

# A Possibilistic Approach for Building Statistical Language Models

Saeedeh Momtazi  
University of Saarland  
Saarbruecken, Germany  
smomtazi@lsv.uni-saarland.de

Hossein Sameti  
Sharif University of Technology  
Tehran, Iran  
sameti@sharif.edu

## Abstract

*Class-based  $n$ -gram language models are those most frequently-used in continuous speech recognition systems, especially for languages for which no richly annotated corpora are available. Various word clustering algorithms have been proposed to build such class-based models. In this work, we discuss the superiority of soft approaches to class construction, whereby each word can be assigned to more than one class. We also propose a new method for possibilistic word clustering. The possibilistic  $C$ -mean algorithm is used as our clustering method. Various parameters of this algorithm are investigated; e.g., centroid initialization, distance measure, and words' feature vector. In the experiments reported here, this algorithm is applied to the 20,000 most frequent Persian words, and the language model built with the clusters created in this fashion is evaluated based on its perplexity and the accuracy of a continuous speech recognition system. Our results indicate a 10% reduction in perplexity and a 4% reduction in word error rate.*

## 1. Introduction

Incorporating a statistical language models is one of the most effective methods for improving the accuracy of continuous speech recognition systems [1]. Word-based  $n$ -grams are arguably the most effective method for constructing such a language model. Due to the data sparsity problem, however, this method, in its purest form, is not optimal. To overcome this problem the class-based model was introduced [6]. Through the use of such a class-based model, a language model that is simultaneously both very detailed, and generalizes well to unseen data can be constructed, and that without the need of a richly annotated corpus.

The idea of sharing the  $n$ -gram statistics of similar words helps to improve the language model. For example, consider this sentence: "He woke up Sunday morning at 7 a.m.". To compute the bigram statistics between the words

of the sentence, the bigram statistics of "Sunday" and "morning" should be found; but we see that there is no difference between the pair "Sunday morning" and the pairs "Monday morning" or "Sunday night", since the role of these words is similar in English. So, it seems reasonable to reduce the number of words in the  $n$ -gram history by considering some words as equivalent and using the class-based model instead of the word-based. This can be implemented by mapping a set of words to a word class while applying a clustering technique.

Suppose  $w_1, \dots, w_T$  are the consecutive words of a sentence  $S$  and  $C_1, \dots, C_T$  are their related classes. The probability of the sentence  $S$  can be computed as follows:

$$\begin{aligned} P(S) &\simeq \prod_{i=1}^T P(w_i|w_{i-1}) \\ &\simeq \prod_{i=1}^T P(C_i|C_{i-1})P(w_i|C_i) \end{aligned} \quad (1)$$

The goal of word clustering is to assign similar words to a unique class according to their functions in the language. As some words have different grammatical roles in different contexts, it is difficult to assign them to just one class. As an example, consider the homograph *dove*, which has two different pronunciations /d?v/ and /d?v/ with different meanings: The former pronunciation refers, of course, to a bird, and the latter is the past participial of *dive*. Regardless of homograph problem, we have another problem with words which have more than one Part Of Speech (POS) like the word *cold* which can be a noun or an adjective. There also exist homonyms, which have different meanings with the same POS and pronunciation; e.g., the word *bank*.

For these reasons, it is worthwhile to utilize a more sophisticated model for capturing the several senses of each word, and assigning it to more than a single class. This issue has motivated a great deal of research on soft clustering methods. Most researchers use fuzzy clustering algorithms to achieve this goal [9]. In this work, we introduce a possibilistic approach for clustering words and demonstrate that this new method is more effective than the fuzzy methods.

The remainder of this paper is organized as follows: In Section 2, we discuss fuzzy and possibilistic clustering, along with the superiority of the latter to the former. The parameters of the possibilistic algorithm for the word clustering application are discussed in Section 3. In Section 4, the construction of the class-based language model is described. The experimental results and comparison with other methods are presented in Section 5. Finally, Section 6 summarizes the paper.

