

# A New Word Clustering Method for Building N-Gram Language Models in Continuous Speech Recognition Systems

Mohammad Bahrani, Hossein Sameti, Nazila Hafezi, and Saeedeh Momtazi

Speech Processing Lab, Computer Engineering Department,  
Sharif University of Technology, Tehran, Iran  
{bahrani, sameti, hafezi, momtazi}@ce.sharif.edu

**Abstract.** In this paper a new method for automatic word clustering is presented. We used this method for building n-gram language models for Persian continuous speech recognition (CSR) systems. In this method, each word is specified by a feature vector that represents the statistics of parts of speech (POS) of that word. The feature vectors are clustered by k-means algorithm. Using this method causes a reduction in time complexity which is a defect in other automatic clustering methods. Also, the problem of high perplexity in manual clustering methods is abated. The experimental results are based on "Persian Text Corpus" which contains about 9 million words. The extracted language models are evaluated by the perplexity criterion and the results show that a considerable reduction in perplexity has been achieved. Also reduction in word error rate of CSR system is about 16% compared with a manual clustering method.

**Keywords:** Class n-gram Models, Continuous Speech Recognition, Part Of Speech, Persian Text Corpus, Word Clustering.

## 1 Introduction

One of the most efficient methods for improving the accuracy of continuous speech recognition system is using language information (statistical, grammatical, etc). There are many methods for incorporating language models in speech recognition. The main method is incorporating statistical language models that appear as n-gram models [1], [2].

In n-gram language models we assume that the probability of occurrence of one word in a text depends on just n-1 previous words [3]. In every language, for extracting this kind of language models, a large amount of data is required. The more amounts of data, the more correct estimations of n-gram models.

Unfortunately, collecting data and providing a suitable text corpus for the Persian language is in its primary stages and the amount of data is not enough. Therefore extracted n-gram models will be too sparse.

One of the ways for solving this problem is using word clustering methods [2]. It means that we can use the statistics between clusters of words instead of the statistics

between words, when the word-based n-gram models are not accurate enough. The n-gram language models that are constructed with these methods are called "class-based n-gram models".

For example, consider this sentence: "He woke up Sunday morning at 7 a.m.". For computing the bigram statistics between the words of the sentence, we should use the statistics between "Sunday" and "morning" but you see that there is little difference between (Sunday, morning), (Monday, morning) or (Sunday, night), etc. because the functions of these words are similar in language.

The idea of word clustering comes from here. We can assign similar words in one unique class and use the statistics between classes instead of words. For achieving this goal and clustering words, the simplest method is clustering them according to their part of speech (POS) tags [4]. This means that the words with similar POS tags (nouns, adjectives, verbs, prepositions, etc) can be assigned in a particular class.

Beside this manual word clustering method, there are other methods which cluster the words automatically [5], [6]. In most of them, the clustering process is based on the perplexity criterion.

In this research, a new method has been used for automatic word clustering in order to build a suitable language model. This method is a concatenation of two different methods: the automatic word clustering and the manual (POS-based) word clustering.

The text corpus that we used for building the language models is "Persian Text Corpus" [13]. This corpus and its features are introduced in the next section. Section 3 deals with the different methods of word clustering. Our new method and its advantages are discussed in section 4. The experimental results go on in fifth section. Finally, in section 6, concluding remarks are made.

## 2 Persian Text Corpus

In this research, we have used the first edition of "Persian Text Corpus", the only available text corpus in Persian.

"Persian Text Corpus" contains of various types of Persian texts, including about 9 million words annotated with POS tags. The texts of this Corpus are gathered from various data sources (like newspapers, magazines, journals, books, letters, hand-written texts, scenarios, news and etc.). The whole set of this data is a complete set of Persian contemporary texts. The texts are about different subjects like politics, art, culture, economics, sport, stories and etc.

All files of "Persian Text Corpus" are in text (.txt) format. For every ".txt" file, there is a file in ".lbl" format which contains the words of ".txt" file plus a tag per word. These tags which include one or more characters, show the syntactic category (POS) and -if it is necessary- the syntactic or semantic subcategories of words.

Generally, 882 POS tags have been used in "Persian Text Corpus". Not only is this number of POS tags very large, but also some of them are so detailed and some of them are rarely used. Therefore after revising the POS tags and extracting the statistics of each tag, we have classified them into 164 classes -based on the syntactic similarity between them- and assigned a general tag for each new class. By this mean the number of POS tags is reduced and infrequent tags are assigned to more general classes.

For those tags which were infrequent and were not included in any defined class, we have assigned a new tag, named "IGNORE".

For the end of a sentence which is the start of the next sentence, another tag is assigned named "NULL". In processing the sentences of corpus, where there was one of the end of a sentence symbols ('.', '?', '!' and ':') we supposed that there is a NULL tag there. After considering all above conditions, the size of tag set was reduced to 166 POS tags and we have replaced the detailed POS tags with the new general ones.

