A study of unsupervised clustering techniques for language modeling

Sangyun Hahn¹, Abhinav Sethy², Hong-Kwang J. Kuo², Bhuvana Ramabhadran²

¹Dept. of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA
²IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract

There has been recent interest in clustering text data to build topic-specific language models for large vocabulary speech recognition. In this paper, we studied various unsupervised clustering algorithms on several corpora. First we compared the clustering methods with quality metrics such as entropy and purity. Of the techniques studied, two-phase bisecting K-means achieved good performance with relatively fast speed. Then we performed speech recognition experiments on English and Arabic systems using the automatically derived topic-based language models. We obtained modest word error rate improvements, comparable to previously published studies. A careful analysis of the correlation between word error rate and the distribution of misrecognized words, including an information-gain metric, is presented.

Index Terms: Clustering, Language Model Adaptation, Entropy.

1. Introduction

Automated Speech Recognition (ASR) systems make errors that stem from both acoustic (e.g., noise, accent, compression) and language (e.g., rare words, domain-specific terms, speaking style) artifacts. Most ASR systems make use of all sources of information at their disposal, i.e., domain-specific acoustic and language model training material and any relevant textual data that closely matches the domain/topic of interest. Recently, several researchers have focused on the use of topic based clustering for improved language modeling. The central idea behind this is to automatically cluster the training data into topics and build language models (LMs) for each topic. During decoding the optimal interpolation weights for these various LMs are chosen dynamically or preset using a heldout data set.

Document level clustering was used in [1] to seed topic clusters. After clustering, an EM algorithm was applied to iteratively estimate the parameters for topic specific language models. These topic specific LMs and a general LM were interpolated and applied to new sentences. In [2], hand-segmented news stories were clustered based on the keywords provided with the broadcast news corpus. A classifier (TFIDF and Naive Bayes classifier) was trained on these clusters, and the stories in the test set were classified using this classifier. The LM corresponding to the top scoring cluster was interpolated with the general LM. Probabilistic Bag of Words (BOW) modeling techniques such as LDA [3] have recently been used as a clustering algorithm to partition the LM training data [4, 5] into topics. N-gram models are built separately for each topic and then interpolated with weights optimized either on a fixed heldout set or the first pass hypothesis. The BOW vector for clustering is restricted to named entities in [6].

In this work, we investigate the use of various clustering algorithms on several corpora. The quality of the clusters obtained is evaluated using two metrics, entropy and purity. Cluster specific language models are then built using these data-driven clusters and used for speech recognition, where performance is evaluated using Word Error Rate (WER). Careful analysis of the effects on ASR systems is also presented using the correlation between the WER and the distribution of misrecognized function words and content words, and the ability of a word to distinguish topics such as the information gain metric.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the various clustering methods prevalent in the speech and language communities. Section 3 presents clustering experiments on the Reuters task for which class labels are known, and speech recognition experiments for English and Arabic broadcast news. Section 4 provides a detailed error analysis and the key lessons learned from this work.

2. Clustering Algorithms

Many document clustering algorithms have been proposed in machine learning, information retrieval, and language processing communities (see [7, 8, 9]). Most of these algorithms require a document to be represented as a feature vector; that is, given a set of \( N \) documents, \( D = \{d_i | i = 1, \ldots, N\} \), a document \( d_i = (x_{i1}, \ldots, x_{iM}) \), where \( M \) is the number of features. For a given vocabulary \( V \), the raw counts of the words in a document or TF*IDF scores are often used as features. A clustering algorithm assigns a document to one of the pre-specified \( K \) clusters (Some algorithms can determine \( K \)). See [8] for determining the number of clusters. The following are brief descriptions of some of the most widely used clustering algorithms that are relevant to our study.