## 2 Possibilistic Clustering

The Fuzzy Theory is one of the prevalent approaches that can be used for soft word clustering. The idea of the Fuzzy Theory is allowing words to belong to more than one classes by a membership degree. The membership degree shows the probability of assigning a word to a class which should satisfy the following constraint:

$$\sum_j u_{ij} = 1 \quad \text{for all } i \quad (2)$$

where  $u_{ij}$  is the membership values of  $i^{th}$  word to the  $j^{th}$  class.

One of the most important soft clustering techniques is fuzzy clustering. The *Fuzzy C-Means* (FCM) algorithm proposed by Bezdek [4] is the most well-known algorithm in fuzzy clustering. This technique attempts to find a fuzzy partitioning of a given training set by minimizing a fuzzy generalization of the least square error function. Let us take the FCM function as our objective function,

$$J_q(U, Y) = \sum_{i=1}^N \sum_{j=1}^K (u_{ij})^q d^2(X_i, C_j) \quad (3)$$

where  $X_i$  is the feature vector of  $i^{th}$  word,  $C_j$  is the centroid of  $j^{th}$  class,  $K$  is the number of classes, and  $N$  denotes the number of elements.  $U = [u_{ij}]$  is the  $N \times K$  fuzzy-partition matrix, containing the membership values of all samples in all clusters;  $d^2(X_i, C_j)$  is the distance between  $i^{th}$  sample and  $j^{th}$  class; and  $q$  is the fuzzification factor that controls the fuzziness of the resulting cluster.

The minimization of  $J_q$  under the probabilistic constraint (3) leads to the following formulae for the update of all membership degrees and class centroids,

$$u_{ij} = \frac{\left| \frac{1}{d^2(X_i, C_j)} \right|^{\frac{1}{q-1}}}{\sum_{k=1}^K \left| \frac{1}{d^2(X_i, C_k)} \right|^{\frac{1}{q-1}}} \quad (4)$$

$$C_j = \frac{\sum_{i=1}^N (u_{ij})^q X_i}{\sum_{i=1}^N (u_{ij})^q} \quad (5)$$

An iteration is performed between these two formulae until a termination criterion is reached.

A major fault of the FCM algorithm is its basis on the probabilistic constraint that the sum of the membership values of a pattern over all clusters must be unity. This constraint implies that the membership of a point in a cluster depends on the membership of that point in all other clusters. If the membership values are intended to represent the degree of compatibility, this probabilistic constraint on  $U$  is too restrictive. This constraint may give meaningful results in applications where it is desired to interpret the membership values as probabilities, or as degrees of sharing. In this work, all to the contrary, we want the membership values to indicate typicality or resemblance with a prototype rather than probabilities. Krishnapuram and Keller [10] noted that the membership values under such a typicality interpretation should be absolute, and not dependent on other memberships. For this reason, they introduced a new clustering algorithm known as *Possibilistic C-Means* (PCM).

Using PCM algorithm of Krishnapuram and Keller [11], we cast the clustering problem in the framework of possibility theory. In the possibilistic approach, the membership values of each word to each class may be interpreted as degrees of possibility of the points belonging to the classes, i.e., the compatibilities of a word with the class prototypes. In other words, membership values are solely a function of the distance of a word from a class center. In order to allow a possibilistic interpretation of the membership function as a degree of typicality, the probabilistic constraint is relaxed in the PCM algorithm. So that the membership degrees must simply verify:

$$Max_i u_{ij} > 0 \quad \text{for all } j \quad (6)$$

Towards this end, the objective function of PCM is defined as follows

$$J_q(U, Y) = \sum_{i=1}^N \sum_{j=1}^K (u_{ij})^q d^2(X_i, C_j) + \sum_{j=1}^K \eta_j \sum_{i=1}^N (u_{ij} \log u_{ij} - u_{ij}) \quad (7)$$

