Hierarchical Class N-Gram Language Models: Towards Better Estimation of Unseen Events in Speech Recognition

Imed Zitouni, Olivier Siohan[*]*, Chin-Hui Lee*[†]

Bell Labs, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974, U.S.A.
zitouni@research.bell-labs.com

Abstract

In this paper, we show how a multi-level class hierarchy can be used to better estimate the likelihood of an unseen event. In classical backoff n-gram models, the (n-1)-gram model is used to estimate the probability of an unseen n-gram. In the approach we propose, we use a class hierarchy to define an appropriate 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. We also investigate in this paper the impact of the hierarchy depth and the Turing's discount coefficient on the performance of the model. We evaluate the backoff hierarchical n-gram models on WSJ database with two large vocabularies, $5,000$ and $20,000$ words. Experiments show up to 26% improvement on the unseen events perplexity and up to 12% improvement in the WER when a backoff hierarchical class trigram language model is used on an ASR test set with a relatively large number of unseen events.

1. Introduction

Language modeling is known to be a very important aspect of speech recognition. Currently, backoff word n-gram language models (LM) have proved extremely useful to estimate the likelihood of n-grams $(w_1, \dots w_n)$ 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 data become sparse and the vocabulary size increases since the number of low-frequency and unseen n-grams increases considerably. One of the approaches that can overcome the probability estimation problem of unseen n-grams event is the class n-gram language model [1]. These models 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.

The backoff hierarchical class n-gram models hierarchically cluster the vocabulary words, to build a word tree. The leaves represent individual words, while the nodes define clusters, or word classes: a node contains all the words of its descendant nodes. The tree is used to balance generalization ability and word specificity when estimating the probability of low-frequency and unseen events. Our approach estimates the probability of an unseen event using the most specific class of the tree that guarantees a minimum number of occurrences 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-grams for unseen or rare events. This approach was first introduced in [2]. Preliminary experiments shown in [2] are based on small databases and small vocabularies, which did not show the real impact of this new model. In this paper, we report new results using large databases as well as large vocabularies: $5,000$ and $20,000$ words. We also investigate in this paper the different parameters of the model to clearly show how it better estimates the probability of low-frequency and unseen events. We study the influence of the hierarchy depth as well as the impact of the Turing's discount coefficient [3] on the backoff hierarchical class n-gram models. We discuss how the approach is sensitive to the number of levels in the class hierarchy: with large number of levels in the class hierarchy, the model become less accurate than the baseline n-gram models. We also show, like in the case of baseline n-gram models [3], how the backoff hierarchical n-gram models are not sensitive to the Turing's discount coefficient. 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. Backoff Hierarchical Class n-gram language model

In the classical backoff word n-gram models, the probability of an unseen n-gram $w_{i-n+1}^{i-1} = w_{i-n+1}, \dots w_{i-1}$ is estimated according to a more general context, which is the (n-1)-gram (w_{i-n+2}^{i-1}) [3]. Instead, in our proposed approach, the conditional probability of an unseen n-gram $P(w_i|w_{i-n+1}^{i-1})$ is estimated according to a more specific context than the (n-1)-gram $P(w_i|w_{i-n+2}^{i-1})$. We suggest to use as context the class of the first word w_{i-n+1} followed by the other words:

$$F(w_{i-n+1}), w_{i-n+2}^{i-1},$$

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 3). Let C_i^j denotes the j^{th} parent of the word w_i:

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

The probability $P(w_i|w_{i-n+1}^{i-1})$ is estimated as follows:

$$P(w_i|w_{i-n+1}^{i-1}) = \begin{cases} \tilde{P}(w_i|w_{i-n+1}^{i-1}) & \textbf{if } N(w_{i-n+1}^{i}) > 0 \\ \alpha'(w_{i-n+1}^{i-1}) P(w_i|C_{i-n+1}^{1}, w_{i-n+2}^{i-1}) & \textbf{otherwise} \end{cases} \quad (1)$$

[*] Now with IBM T.J. Watson Research Center, e-mail contact at siohan@us.ibm.com.