**K-means** first initializes \( K \) cluster centers as randomly chosen documents in the given dataset. Then, it assigns each document to the cluster whose center is the most similar to the document. Next, each center is recomputed from the documents assigned to that cluster. This process is repeated until the cluster membership of no document is changed. As a similarity measure, the cosine similarity \( \cos(x, y) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} \) is often used for document clustering. The K-means algorithm is easy to implement, but the cluster quality is sensitive to initialization. It is only guaranteed to find a local optimum. It can be very slow on a large dataset since its complexity is \( O(NK^2) \), but incremental K-means can improve the speed significantly with better overall similarity and lower entropy [10].

**Bisecting K-means** is variant of K-means, which repeatedly splits a selected cluster into two using K-means, starting from the entire dataset, until \( K \) clusters are obtained. The largest or the most dissimilar cluster can be selected to be split. This runs faster than K-means, and it can produce a hierarchical structure among clusters, but it is not guaranteed to find a local optimum. To overcome this problem two-phase bisecting K-means was proposed [11]. This first runs bisecting K-means followed by K-
In these algorithms, the association between topics and clusters is not modeled explicitly but achieved implicitly by the similarity of word distributions in documents. Recently, various attempts have been made to model this association explicitly [12, 3]. Probabilistic latent analysis (PLSA) [12] probabilistically models the relation between a document, words in the document, and a topic which is an unobserved latent variable. It assumes a document \( d \) and a word \( w \) is conditionally independent given a topic \( z \). \( P(d, w, z) = P(z)P(w|z)P(d|z) \) Expectation-Maximization (EM) algorithm is used to estimate the model parameters. Latent Dirichlet Allocation (LDA) [3] addresses this problem by assuming a document \( d \) as a set of words \( \{w_i\} \), \( i = 1, \ldots, l \), whose positions in the document are exchangeable, given a topic distribution \( \Theta \) which is a \( K \)-dimensional Dirichlet random variable. \( P(\Theta, z, d|\alpha, \beta) = P(\Theta|\alpha) \prod_{i=1}^{n} P(z_n|\Theta)P(w_n|z_n, \beta) \) where \( z_n \) is a topic for the word \( w_n \), and \( \alpha \) and \( \beta \) are hyper-parameters for \( \Theta \) and \( P(w|z) \) respectively. Model parameters are estimated approximately using variational EM procedure [3].

### 3. Experiments and results

Several mechanisms exist to choose the number of clusters in automated clustering methods. They range from prefixing the number of clusters based on external knowledge to the use of a stopping criterion in the clustering procedure. The metrics commonly used in evaluating clustering algorithms are entropy and purity. The entropy of a cluster \( S_r \) is defined as

\[
E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{n_r} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r}
\]

where \( q \) is the number of clusters in the dataset, \( n_r \) is the number of documents assigned to the cluster \( S_r \), and \( n_r^i \) is the number of documents of the \( i \)th cluster that were assigned to the \( r \)th cluster. The purity of a cluster \( S_r \) is defined as:

\[
P(S_r) = \frac{1}{n_r} \max(n_r^i)
\]

which is a fraction of the cluster size that belongs to the largest class of documents that were assigned to a cluster.

We begin by clustering the Reuters news corpus [13] to gain a better understanding of the clustering algorithms described in Section 2. KM, BKM, and 2PKM are implemented in the software package, Cluto [14] which we have used throughout this work. For PLSA, we used Lemur package [15], and for LDA, we used the LDA software by Blei [3].

