the Humanities*, **36** (2002) 381-410.
10. G. Eryiğit, J. Nivre, and K. Oflazer, Dependency Parsing of Turkish, in *Computational Linguistics*, **34**(3) (2008).
11. M. Ganapathiraju, D. Weisser, R. Rosenfeld, J. Carbonell, R. Reddy, and J. Klein-Seetharaman, Comparative n-gram analysis of whole-genome protein sequences, in *Human Language Technologies Conference* (2002).
12. W. B. Cavnar and J. M. Trenkle, N-gram-based text categorization, in *3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)* (1994).
13. M. F. Caropreso, S. Matwin, and F. Sebastiani, A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization, in *Text Databases and Document Management: Theory and Practice* (IGI Publishing, 2001). pp. 78-102.
14. Z. Wei, D. Miao, J.-H. Chauchat, R. Zhao, and W. Li, N-grams based feature selection and text representation for Chinese Text Classification, in *International Journal of Computational Intelligence Systems*, **2**(4) (2009) 365-374.

15. E. Frank, C. Chui, and I. Witten, Text categorization using compression models, in *Conference on Data Compression*, (IEEE Computer Society, Washington DC, 2000), pp. 555.
16. J. Cleary and I. Witten, Data compression using adaptive coding and partial string matching, in *IEEE Transactions on Communications*, **32**(4) (1984) 396-402.
17. S. Zhou, K. Li, and Y. Liu, Text Categorization Based on Topic Model, in *International Journal of Computational Intelligence Systems*, **2**(4) (2009) 398-409.
18. W. Teahan and D. Harper, Using compression-based language models for text categorization, in *Language Modeling for Information Retrieval*, (2003) 141-166.
19. F. Peng and D. Schuurmans, Combining naive Bayes and n-gram language models for text classification, in *Lecture Notes in Computer Science*, (2003) 335-350.
20. T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, Stemming and lemmatization in the clustering of Finnish text documents, in *Conference on Information and Knowledge Management* (2004).
21. P. Halacsy and V. Tron, Benefits of resource-based stemming in Hungarian information retrieval, in *Lecture Notes in Computer Science*, **4730** (2007) 99-106.
22. A. Tordai and M. De Rijke, Four stemmers and a funeral: Stemming in hungarian at clef 2005, in *Lecture Notes in Computer Science*, **4022** (2006) 179.
23. A. Zelaia, I. Alegria, O. Arregi, and B. Sierra, Analyzing the effect of dimensionality reduction in document categorization for Basque, in *Archives of Control Sciences*, **600** (2005) 202.
24. M. Amasyali and B. Diri, Automatic Turkish Text Categorization in Terms of Author, Genre and Gender, in *Lecture Notes in Computer Science*, **3999** (2006) 221.
25. L. Özgür, T. Güngör, and F. Gürgen, Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish, in *Pattern Recognition Letters*, **25**(16) (2004) 1819-1831.
26. A. C. Tantug and G. Eryiğit, Performance Analysis of Naïve Bayes Classification, Support Vector Machines and Neural Networks for Spam Categorization in *Applied Soft Computing Technologies: The Challenge of Complexity* (Springer Berlin, 2006). pp. 495-504.
27. U. Ilhan, Application of K-NN and FPTC based text categorization algorithms to Turkish news reports, in *Dept. of Comp. Eng.*, (Bilkent University, Ankara, 2001).
28. M. Amasyali and T. Yildirim, Automatic text categorization of news articles, in *IEEE 12th Signal Processing and Communications Applications Conference* (2004).
29. Z. Cataltepe, Y. Turan, and F. Kesgin, Turkish Document Classification Using Shorter Roots, in *IEEE 15th Signal Processing and Communications Applications*, (2007), pp. 1-4.
30. J. T. Goodman, A Bit of Progress in Language Modeling Extended Version, (Microsoft Research, Redmond, US, 2001).
31. S. Chen and J. Goodman, An empirical study of smoothing techniques for language modeling, in *Computer Speech and Language*, **13**(4) (1999) 359-394.