### 3 Class-Based N-Gram Models

As we discussed in the introduction, because of the lack of data in text corpus, we have to use the "class-based n-gram models". In these models the n-gram probability of word  $w_n$  given word sequence  $w_1w_2...w_{n-1}$  that are assigned to classes  $c_1c_2...c_n$  is calculated as follows [2]:

$$P(w_n | w_1w_2 \dots w_{n-1}) = P(c_n | c_1c_2 \dots c_{n-1}) \cdot P(w_n | c_n) \tag{1}$$

For clustering words and determining the cluster of each word, there are two different forms. The first form is automatic clustering and the second one is manual clustering. In ordinary automatic word clustering, the goal is to optimize the perplexity or the average mutual information between classes. Brown's [5] and Martin's [6] algorithms are the most famous automatic word clustering algorithms. But the time complexity of them is very high and especially when the vocabulary size or the number of clusters is large, these types of word clustering algorithm are costly.

In other methods, each word is represented as a feature vector and the clustering procedure is accomplished based on the similarity between those vectors. Korkmaz algorithm [7] is the most prevalent algorithm that uses words feature vector. This algorithm uses the bigram probabilities of each word as its feature vector.

The manual word clustering methods are usually based on the POS or semantic categories of words. POS-based n-gram models can be extracted from text corpora which have POS tags for words [4], [12].

Because every word can have different POSs in different sentences, each word can be assigned to more than one class by this method. This defect leads to high perplexity of language in comparison with automatic methods. Table 1 shows a comparison between automatic and manual techniques:

**Table 1.** The comparison between automatic and manual clustering

	Automatic Clustering	Manual Clustering
Time Complexity	Very High	Very Low
Perplexity	Low	High
Number of Clusters for each Word	Only One Cluster	More than One Cluster

Considering the above table, you see that there are some problems in using each of these methods. In automatic word clustering algorithms that optimize the perplexity, the time complexity is very high. Also in the POS-based manual word clustering

algorithm the perplexity of language is high and one word may be assigned to more than one class.

To tackle these problems successfully, according to this comparison, a new idea has been proposed. We decided to merge these two methods and reached a new method for word clustering which obeys the disciplines of automatic word clustering and has the concept of manual word clustering. According to the results, the new method leads to highlighting the advantages and abates the problems of both previous methods.

## 4 A New Word Clustering Method

As we discussed in previous sections, in Persian language, every single word can get different POS tags in different contexts based on its syntactic function and its meaning. The POS tags statistics for each word have been extracted from "Persian Text Corpus" through the following steps:

- Step 1.* The varieties of words and the frequency of occurrence of them in "Persian Text Corpus" were extracted.
- Step 2.* The 20000 most frequent words were defined as the "vocabulary" of system; the other words were considered as "out of vocabulary".
- Step 3.* The number of occurrence of each POS tag for each word of vocabulary was extracted. As we decreased the number of POS tags to 166, the result of this step is saved in a 20000\*166 matrix which we called "POS Matrix".

For example, the word "zib/" (زیبا) in "Persian Text Corpus" has three different POS tags according to the text which it has been used in. It appears as non-quantitative simple adverb (ADV\_NQ\_SIM) 9 times, singular private noun (N\_SING\_PR) 7 times and simple adjective (ADJ\_SIM) 420 times. The statistics of other POS tags for this word are zero. Therefore the row of POS matrix for "zib/" has three non-zero elements (9, 7 and 420) and has zero values in all other elements.

As another example, we discuss about the word (مرد). As in Persian language we use Arabic script, short vowels are not written and capitalization is not used. So the word (مرد) has several POS tags, based on its pronunciation and meaning. In "Persian Text Corpus", it appears as common single noun (N\_SING\_COM) 1201 times and as simple adjective (ADJ\_SIM) 32 times when it is pronounced as "mard" (man) and it appears as verb with simple past tense (V\_PA\_SIM) 47 times when it is pronounced as "mord" (died). It is necessary to mention that because separating the statistics of Persian homograph words is very difficult, it was not considered in this research.

We want to use the fact that every word can appear with different POS tags, for word classification. Let us suppose that each row of POS matrix is the feature vector of related word. These vectors are used for word clustering. The reason that we use these vectors for word clustering is obvious, based on the above examples. Clearly, two words that have the same POS tags distribution have similar feature vectors and both of them should be assigned to one class. For example, all verbs with simple past tense are classified in one class and all singular common nouns are assigned in another one.

