ABSTRACT

In this paper, we propose an extension of the backoff word \(n\)-gram language model that allows a better likelihood estimation of unseen events. Instead of using the \((n-1)\)-gram to estimate the probability of an unseen \(n\)-gram, the proposed approach uses a class hierarchy to define a context which is more general than the unseen \(n\)-gram but more specific than the \((n-1)\)-gram. Each node in the hierarchy is a class containing all the words of the descendant nodes (classes). Hence, the closer a node is to the root, the more general the corresponding class is. Performance is evaluated both in terms of test perplexity and word error rate (WER) on a simplified WSJ database. Experiments show an improvement of more than 26\% on the unseen events perplexity.

1. INTRODUCTION

Data sparseness constitutes a crucial point to take into account when building language models (LM) for speech recognition. When enough data is available, word \(n\)-gram language models have proved extremely useful to estimate the likelihood of \(n\)-grams \((w_1, \ldots, w_n)\) that occur frequently. However, the estimation of the probability of low-frequency and unseen \(n\)-grams is still inherently difficult. The problem becomes more acute as the vocabulary size increases since the number of low-frequency and unseen \(n\)-grams increases considerably. Many approaches have been proposed to overcome the probability estimation problem of low-frequency \(n\)-grams. One of them is the class \(n\)-gram language model [1, 2]. Language models based on word classes are more compact and generalize better on unseen \(n\)-grams than standard word-based language models. Nevertheless, for large training corpus, word \(n\)-gram LMs are still better in capturing collocational relations between words. A simple class \(n\)-gram model is usually unable to capture the ambiguous nature of many words. Hence, a better approach is to find a set of classes that is general enough to better model unseen events, but specific enough to capture the ambiguous nature of words. One solution that springs to mind is to hierarchically cluster the vocabulary words, building a tree. The leaves represent individual words, while the nodes define clusters, or word classes: a node contains all the words of its descendant nodes. Hence, the closer a node is to the leaves, the more specific the corresponding class is. At the top of the tree, the root cluster contains all the words in the vocabulary. This tree can be then be used to balance generalization ability and word specificity when estimating the probability of low-frequency and unseen events.

In this paper we introduce a new approach for language modeling that estimates the probability of unseen events based on hierarchical word clustering. This approach can be considered as a generalization of the well-known backoff word \(n\)-gram language model. The backoff word \(n\)-gram model estimates the probability of any unseen word \(n\)-gram according to the word \((n-1)\)-gram [3]. In our approach, the probability of an unseen event is estimated using the most specific class of the tree that guarantees a minimum number of occurrence of this event, hence allowing accurate estimation of the probability. This approach allows us to take advantage of both the power of word \(n\)-grams for frequent events and the predictive power of class \(n\)-gram for unseen or rare events. The generalization is controlled by the class hierarchy, which is not the case with a simple class \(n\)-gram. While the idea of using classes to estimate the probability of unseen events in a backoff word \(n\)-gram model was proposed by many researchers [4, 5], the originality of our approach is the use a hierarchical clustering rather than a simple set of classes.

2. HIERARCHICAL WORD CLUSTERING ALGORITHM

The hierarchical word clustering algorithm proceeds in a top-down manner to cluster a vocabulary word set \(V\), and is controlled by two parameters: (1) the maximum number of descendant nodes (classes) \(C\) allowed at each node, (2) the minimum number of words \(K\) in one class \(O_c\): \(N(O_c) \geq K\). Starting at the root node, which contains a single cluster representing the whole vocabulary, we build a maximum number of \(C\) clusters (classes) to define the immediate child nodes of the root node (cf. Section 2.2). We then proceed recursively on each descendant node to grow the tree.

The algorithm stops when a predefined number of levels (depth) is reached or when the number of proposed classes for one node \(O_c\) is equal to 1 (\(C = 1\)) (cf. step 5 of Section 2.2). In this last case, \(O_c\) become the last node before the leaves; leaves representing the \(O_c\) word set will be connected to the node \(O_c\).

The criterion used to build the word tree is based on the work of S. Bai et al. [6] and uses a concept of minimum discriminative information. It is presented in the next section, followed by a detailed description of the word clustering algorithm.