To compare the quality of clusters obtained by different clustering algorithms, we first performed an experiment on a subset of the Reuters-21578 dataset. This dataset contains 11413 documents with a vocabulary size of 20496 words. The data is known to belong to 90 classes and one other class. Table 1 shows the entropy and purity measures and the approximate times elapsed during clustering. In all cases, the number of clusters was set to 91 (as reflected by the manual labeling) and the quality of the clustering algorithms was compared. For PLSA, we ran EM algorithm for 10 iterations with two restarts. For LDA, we ran EM for 20 iterations, and for estimating hyper-parameters \( \alpha \) and \( \beta \) we used the method described in [3]. It can be seen from Table 1 that K-means, Bisecting K-means and Two-phase Bisecting K-means resulted in clusters with similar purities and entropies. However, PLSA and LDA, produced clusters that were not as pure as the first three algorithms with increased values for entropy. The computation time in this table reflects the efficiency of the clustering algorithms. It can be seen that PLSA and LDA are also the most time consuming of the clustering algorithms. Since Two-Phase Bisecting K-means was initialized with clusters from Bisecting K-means, even though it performs a two-step clustering procedure, it is still far more efficient (16 sec.) than K-means (150 sec). Based on these results, we decided to use the Two-Phase Bisecting K-means (2PKBM) and K-means algorithm, which were faster and more efficient, to cluster the data used to build language models for Large Vocabulary Speech Recognition Systems (LVCSR). We then study their impact on Word Error Rate (WER).

#### 3.1. Experiments on LVCSR tasks

We chose two LVCSR tasks to study the impact of automated clustering algorithms on language modeling, namely, transcription of Broadcast News in English and Arabic. Having determined the clustering algorithms to use, we experimented with the number of clusters to use for this set of LVCSR tasks. Since the language model training data is not manually annotated into clusters, entropy and purity could not be used as metrics to decide the optimal number of clusters. We used WER as our metric to determine the number of clusters.

The acoustic model for the English Broadcast News system [16] is trained on 450 hours of speech comprising the 1996 and 1997 English Broadcast News (BN) Speech collections and the English broadcast audio from TDT-4. This is a state-of-the-art English Broadcast News transcription system with a baseline WER of 13.4% on the rt04 evaluation test set (4 hours, 40K words) defined as part of the EARS program. The language model used to build the decoding graph is trained on a 192M word corpus comprising of several sources. The final language model is 4-gram, Kneser-Ney smoothed and has 3.2M n-grams. The vocabulary has 77K words. We use the test sets RT-03, Dev-04 and RT-05 as defined for the English portion of the EARS program, which after silence removal have lengths of 2:15, 2:00 and 4:00 hours respectively.

The Arabic acoustic models were trained on about 1200 hr. of Broadcast News and Broadcast Conversation speech data. The acoustic models were speaker-adapted and included both MPE and MPE training. Language model experiments were performed through re-scoring using lattices generated by a first-pass decoding. The general Arabic language model is an interpolated LM with interpolation weights optimized on a heldout set. The vocabulary size is 737K words. This state-of-the-art LVCSR system has a WER of 12.3%, 20.8% and 14.2% on the dev07, portion of eval06, and eval07 test sets released as part of the GALE program. These test sets are 2.6, 1.7, and 4.1 hours in duration and contain 18K, 12K, and 29K words. Initial experiments were conducted using the English BN
Table 2: WERs on English and Arabic test sets using Two-Phase Bisected K-means clustering

<table>
<thead>
<tr>
<th>Metric</th>
<th>English</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>13.3</td>
<td>12.3</td>
</tr>
<tr>
<td>dev07</td>
<td>20.6</td>
<td></td>
</tr>
<tr>
<td>eval06</td>
<td>14.1</td>
<td></td>
</tr>
</tbody>
</table>

system to fix an optimal number of clusters. Motivated by the number of topics in the Fisher corpus (40), we explored the use of several clusters ranging from 10 to 60. After discovering that the WERs ranged between 13.3% and 13.4% for all these choices, we decided to fix the number of clusters at 40 for the remaining experiments in English and Arabic. The data used for clustering was the language model data described earlier in this section. Language models were built for each of the clusters and interpolated with the baseline language model for all our experiments. Both word counts and TF-IDF were explored as input to the clustering algorithms. We present in Table 2, the best result we could obtain after experimenting with cluster sizes on the 2PBKM and K-means algorithm. This best result was achieved by the 2PBKM algorithm when the number of clusters was set to 40. It can be seen that we obtain an improvement of 0.2% absolute on some tasks while there is no change in WER on some other tasks. A detailed analysis of the errors and their relationship to the clusters identified is presented in Section 4.