In this research, we use k-means clustering algorithm for clustering the words. At the first step, primary clusters of words are made and the centroid of each cluster is computed. Then we continue the clustering process with k-means algorithm. It means that we compute the distances between each vector and the centroids of all clusters. Then the centroid that has minimum distance to the vector is updated according below [14]:

$$\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} + \varepsilon(t)(\mathbf{w} - \mathbf{c}^{(t)}) \quad (2)$$

Where  $\mathbf{c}'$  is the nearest centroid with word vector  $\mathbf{w}$  at iteration  $t$  and  $\varepsilon(t)$  is a non-increasing function of  $t$ .

The process of computing the distances between each vector and the centroids and updating the centroids will be continued until no change appears in the results of two consequential iterations or until a given number of iterations is fulfilled.

The cosine distance (equation 3) is used for computing the distances between the word vectors  $\mathbf{w}$  and  $\mathbf{v}$ :

$$d(\mathbf{w}, \mathbf{v}) = \cos^{-1} \frac{\mathbf{w} \cdot \mathbf{v}}{|\mathbf{w}| |\mathbf{v}|} \quad (3)$$

In k-means algorithm the number of clusters should be predefined. We assumed different numbers of clusters for implementing our new clustering method.

The most important advantages of this algorithm in comparison with other automatic word clustering algorithms (like Martin's and Brown's algorithms) are its low time complexity and using POS tags for assigning each word to a cluster. Table 2 shows a comparison between time complexities of our clustering algorithm, the Brown's algorithm [5] and the Martin's algorithm [6]. In this table  $N$ ,  $V$ ,  $C$  and  $d$  represent corpus size, vocabulary size, number of classes, and dimension of feature vectors respectively.

**Table 2.** A comparison between time complexities of our clustering method, Brown's algorithm and Martin's

Our method	$O(VCd)$
Brown's algorithm	$O(V^3)$
Martin's algorithm	$O(N+VC^2)$

Also, with this method each word appears in just one cluster. In other techniques that use the feature vector for each word (like Korkmaz algorithm), the bigram statistics of each word are used as the feature vector. In these techniques if the size of vocabulary is large, the dimension of feature vectors become too large and it makes the clustering procedure very complicated, while in our word clustering method the dimension of feature vectors are small and independent of the size of vocabulary.

After the clustering process, we have built n-gram language models for word classes instead of words. The training set for clustering words is about 380000 sentences of "Persian Text Corpus" which is about 8 million words. We have built only the class bigram and class trigram language models in this research, because of the small size of our corpus.

Suppose that we partition a vocabulary of  $V$  words into  $C$  classes. Then the class bigram model has  $C(C-1)$  independent parameters of the form  $P(C_n|C_{n-1})$ ,  $C-1$  parameters of the form  $P(C_n)$ ,  $V$  parameters of the form  $P(w_n)$ , plus  $V$  parameters for determining the class of each word. Beside the class bigram model parameters, the class trigram model also has  $C^2(C-1)$  independent parameters of the form  $P(C_n|C_{n-1} C_{n-2})$ .

## 5 Experimental Results

The class language models extracted from our word clustering method are tested for evaluating two different criteria: perplexity and word error rate of a CSR system. We computed the perplexity based on class bigram and class trigram models on a test set of the last 43800 sentences (about 1 million words) of "Persian Text Corpus" not included in training data. The perplexity can be computed by equation (4) as follows:

$$PP = \hat{P}(w_1 w_2 w_3 \cdots w_m)^{-1/m} \quad (4)$$

In this equation  $m$  is the number of words in test set and  $w_1 w_2 w_3 \cdots w_m$  show the words sequence of test set.  $\hat{P}(w_1 w_2 w_3 \cdots w_m)$  can be computed by equations (5) and (6) for the bigram and trigram models respectively.

$$PP_{bi} = \left( \prod_{i=1}^m \hat{P}(w_i | w_{i-1}) \right)^{-1/m} \quad (5)$$

$$PP_{tri} = \left( \prod_{i=1}^m \hat{P}(w_i | w_{i-1} w_{i-2}) \right)^{-1/m} \quad (6)$$

In these equations,  $\hat{P}(w_i | w_{i-1})$  and  $\hat{P}(w_i | w_{i-1} w_{i-2})$  are computed by equation (1).

Table 3 shows the perplexity of "Persian Text Corpus" based on class bigram and class trigram models with different number of classes.

**Table 3.** The perplexity of "Persian Text Corpus" based on class bigram and class trigram models with different number of classes

Language Model	Num. of Classes	Perplexity
Class Bigram	100	739
	200	699
	300	652
	500	683
Class Trigram	100	681
	200	644
	300	589
	500	651
POS-based Bigram	166	990
Martin's Algorithm (Bigram)	200	502

For comparing the new clustering method with manual POS-based one and Martin's algorithm, the last two rows of table 3 show the perplexity using manual method and Martin's algorithm. According to this table, there is an obvious reduction in perplexity using our suggested clustering method compared with manual method.