2.1. Minimum Discriminative Information

The left and the right contexts are used to cluster a word set, meaning that words that occurs frequently in similar contexts should be assigned to the same class. The contextual information of the word \(w\), \(p(w|w)\), is estimated by the probability value of \(w\) given its neighboring words at distance \(d\). Given a set of \(V\) words, in the case of \(d = 1\), we have the following:

\[
p_1\{w\} = \{p_1\{w\}, p_1r\{w\}\}
\]

(1)

where

\[
p_1\{w\} = \{p(w_1, w), \ldots, p(w, w)\}
\]

(2)

and

\[
p_1r\{w\} = \{p_r(w, w_1), \ldots, p_r(w, w)\}.
\]

(3)
The terms \( p_l(w, w) \) and \( p_r(w, w) \) denote respectively the left-bigram and right-bigram probability for the word \( w \). Let \( p(w) \) denote the word probability of \( w \):

\[
p(w) = \sum_{v=1}^{V} p_l(w, w) = \sum_{v=1}^{V} p_r(w, w).
\]

(4)

The clustering algorithm is based on two principles. First, the word class can be determined by the class of its neighboring words (contextual information). Second, words with similar POS function are merged into the same class. Therefore, the problem becomes how to define the similarity of two words in terms of their POS function, or their contextual information.

The discriminative information between two words \( w_1 \) and \( w_2 \) is estimated as follows:

\[
D(w_1, w_2) = \sum_{v=1}^{V} p_l(w_1 | w_2) \log \frac{p_l(w_1 | w_2)}{p_l(w_1 | w_1)} + \sum_{v=1}^{V} p_r(w_2 | w_1) \log \frac{p_r(w_2 | w_1)}{p_r(w_2 | w_2)}
\]

(5)

which is also known as Kullback-Leibler distortion measure or relative entropy.

The objective of partitioning the vocabulary is to find a set of centroids \( \{ o_c \} \) for clusters \( \{ O_c \} \), \( c = 1, \ldots, C \) which leads to the minimum global discriminative information:

\[
GDI = \sum_{c=1}^{C} \sum_{i \in O_c} D(w_i, o_c)
\]

\[
= \sum_{v=1}^{V} \sum_{i=1}^{V} p_l(w_i | w) \log \frac{p_l(w_i | w)}{p_l(w_1 | w)} + \sum_{v=1}^{V} \sum_{i=1}^{V} p_r(w_i | w) \log \frac{p_r(w_i | w)}{p_r(w_2 | w)}
\]

(6)

The term \( H(w) \) is a constant independent of the partitioning. Hence, when the global discriminative information minimized, \( R(w) \) is maximized.

Each cluster \( O_c \) is represented by a centroid \( o_c \) which carries the common POS functions for the class. According to equation 1, \( p_l \{ w \} \) is defined as a vector of dimension \( 2 \cdot V \), whose first \( V \) components are based on the left context bigrams, and last \( V \) components are based on the last \( V \) right context bigrams. For simplicity, let us drop the left/right indices, and represent \( p_l \{ w \} \) as follows:

\[
p_l \{ w \} = \{ p(k | w), k = 1, \ldots, 2V \}
\]

(7)

Given a class \( O_c = \{ w_i, i = 1, \ldots, n_c \} \), the centroid of \( O_c, o_c = \{ o(k | o_c), k = 1, \ldots, 2V \} \) can be estimated by using the minimum distance rule [7]. In this paper to estimate the distance measuring order \( D(o, w) \), we use the following equation:

\[
o(k | o_c) = \frac{1}{n_c} \sum_{i=1}^{n_c} p(k | w_i)
\]

(8)

In other words, the centroid \( o_c \) of cluster \( O_c \) is defined as the mean of all context vectors \( p_l \{ w \} \) of the words \( w_i \) belonging to \( O_c \).