To minimize  $J$ , under the constraint (7),  $u_{ij}$  should be computed according to

$$u_{ij} = \frac{1}{1 + \left( \frac{d^2(X_i, C_j)}{\eta_j} \right)^{\frac{1}{q-1}}} \quad (8)$$

where  $\eta_j$  denotes the distribution degree of each class. To estimate the parameters  $\eta_j$ , a bootstrap clustering algorithm must be applied before starting PCM. In this paper, we use the outputs of the FCM to estimate  $\eta_j$ . After estimating  $\eta_j$  with FCM memberships, we update it in each iteration with new PCM memberships,

$$\eta_j = \frac{\sum_{i=1}^N (u_{ij})^q d^2(X_i, C_j)}{\sum_{i=1}^N (u_{ij})^q} \quad (9)$$

### 3 Algorithm Parameters

#### 3.1 Centroid Initialization

In most practical applications of soft clustering algorithms, the procedure of initializing the centroids of classes is random [7]. There are two methods for random initialization; in the first method, the elements of membership matrix are filled randomly at first. Then the centroids of random classes are calculated. In the second method, some elements are selected randomly as the centroids of the classes at the beginning of the process; and then each word receives a membership degree based on the random centers. Zhang [14] noted that the second initialization method works better than the first one. None of these methods are recommended, however, due to the strong possibility of getting stuck in a local minimum.

Here we propose a method for initialization that provides superior results. This method uses the results of a hard clustering method to initialize the cluster centroids. For instance, the outputs produced by the algorithms proposed by either Martin [12] or Brown [6] could be used for initialization.

#### 3.2 Distance Measure

Various vector-based similarity measures, such as Euclidian or cosine distance, could conceivably be used for computing the similarity or dissimilarity of two words. To use the Euclidian or cosine distance, each class should be considered as a known sample. In most of the cases, the centroid of the class is selected and the distance between each word and its class centroid is computed. In these techniques, other patterns in the class do not affect the distance and only the class centroid is considered. For incorporating all class samples, it is better to use a distance measure that computes the distance between a distribution and a term. In this manner, we can measure the distance between a class as a distribution and a word as a term. A popular distance measure for our goal is Mahalanobis distance, which can determine the similarity of an unknown sample set to a known one.

#### 3.3 Feature Vector

In possibilistic word clustering, like in most clustering algorithms for various types of data, each word is represented by a feature vector. One possible feature vector for each word is the bigram statistics between that word and other words. Also, this feature vector can be defined as the bigram statistics between other words and that word. The combination of both of these statistics was proposed by Korkmaz [9].

A new idea for defining the feature vectors is using the POS statistics of each word as its feature vector. Using this feature vector, we get better results in comparison to other feature vectors. In Persian, like most languages, each single word might have different POS tags in different contexts based on its usage in a sentence and its meaning. In this research, the statistics of POS tags for each word have been extracted from the *Persian Text Corpus* through four steps described by Bahrani [3]. The extracted statistics are stored in the *POS Matrix*, each of whose rows represents the feature vector of one word.

### 4 Building Class-based Language Model

After the word clustering has been completed, the next step is to extract  $n$ -gram statistics between classes to build a class-based language model. In hard clustering algorithms, the bigram probability between two classes  $C_i$  and  $C_j$  is computed by summing bigram statistics between the words of those classes, according to

$$N(C_i, C_j) = \sum_{w_k \in C_i, w_l \in C_j} N(w_k, w_l) \quad (10)$$

Having soft membership degrees at hand, we now use a new approach to consider the membership degree of words to classes. One suitable formula can be obtained by multiplying both membership functions to bigram statistics, such that

$$N(C_i, C_j) = \sum_{w_k, w_l} u_{ki} u_{lj} N(w_k, w_l) \quad (11)$$

Another formula is taking the minimum rather than the product of the two terms and use it for calculating the bigram statistics:

$$N(C_i, C_j) = \sum_{w_k, w_l} \min(u_{ki}, u_{lj}) N(w_k, w_l) \quad (12)$$

After calculating the bigram statistics, the class-based bigram can be specified as [8]

$$P(w_m | w_{m-1}) = \sum_{C_j} [P(w_m | C_j) \sum_{C_i} P(C_j | C_i) P(C_i | w_{m-1})] \quad (13)$$