To evaluate our statistical language models in CSR system, we used SHARIF speech recognition system [9], which is a Persian speaker independent continuous speech recognition system. This system performs modeling of monophones using Hidden Markov Model (HMM) and utilizes the word search algorithm described in [10] for word recognition. In this algorithm, while recognizing the phonemes, the lexicon tree is also searched in order to find the best word sequence according to the phoneme sequence.

To run experiments, the HMMs were trained for each 30 phonemes of Persian language using 5940 sentences (about 5 hours of read speech) of FARSDAT speech database [11]. We performed the experiments on 140 sentences of FARSDAT database which don't overlap with the training data.

In general, in speech recognition systems, the language model score can be combined with acoustic model score through two methods: "during search" and "at the end of search" [8]. In this paper we have used "during search" method. In "during search" method when the search process recognizes a new word within expanding the different hypothesis, the new hypothesis score is computed via multiplication of following three terms: the n-gram score of new word, the acoustic model score of new word and current hypothesis score.

Table 4 presents the word error rates (WER) obtained when incorporating different class n-gram models in the CSR system. The last two rows show the results using POS-based method and Martin's algorithm. The results show that about 16% reduction in word error rates has been achieved compared with POS-based manual method.

**Table 4.** The word error rate obtained by incorporating different class n-gram models in CSR system

Language Model	Num. of Classes	WER [%]
Class Bigram	100	24.45
	200	24.33
	300	24.02
	500	24.35
Class Trigram	100	23.87
	200	23.40
	300	22.98
	500	23.32
POS-based Bigram	166	27.36
Martin's Algorithm (Bigram)	200	23.90

## 6 Concluding Remarks

In this paper we presented a new automatic word clustering technique based on POS tags. The class n-gram language models have been extracted from the clustering

results and evaluated in our continuous speech recognition system. The time complexity of this method is lower than other automatic methods and it is easy to implement. The perplexity of language and word error rate of CSR system reduced considerably.

## References

1. Huang, X., Alleva, F., Hon, H., Hwang, M., Lee, K., Rosenfield, R.: The SPHINX-II Speech Recognition System: An Overview. *Computer Speech and Language* 2, 137–148 (1993)
2. Young, S.J., Jansen, J., Odell, J.J., Ollason, D., Woodland, P.C.: *The HTK Hidden Markov Model Toolkit Book* (1995)
3. Rabiner, L., Juang, B.H.: *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey (1993)
4. Heeman, P.A.: POS tagging versus Classes in Language Modeling, Proc. 6th Workshop on Very Large Corpora, August 1998, pp. 179–187 (1998)
5. Brown, P., Della Pietra, V., de Souza, P., Lai, J., Mercer, R.L.: Class-based n-gram models of natural language. *Computational Linguistics* 18(4), 467–479 (1992)
6. Martin, S., Liermann, J., Ney, H.: Algorithms for bigram and trigram word clustering. *Speech Communication* 24, 19–37 (1998)
7. Korkmaz, E.E., Ucoluk, G.: A Method for Improving Automatic Word Categorization, Workshop on Computational Natural Language Learning, Madrid, Spain, pp. 43–49 (1997)
8. Harper, M.P., Jamieson, L.H., Mitchell, C.D., Ying, G.: Integrating Language Models with Speech Recognition. In: *AAAI-94 Workshop on the Integration of Natural Language and Speech Processing*, August 1994, pp. 139–146 (1994)
9. Babaali, B., Sameti, H.: The Sharif Speaker-Independent Large Vocabulary Speech Recognition System. In: *The 2nd Workshop on Information Technology & Its Disciplines*, Kish Island, Iran, February 24–26 (2004)
10. Ney, H., Haeb-Umbach, R., Tran, B.H., Oerder, M.: Improvements in Beam Search for 10000-Word Continuous Speech Recognition, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 13–16 (1992)
11. Bijankhan, M.: FARSDAT-The Speech Database of Farsi Spoken Language. In: *Proc. The 5th Australian Int. Conf. on Speech Science and Tech.*, Perth, vol. 2 (1994)
12. Bahrani, M., Samet, H., Hafezi, N., Movasagh, H.: Building and Incorporating Language Models for Persian Continuous Speech Recognition Systems. In: *Proc. 5th international conference on Language Resources and Evaluation*, Genoa, Italy, pp. 101–104 (2006)
13. BijanKhan, M.: Persian Text Corpus, Technical report, Research Center of Intelligent Signal Processing (2004)
14. Fritzke, B.: Some competitive learning methods, System Biophysics Institute for Neural Computation Ruhr-Universität Bochum (1997),  
<ftp://ftp.neuroinformatik.ruhr-unibochum.de/pub/software/NN/DemoGNG/sclm.ps.gz>