4. Analysis

We begin our analysis by sorting the misrecognized words according to the number of times they are misrecognized. Table 3 contains a sample set of misrecognized words. The top of the sorted list, is occupied by mostly function words. The individual WER or each of these words is around 10%. However, their cumulative contribution to the WER is very high. The top 10 words in the sorted list account for as much as 17% to the overall WER. These are mostly monosyllable words that are easily confusable. The next set of words shown in Table 3 are those that occur towards the middle of sorted list of errors and are seen less frequently than the ones in the top. Their individual WERs vary between 5% and 70%. This takes the cumulative contribution to the WER of the top 200 misrecognized words to approximately 50%. However, it can be seen from the last few entries in Table 3 that content words such as breathe, salt, development, support account for the remaining WER. They occur least frequently and are misrecognized most of the time. Their individual WERs are therefore very high.

Figure 1 illustrates the cumulative percentage contribution from each of the misrecognized words in the test set. It can be seen that the stop words and function words are the most misrecognized words and cumulatively contribute between 60% (for the English BN system) to 40% (for the Arabic BN system) to the overall WER. The content words which are the last few entries we show in Table 3, contribute about 20% to the overall WER. In order to analyze if these words are indeed representative of topics, we computed the information gain [8] of these words, defined as $IG(w) = H(C) - P(w)H(C|w) - P(\bar{w})H(C|\bar{w})$. The information gain of a word measures the reduction in entropy of the document class distribution when the word is known to exist in the document. Words with higher information gain have more discriminative power between class labels. Table 4 shows examples of words with high information gain for some representative clusters of the English train-

Table 3: Examples of misrecognized words, sorted by error count. Error Count refers to the number of times the word was misrecognized. Reference Count = number of times the word occurs in the reference. Error(%) = error rate for the word. Cumulative Error (%) = cumulative error rate of all words prior to the current word in the sorted list

<table>
<thead>
<tr>
<th>Word</th>
<th>Error Count</th>
<th>Reference Count</th>
<th>Error(%)</th>
<th>Cumulative Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>douglass</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
<td>80.55</td>
</tr>
<tr>
<td>breathe</td>
<td>1</td>
<td>2</td>
<td>50.00</td>
<td>80.58</td>
</tr>
<tr>
<td>salt</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
<td>80.61</td>
</tr>
<tr>
<td>bubbas</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
<td>80.63</td>
</tr>
<tr>
<td>negate</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
<td>80.66</td>
</tr>
</tbody>
</table>

Table 4: Top words ranked by infogain for some representative clusters

<table>
<thead>
<tr>
<th>Word</th>
<th>Error Count</th>
<th>Reference Count</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>alvy, mulder, audry, scully, mulder’s</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>smokers, nicotine, zyban, smoker, nonsmokers</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>iraqis, uncom, iraqi, saddam, Hussein’s</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>gameplay, gamer’s, multiplayer, gamer, xbox</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>disciples, jesus, pentecost, christ, scripture</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
<tr>
<td>arafat, palestinians, natanyahu, israelis, ramallah</td>
<td>1</td>
<td>1</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Table 4: Top words ranked by infogain for some representative clusters

5. Conclusions

In this paper, we studied various unsupervised clustering algorithms on several corpora for the purpose of building topic-specific language models for large vocabulary speech recognition. We used a Reuters task with known class labels, we compared the clustering methods using quality metrics such as entropy and purity, as well as speed of clustering. Of the
techniques studied, the two-phase bisecting K-means method achieves good performance with relatively fast clustering speed. We performed several speech recognition experiments on both English and Arabic systems using the automatically derived topic-based language models. We obtained small WER improvements, comparable to previously published studies [4]. A careful analysis of the correlation between word error rate (WER) and the distribution of misrecognized words, including an information gain metric, was undertaken. Results indicate that words with high info-gain are either words that are already correctly recognized or so rare that they do not have a significant impact on the overall WER, explaining the modest gains we were able to achieve.

6. References