### 2.2. Word Clustering Algorithm

In this section, we present how to classify a word set in at most \( C \) classes, assuming that at least \( K \) words should appear in each class \( O_c, \quad N(O_c) > K \). In our case, \( K \) is set to 1. We start by computing the centroid \( o_c \) of the whole space (word set). An initial codebook is then built by assigning the \( C \) closest words into \( C \) clusters. The cluster centroids are then recomputed, and the process is iterated until the average distortion GDI converges.

The pseudo-code of the algorithm is as follows:

- step 1: start with an initial codebook;
- step 2: for each \( w_i, i = 1, \ldots, V \),
  - find the closest class \( O \) to \( w_i \) using equation 5 and add \( w_i \) to it;
- step 3: update the codebook with equation 8;
- step 4: if \( GDI > t \) then step go to 2
- step 5: if \( \exists O_c / N(O_c) < K \) then \( (C \leftarrow C - 1) \) and step 1, else stop.

Usually only a few iterations of the algorithm is required to achieve a fairly good result [6]. Since each word is characterized by the contextual statistical vector \( p_l \{ w \} \), the centroid of each class is easily found using equation 8, which represents the POS features of the class. The advantage of this algorithm is its simplicity to find the centroid; the cost of merging words or classes become less expensive. Once the \( C \) classes have been defined, the previous algorithm is recursively applied within each class to grow the tree.

### 3. BACKOFF HIERARCHICAL CLASS N-GRAM LANGUAGE MODEL

Classical backoff word \( n \)-gram models assume that the conditional probability of a word \( w \) given a history \( h \), \( p(w | h) \), is estimated according to the \( n - 1 \) precedent words in \( h \):

\[
P(w_i | h) = P(w_{i-1} | w_{i-n+1}).
\]

(9)

Using this approach, the probability of an unseen \( n \)-gram \( w_{i-n+1}^{i-1} \) is estimated according to a more general context which is the \( n-1 \)-gram \( w_{i-n+1}^{i-2} \):

\[
P(w_i | w_{i-n+1}^{i-1}) = \begin{cases} 
\tilde{P}(w_i | w_{i-n+1}^{i-2}) & \text{if } N(w_{i-n+1}^{i-2}) > 0 \\
\alpha(w_{i-n+1}^{i-2}) \tilde{P}(w_i | w_{i-n+1}^{i-2}) & \text{otherwise} 
\end{cases}
\]

(10)

where \( N(\cdot) \) denotes the frequency of the argument in the training data, \( \alpha(\cdot) \) is a normalizing constant [3], and \( \tilde{P}(\cdot) \) is estimated as:

\[
\tilde{P}(w_i | w_{i-n+1}^{i-2}) = d_{N(w_{i-n+1}^{i-2})} N(w_{i-n+1}^{i-2})
\]

(11)

where \( d_{N(\cdot)} \) is the Turing’s discount coefficient [3]. The normalizing constant \( \alpha(\cdot) \) in equation 10 is derived according to the following equation:

\[
\alpha(w_{i-n+1}^{i-2}) = \frac{1 - \sum_{w_i : N(w_{i-n+1}^{i-2}) > 0} \tilde{P}(w_i | w_{i-n+1}^{i-2})}{1 - \sum_{w_i : N(w_{i-n+1}^{i-2}) > 0} P(w_i | w_{i-n+1}^{i-2})}
\]

(12)

Instead, in our proposed approach, the conditional probability of an unseen \( n \)-gram \( P(w_i | w_{i-n+1}^{i-2}) \) is estimated according to a more specific context than the \( (n-1) \)-gram \( P(w_i | w_{i-n+2}) \). We suggest to use as context the class of the first word \( w_{i-n+1} \) followed by the other words:

\[
F(w_{i-n+1}, w_{i-2})
\]
where the function $F(x)$ represents the class (parent) of $x$ within the hierarchical word tree, where $x$ can be a class itself, or a single word, depending on where it is located in the tree (cf. Section 2). Hence, the probability $P(w_i|w_i^{j+1})$ is estimated as follows:

$$P(w_i|w_i^{j+1}) = \begin{cases} \tilde{P}(w_i|w_i^{j-1}) & \text{if } N(w_i^{j+1}) > 0 \\ \alpha'(w_i^{j-1})\tilde{P}(w_i|C_i^{j+1}, w_i^{j+1}) & \text{otherwise} \end{cases} \tag{13}$$