[†] Now a professor at Georgia Institute of Technology, email contact at chl@ece.gatech.edu.

where $N(.)$ denotes the frequency of the argument in the training data, and $\tilde{P}(.)$ is estimated as:

$$\tilde{P}(w_i|w_{i-n+1}^{i-1}) = d_{N(w_{i-n+1}^{i})} \frac{N(w_{i-n+1}^{i})}{N(w_{i-n+1}^{i-1})}. \quad (2)$$

The term $d_{N(.)}$ denotes the Turing's discount coefficient [3]. If the event $C_{i-n+1}^{j}, w_{i-n+2}^{i}$ is not found in the training data

$$N(C_{i-n+1}^{j}, w_{i-n+2}^{i}) = 0,$$

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:

$$P(w_i|C_{i-n+1}^{j}, w_{i-n+2}^{i-1}) = \begin{cases} \tilde{P}(w_i|C_{i-n+1}^{j}, w_{i-n+2}^{i-1}) & \textbf{if } N(C_{i-n+1}^{j}, w_{i-n+2}^{i}) > 0 \\ \alpha'(C_{i-n+1}^{j}, w_{i-n+2}^{i-1}) P(w_i|w_{i-n+2}^{i-1}) & \textbf{if } C_{i-n+1}^{j+1} \textbf{ is the root} \\ \alpha'(C_{i-n+1}^{j}, w_{i-n+2}^{i-1}) P(w_i|C_{i-n+1}^{j+1}, w_{i-n+2}^{i-1}) & \textbf{otherwise} \end{cases} \quad (3)$$

where the normalizing constant $\alpha'(.)$ is computed to guarantee that all probabilities sum to 1 [2].

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

2.1. Turing's discount coefficient

Usually, word backoff n-gram language models shall leave intact the estimate count for the probability of all unseen n-grams. It shall also not discount high values of counts $r > k$, considering them as reliable. To achieve this, the discount coefficient d_r (cf. equation 2) is equal to 1 when $r > k$:

$$d_r = \begin{cases} 1 & \textbf{for } r > k \\ \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} & \textbf{for } 1 \leq r \leq k \end{cases} \quad (4)$$

where $r^* = (r+1)\frac{n_{r+1}}{n_r}$. The term n_r denotes the number of n-grams which occur exactly r times in the training set. As for the parameter k, in practice, a value close to $k = 7$ is a good choice. In the case of the hierarchical language model, the parameter k is set up to the one used at the word level and the discount values (d_r) are computed as in the baseline n-gram language model (cf. equation 4).

3. 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 (clusters) 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 to define the immediate child nodes of the root node. We then continue the process 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 clusters for one node O_c is equal to 1 ($C = 1$). 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.

3.1. Minimum Discriminative Information

The clustering algorithm is based on two principles. First, words with similar POS function are merged into the same cluster. Second, the word cluster can be determined by the cluster of its neighboring words (contextual information). The contextual information of the word w, $p\{w\}$, is estimated by the probability value of w given its right and left context bigrams. To define the similarity of two words w_1 and w_2 in terms of their POS function or their contextual information, we use the Kullback-Leibler distortion measure $D(w_1, w_2)$ as defined in [2].

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:

$$GDI = \sum_{c=1}^{C} \sum_{i \in O_c} D(w_i, o_c) \quad (5)$$

Each cluster O_c is represented by a centroid o_c, which carries the common POS functions for the cluster. The centroid of O_c is estimated by using the minimum distance rule [7, 2]. Since we are working in a discrete space, o_c might not be a valid word. Hence, a pseudo-centroid of a cluster O_c can be found by looking for the closest word to o_c. The reader may refer to [2, 7] for more details regarding the estimation of these parameters.

3.2. 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: $N(O_c) > K$. In our case, K is set to 2. We start by computing the centroid o_i of the whole space (word set). An initial codebook is then built by assigning the C closest words to o_i 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 Kullback-Leibler distortion measure and add w_i to it [2].
- step 3: update the codebook using the minimum distance rule [2, 7];
- step 4: **if** $GDI > t$ **then** go to 2
  - { where t is a threshold used to terminate the convergent process.
- step 5: **if** $\exists\ O_c\ /\ N(O_c) < K$ **then** ($C \leftarrow C - 1$) and go to 1, **else** stop.

Usually only a few iterations of the algorithm is required to achieve a fairly good result [6]. Once the C classes have been defined, the previous algorithm is recursively applied within each class to grow the tree.