Here,  $P(C_i | w_{m-1})$  is the degree of membership of  $w_{m-1}$  to  $i^{th}$  class. The other parameters of (13) can be computed as

$$P(C_j | C_i) = \frac{N(C_i, C_j)}{N(C_i)} \quad (14)$$

$$P(w_m | C_j) = \frac{P(C_j | w_m) N(w_m)}{N(C_j)} \quad (15)$$

### 5 Experimental Results

For our experiments, we used a bigram model, but intend to use a higher order  $n$ -gram model in future. The

**Table 1.** The perplexity obtained by incorporating different class bigram models on the ‘‘Persian Text Corpus’’

| LM Type              | Classes | Perplexity   |
|----------------------|---------|--------------|
| POS Tagger           | 166     | 1052         |
| Korkmaz Fuzzy Method | 100     | 792.8        |
|                      | 200     | 770.9        |
|                      | 500     | 784.1        |
| FCM Bigram Vector    | 100     | 760.4        |
|                      | 200     | 738.7        |
|                      | 500     | 749.4        |
| FCM POS Vector       | 100     | 740.6        |
|                      | 200     | 705.3        |
|                      | 500     | 720.8        |
| PCM Bigram Vector    | 100     | 746.4        |
|                      | 200     | 723.4        |
|                      | 500     | 730.5        |
| PCM POS Vector       | 100     | 716.9        |
|                      | 200     | <b>693.2</b> |
|                      | 500     | 705.3        |

class-based bigram model extracted from our word clustering method are evaluated using two different criteria, *Perplexity* and *Word Error Rate* (WER) of a continuous speech recognition system.

For calculating the perplexity of a text using a language model, we need the *Entropy* which is defined as

$$H = -\frac{1}{M} \sum_{m=1}^M \log(P(w_m|w_{m-1})) \quad (16)$$

where  $M$  is the number of words in the text. Having the entropy of a language model, the perplexity is calculated as follows

$$\text{Perplexity} = 2^H \quad (17)$$

The perplexity of our language model was computed on a test set of the ‘‘Persian Text Corpus’’ consisted of 43,800 sentences. This test set is disjoint from the training set we used. Since neither the FCM nor PCM algorithms provide a way for determining the optimum number of classes, we tested different numbers of classes to cluster the first 20,000 most frequent Persian words derived from our corpus. We achieved the best result with 200 classes.

As mentioned, there are various numbers of parameters that should be defined. For initializing centroids, we tried on two random models plus our new model while using the results of Martin’s word clustering algorithm [12]. As expected, the new initialization performed better than the random ones; so that it is used for all of our experiments reported here. We applied three different distance measures

**Table 2.** The WER obtained by incorporating different class bigram models in the ‘‘Sharif’’ CSR system

| LM Type              | Classes | WER%        |
|----------------------|---------|-------------|
| POS Tagger           | 166     | 31.5        |
| Korkmaz Fuzzy Method | 100     | 30.3        |
|                      | 200     | 29.7        |
|                      | 500     | 29.9        |
| FCM Bigram Vector    | 100     | 29.0        |
|                      | 200     | 28.6        |
|                      | 500     | 28.7        |
| FCM POS Vector       | 100     | 27.0        |
|                      | 200     | 26.5        |
|                      | 500     | 26.7        |
| PCM Bigram Vector    | 100     | 28.2        |
|                      | 200     | 27.5        |
|                      | 500     | 27.7        |
| PCM POS Vector       | 100     | 26.3        |
|                      | 200     | <b>25.8</b> |
|                      | 500     | 25.9        |

(Euclidian, Cosine, and Mahalanobis) too; The Cosine distance measure performs the best for fuzzy clustering and the Mahalanobis distance outperforms the other methods for possibilistic clustering. Both word feature vectors (Bigram and PSO) have also been used in all of our experiments.