32. H. Ney, U. Essen, and R. Kneser, On structuring probabilistic dependences in stochastic language modeling, in *Computer, Speech and Language*, **8**(1) (1994) 1-38.
33. H. Jeffreys, *Theory of Probability*. 2nd ed. (Clarendon Press, Oxford, 1948).
34. W. Johnson, Probability : The Deductive and Inductive Problems, in *Mind*, **41**(164) (1932) 409.
35. G. Lidstone, Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities, in *Transactions of the Faculty of Actuaries*, **8** (1920) 182-192.
36. I. H. Witten and T. C. Bell, The Zero-Frequency Problem : Estimating the Probabilities of Novel Events in Adaptive Text Comprassion, in *IEEE Transactions on Information Theory*, **37**(4) (1991) 1085-1094.
37. S. M. Katz, Estimation of Probabilities from Sparse Data for Language Model Component of a Speech Recognizer, in *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**(3) (1987) 400-401.
38. I. Good, The population frequencies of species and the estimation of population parameters, in *Biometrika*, **40**(3-4) (1953) 237-264.
39. D. Jurafsky and J. Martin, *Speech and language processing* (Prentice Hall, 2008).
40. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing* (The MIT Press, Cambridge, 1999).
41. F. He and X. Ding, Improving Naive Bayes Text Classifier Using Smoothing Methods, in *29th European Conference on IR Research, ECIR 2007*, (Springer, 2007), pp. 703.
42. J. Hankamer, Finite State Morphology and Left to Right Phonology, in *West Coast Conference on Formal Linguistics Forum* (Stanford University, 1986).
43. K. Oflazer, Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction, in *Computational Linguistics*, **22**(1) (1996) 73-89.
44. D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language*

Processing, Computational Linguistics and Speech Recognition (Prentice Hall 2000).
45. E. Arısoy, H. Dutağacı, and L. Arslan, A unified language model for large vocabulary continuous speech recognition of Turkish, in *Signal Processing*, **86**(10) (2006) 2844-2862.
46. D. Yuret and E. Biçici, Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies, in *The Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, (2009).
47. G. Eryiğit and K. Oflazer, Statistical dependency parsing of Turkish, in *The 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (2006).
48. A. C. Tantuğ, E. Adalı, and K. Oflazer, Lexical Ambiguity Resolution for Turkish in Direct Transfer Machine Translation Models, in *Lecture Notes in Computer Science*, **4263** (2006) 230-238.
49. F. Ekmekcioglu, M. Lynch, and P. Willett, Stemming and n-gram matching for term conflation in Turkish texts, in *Information Research*, **2**(2) (1996) http://informationr.net/ir/2-2/paper13.htm.
50. H. Sever and Y. Tonta, Truncation of content terms for Turkish, in *Conference on Intelligent Text Processing and Computational Linguistics, CICLing*, (2006).
51. F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. Ocalan, and O. Vursavas, Information retrieval on Turkish texts, in *Journal of the American Society for Information Science and Technology*, **59**(3) (2008) 407-421.
52. M. Porter, An algorithm for suffix stripping, in *Readings in information retrieval* (Morgan Kaufmann Publishers Inc., San Francisco, CA,, 1997). pp. 313-316.
53. K. Oflazer, Two-level Description of Turkish Morphology, in *Literary and Linguistic Computing*, **9**(2) (1995) 137-148.
54. A. C. Tantuğ, E. Adalı, and K. Oflazer, A Prototype Machine Translation System Between Turkmen and Turkish, in *Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN*, (2006).
55. T. Güngör, Lexical and morphological statistics for Turkish, in *International Twelfth Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN 2003*, (2003).
56. A. Stolcke, SRILM - An Extensible Language Modeling Toolkit, in *International Conference on Spoken Language Processing*, (2002).