3.3. Initial codebook processing

The choice of the initial codebook seems to be not very crucial for the clustering algorithm (cf. step 1 in section 3.2). We did two experiments using different initial codebooks: we first compute the centroid of the word set to be clustered and then we choose the C words which are the furthest away from this centroid, building the initial centroids of the C classes. In the second experiment, we choose the C closest words to the word set centroid to build the initial codebook. In these two experiences, we found that the clustering algorithm almost converges to the same set of classes.

4. Experiments

Performance is evaluated in terms of test perplexity and in terms of word error rate (WER) obtained with our speech recognizer [8]. A comparison between the approach presented in this paper and the backoff word n-gram language models will be shown. We also investigate the impact of the number of levels in the class hierarchy on the performance of this approach.

4.1. Data Description

Experiments are performed on Wall Street Journal 94-96 text (WSJ) corpus. This database is divided into two sets: the training and the test sets. For LM purposes, the training set contains 56 million words, and the test set contains approximately 6 million words. Two vocabulary sizes are used: a first one containing 5,000 words ($5K$) and a second one including 20,000 words ($20K$). Note that the $5K$ vocabulary leads to about 2% 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 amounts of unseen events. For ASR experiments, the word error rate (WER) on 5K has been evaluated on the 330 sentences of the si_et_05 evaluation set. The 333 sentences of the si_et_20 evaluation set were used for the 20K ASR experiment. We used tied-state triphone acoustic models built on the WSJ 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 $C = 6$. Other experiments with different values of C led to similar performance. Note that according to equation 3, 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. We present in this Figure the unseen event test perplexity as a function of the number of levels in the class hierarchy. The first plot is based on the $5K$ vocabulary, while the second plot uses the $20K$ vocabulary. A number of levels equal to 0 represents the backoff word bigram test perplexity, which is considered as the baseline.

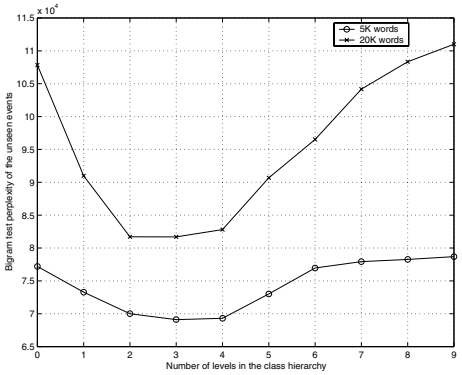

Figure 1: Unseen event bigram test perplexity with different number of levels in the class hierarchy.

The number of unseen events is approximately equal to 100,000 with the $5K$ vocabulary, compared to 300,000 with the $20K$ one. With the $5K$ vocabulary, we observe an improvement of 10% (77198 vs. 69111). More than 24% improvement is reported with the $20K$ vocabulary (107824 vs. 81689). Hence, results show that the test perplexity decreases when the number of unseen events increases. Experimental results also suggest that only few numbers of levels (e.g., 3 or 4) are required in the class hierarchy to yield improvements over the baseline. The impact of the hierarchy depth on the accuracy of the model will be discussed in section 4.3.

When using the $5K$ vocabulary, 3% improvement of the perplexity on the whole test set is observed (100.3 vs. 103.0). When using the $20K$ vocabulary, the backoff bigram perplexity on the whole test set is equal to 216.7, compared to 210.6 obtained by the hierarchical class bigram model. As expected, the proposed approach still outperforms the baseline approach even with a large vocabulary.

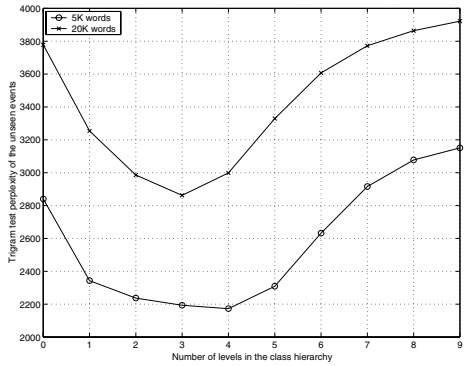

Figure 2: Unseen event trigram test perplexity with different number of levels in the class hierarchy.