where $C_i^j$ denotes the $j^{th}$ parent of the word $w_i$:

$$C_i^j = F^{(j)}(w_i).$$

If the event $C_i^{j+1}, w_i^{j+1}$ is not found in the training data ($N(C_i^{j+1}, w_i^{j+1}) = 0$), we recursively use a more general context by going up one level in the hierarchical word clustering tree. This context is obtained by taking the parent of the first class in the hierarchy followed by the $n - 2$ last words:

$$P(w_i|C_i^{j+1}, w_i^{j+1}) = \begin{cases} \tilde{P}(w_i|C_i^{j+1}, w_i^{j+1}) & \text{if } N(C_i^{j+1}, w_i^{j+1}) > 0 \\ \alpha'(C_i^{j+1}, w_i^{j+1})P(w_i|C_i^{j+1}, w_i^{j+1}) & \text{otherwise} \end{cases} \tag{14}$$

where the normalizing constant $\alpha'(\cdot)$ is computed as follows, to guarantee that all probabilities sum to 1:

$$\alpha'(C_i^{j+1}, w_i^{j+1}) = \frac{\prod_{i}P(\tilde{w}_i^{j-1})}{\sum_{i}P(\tilde{w}_i^{j-1})} \tag{15}$$

As a result, the whole procedure provides a consistent way to compute the probability of a rare or unseen $n$-gram by backoff-off along the classes that are defined in the hierarchical word tree. If the parent of the class $C_i^{j+1}$ (respectively, the word $w_i^{j+1}$) is the class root, the context become the last $n - 2$ words, which is similar to the traditional back-off word $n$-gram model.

### 4. EXPERIMENTS

#### 4.1. Data Description

Experiments are performed on Wall Street Journal (WSJ) corpus. This database is divided into two sets: the training and the test sets. For LM purposes, the training set contains 19 million words, and the test set contains approximately 6000 words. Two vocabulary sizes are used: a first one containing 1000 words (1K) and a second one including 5000 words (5K), both built by selecting the most frequent words in the training data. Note that the 5K vocabulary leads to about 2.1% of out-of-vocabulary words on the test data, and in that regard differs substantially from the official WSJ 5K lexicon that was designed for a closed-set evaluation (no OOV words). The objective of using various vocabulary sizes is to investigate the impact of the proposed approach for various amount of unseen events. For ASR experiments, the word error rate (WER) has been evaluated on the 330 sentences of the si_ett05 evaluation set, using tied-state triphone acoustics built on the SI-84 database.

#### 4.2. Results

Perplexity is typically used to measure the performance of language models. It is therefore interesting to look at the perplexity obtained by the backoff hierarchical class $n$-gram models for different number of levels in the word tree. The number of levels in the hierarchy represents the depth of the word tree. The maximum number of direct descendant for one class is fixed to $6$: $C = 6$. Other experiments with different values of $C$ led to similar performance of the algorithm. Note that according to equation 14, only the probability of unseen events is different between the word $n$-gram model and hierarchical class $n$-gram model. To show the real impact of this new approach, the bigram test perplexity presented in figure 1 is computed only on the unseen events. The first plot is based on the 1K vocabulary, while the second plot uses the 5K vocabulary. A number of levels equal to 0 represents the backoff word bigram test perplexity, which is considered as the baseline.

![Fig. 1. Unseen events bigram test perplexity with different number of levels in the class hierarchy.](image)

Results presented in figure 1 show that the test perplexity decreases when the number of unseen events increases; the number of unseen events is equal to 23 with the 1K vocabulary, compared to 122 with the 5K one. With the 1K vocabulary, we observe an improvement of 1% (67801 vs. 66906). While more than 4% improvement is reported with the 5K vocabulary (75780 vs. 72692). This suggests that we do not need a large number of levels in the class hierarchy to improve upon the baseline. Experiments show that the best results are obtained with the number of levels in the class hierarchy equals to 2 or 3.