To build the final language model, we used Equations 11 and 12; the first formula performed better than the second one for the fuzzy clustering. However, the experiments on possibilistic clustering shows that the second formula achieves a better performance; so, we have used Equation 11 for our experiments on FCM and Equation 12 for PCM.

Table 1 represents the perplexity obtained when using bigram models of different types, with various numbers of classes on our corpus. We compared our results with POS tagging as the baseline of the class-based model and the Korkmaz [9] method as the most well-known algorithm in fuzzy word clustering. We also presented the results of the FCM method to show the superiority of possibility clustering to fuzzy clustering. As we can see from the table, the PCM models outperform both the baselines and the FCM models in which there is a considerable reduction of up to 10% in perplexity as compared to the Korkmaz method.

To evaluate our model in a continuous speech recognition system, we used the ‘‘Sharif’’ speech recognition system [2], which is a Persian speaker independent continuous speech recognition system. This system models monophones using hidden markov model and utilizes the word search algorithm described by Ney [13] for word recognition. While recognizing the phonemes in this algorithm, the lexicon tree is also searched to find the best word sequence

according to the sequence of phonemes.

To run experiments, the hidden markov models were trained for each of 30 phonemes of the Persian language using 5,940 sentences, which is approximately five hours of reading speech from the “Farsdat” speech database [5]. We performed the experiments on 140 sentences of “Farsdat”. This test set was similarly disjoint from the training set. Table 2 presents the obtained WER when incorporating different class bigram models in the continuous speech recognition system.

The results on this table show a considerable reduction of up to 4% in WER when we applied our method on the “Sharif” continuous speech recognition system.

## 6 Concluding Remarks

In this paper, we described the advantage of possibility theory and presented a clustering algorithm based on this theory for clustering words. To reach an optimal clustering method, different parameters of possibilistic word clustering algorithm were considered and evaluated. The criterion for choosing the appropriate parameters was the perplexity of context. Finally the extracted class-based models of this research were incorporated in a Persian continuous speech recognition system and decreased the WER about 4%.

## References

- [1] J. Allen. *Natural Language Understanding*. The Benjamin-Cummings Publishing Company, 1994.
- [2] B. Babaali and H. Sameti. The sharif speaker-independent large vocabulary speech recognition system. In *Proceeding of 2nd Workshop on IT and Its Disciplines*, pages 24–26, 2004.
- [3] M. Bahrani, H. Sameti, N. Hafezi, and S. Momtazi. A new word clustering method for building n-gram language models in continuous speech recognition systems. In *IEA/AIE International Conference Proceedings*. Springer, 2008.
- [4] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [5] M. Bijankhan. Farsdat:the speech database of farsi spoken language. In *Proceedings of 5th Australian International Conference on Speech Science and Technology*, volume 2, 1994.
- [6] P. Brown, V. D. Pietra, P. deSouza, J. Lai, and R. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
- [7] G. Fung. A comprehensive overview of basic clustering algorithms. Technical report, University of Wisconsin, Madison, 2001.
- [8] M. Jardino and G. Adda. A class bi-gram model for very large corpus. In *Proceedings of 3rd European Conference on Spoken Language Processing*, Yokohama, Japan, 1994.
- [9] E. Korkmaz and G. Ucoluk. A method for improving automatic word categorization. In *CONLL International Conference Proceedings*, 1997.
- [10] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1:98–110, 1993.
- [11] R. Krishnapuram and J. Keller. The possibilistic c-means algorithm: Insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4:385–393, 1996.
- [12] S. Martin, J. Liemann, and H. Ney. Algorithms for bi-gram and tri-gram word clustering. *Speech communication*, 24, 1998.
- [13] H. Ney, R. Haeb-Umbach, B. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. In *IEEE ICASSP International Conference Proceedings*, pages 13–16, 1992.
- [14] L. Zhang. Comparison of fuzzy c-means algorithm and new fuzzy clustering and fuzzy merging algorithm. Technical report, University of Nevada, Reno, 2001.