When using the 1K vocabulary, the perplexity on the whole test set is similar for the proposed approach and the baseline (51.6 vs. 51.7). The small difference in perplexity between the two approaches can be explained by the small number of unseen events in the test corpus (only 32 unseen events). However, when using the 5K vocabulary, the backoff word $n$-gram perplexity on the whole test set is equal to 109.9, compared to 108.5 obtained by the hierarchical class $n$-gram model. As expected, it suggests that as the vocabulary size increases, the proposed approach will outperform the baseline approach even more.

We report in figure 2 other experimental results using trigram language models. The number of unseen events increases to 320 for the 1K vocabulary and to 911 with the 5K one. In the case of trigrams, we observe a 17% improvement over the baseline on the
1K vocabulary (1361.2 vs. 1117.0) and more than 26% improvement on the 5K one (1986.9 vs. 1466.4). The obtained trigram results confirm that as the number of unseen events increases, the proposed approach improves the perplexity compared to the baseline, which makes it a promising approach for very large vocabulary applications. The perplexity on the whole test set, using the 1K vocabulary, is equal to 30.3 compared to 31.9 given by the word backoff model. When using the 5K vocabulary our approach show 8% improvement compared to the baseline one (51.2 vs. 55.5).

We confirm the fact that we obtain better improvement when the number of unseen events increase.

Fig. 2. Unseen events trigram test perplexity with different number of levels in the class hierarchy.

Experiments show that with a number of levels in the class hierarchy greater than 3 or 4, the test perplexity becomes worse than the baseline model. To understand the reason, we analyzed the unseen event probabilities at different levels of the class hierarchy. We noticed that compared to the backoff word \(n\)-gram model, the probability of unseen events using the lower levels of the class hierarchy usually increases, compared to those using higher (deeper) levels that tend to decrease. Hence, we think that a shallow word (few levels) tree should give better result compared to a deep one (many levels).

<table>
<thead>
<tr>
<th></th>
<th>W-bigram</th>
<th>W-trigram</th>
<th>H-bigram</th>
<th>H-trigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>34.1%</td>
<td>32.1%</td>
<td>34.2%</td>
<td>32.5%</td>
</tr>
<tr>
<td>5K</td>
<td>10.4%</td>
<td>8.4%</td>
<td>10.3%</td>
<td>8.2%</td>
</tr>
</tbody>
</table>

Table 1. WER on 1K and 5K vocabularies using word bigram, word trigram, hierarchical class bigram and hierarchical class trigram respectively

We performed preliminary speech recognition experiments using Bell Labs ASR system [8]. In Table 1, we report the WER obtained using a class hierarchy of two levels. Results show that there is no significant difference in performance between the baseline backoff word \(n\)-gram and the proposed approach; only 2% improvement in the WER is reached using the trigram with the 5K vocabulary. We believe these results can be explained by the small number of unseen events in this experimental setup (small vocabulary size of 1K and 5K). As a future work, we will explore this new approach on a much larger corpus.

5. CONCLUSION

We have proposed a new approach to estimate the probability of unseen events using a hierarchical backoff technique. The originality of this approach is the use of a class hierarchy that leads to a better estimation of the likelihood of unseen events compared to the traditional backoff. Experiments on WSJ database with two different vocabulary size (1K and 5K) show that the improvement of the test perplexity over the standard backoff approach is directly related to the total number of unseen events: 10% improvement with the 5K vocabulary compared to a marginal 2% improvement with the 1K vocabulary. Speech recognition results show only 2% improvement when the hierarchical class trigram model is used with the 5K vocabulary (8.2% vs. 8.4%). In the other experiments, the WER is still comparable to the baseline, which we attribute to the small number of unseen events in the database, and therefore the lack of room for any significant improvement. We believe however that we have illustrated the potential of the proposed approach, and that improvement of the WER is expected as the total number of unseen events becomes larger.

Future work will be based on larger vocabulary applications and more robust techniques to build the class hierarchy. Preliminary results on Switchboard corpus show more than 18% improvement on the bigram unseen events test perplexity, clearly illustrating the potential of the proposed approach.

6. REFERENCES