We report in figure 2 other experimental results using trigram language model. We present in this Figure the trigram unseen event test perplexity as a function of the number of levels in the class hierarchy. The number of unseen events increases to approximately $900,000$ for the $5K$ vocabulary and to approximately 1 million for the $20K$ one. In the case of trigram, we observe a 23% improvement of the unseen event perplexity over the baseline on the $5K$ vocabulary (2840 vs. 2172) and more than 24% improvement on the $20K$ one (3778 vs. 2862). The obtained trigram results confirm that as the number of unseen events increases, the proposed approach improves the perplexity compared to the baseline, making it a promising approach for applications using sparse data. The perplexity on the whole test set, using $5K$ or $20K$ vocabularies, improves by 6% approximately (from 69.1 to 64.6 for the $5K$ vocabulary, and from 140.8 to 132.2 for the $20K$ one). Compared to the bigram, more unseen events were observed and consequently more improvement was obtained (6% vs. 3%).

4.3. Influence of the hierarchy depth

We first remind that a number of levels in the class hierarchy (X-axis) equal to 0 in Figures 1 and 2 represents the baseline word backoff n-gram models. Experiments reported in these two figures show that with a large number (e.g., 7) of levels in the class hierarchy, the test perplexity becomes worse than the baseline model test perplexity. To understand the reason, we analyzed the unseen event probabilities at different levels of the class hierarchy. We noticed that 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. A higher level in the class hierarchy consists in a more general model.

Let $\beta(w_1^{n-1})$ denotes the probability sum of all unseen events:

$$\beta(w_1^{n-1}) = \sum_{w_n : N(w_1^n)>0} \tilde{P}(w_n|w_1^{n-1}). \quad (6)$$

Usually, in the baseline n-gram language models, most of $\beta(w_1^{n-1})$ is distributed among the (n-1)-grams and only a small fraction of it is assigned to the more general models: (n-2)-grams, (n-3)-grams, etc. We observed the same behavior with the hierarchical language model: most of $\beta(w_1^{n-1})$ is distributed among the lower levels in the class hierarchy and a small part go to the higher levels, (n-1)-grams, etc. In the case of a baseline backoff 3-gram language model, we may not notice this phenomenon since only two levels are available: bigram and unigram distributions. Notice that for the hierarchical class n-gram language models, the (n-1)-gram is obtained when we move up to the root in the class hierarchy.

Consequently, the depth of the hierarchy is an important point to consider when building the model. A very flat hierarchy can results in an unwanted over-generalization. On the other hand, a too deep tree can leads to poor generalization for some unseen events. We think that a shallow word (few levels) tree should give better result compared to a deep one (many levels).

4.4. Impact of the Turing's discount coefficient

At the word class hierarchy level (tree node), the hierarchical language model sets the parameter k of Eq. 4 to the one used at the word level ($k = 7$). The discount values (d_r) are then computed as in the baseline n-gram language model (cf. Eq. 4). We denote this as the $E1$ experiment. However, to study the impact of the discounting values d_r on our approach, three other experiments were carried out. In these experiments, a constant is assigned to the discount value d_r of the class hierarchy level and three different values are assigned to the parameter k: the one used at the word level, the maximum occurrence count, and the average occurrence count. We denote those experiments $E2$, $E3$ and $E4$, respectively. In $E2$, $E3$ and $E4$, the Tuning's discount value d_r is set to 0.95. Experiments with different values of d_r varying from 0.8 to 0.99 give approximately the same result: less than one point difference in terms of perplexity. Another experiment $E5$ was also carried out using the word level discount coefficient d_r at the class hierarchy level (tree node).

	E1	E2	E3	E4	E5
PP	100.3	100.4	100.4	100.5	99.8

Table 1: Perplexity on WSJ $5K$ vocabulary using different methods in computing the Turing's discount value

Table 1 shows the bigram perplexity obtained with WSJ 5K corpus. Results show that the model is not sensitive to the value of d_r: only small differences in terms of perplexity are observed. Consequently, we arrive at the same conclusion as Katz [3] for the baseline n-gram language model: the hierarchical class n-gram language model is not sensitive to the value of d_r.

4.5. ASR Experiments

The speech recognition experiments were performed using Bell Labs ASR system [8]. We remind that the $5K$ vocabulary differs from the official WSJ $5K$ lexicon that was designed for a closed-set evaluation (cf. §4.1). In table 2, we report the WER obtained using a class hierarchy of two levels. Results show that there is no significant improvement in performance between the baseline backoff bigram model and the hierarchical class bigram model. These results can be explained by the small number of unseen bigrams in this experimental setup and therefore the lack of room for any significant improvement: only 4% and 8% of unseen bigrams on the $5K$ and $20K$ vocabularies respectively. However, when the trigram is used the number of unseen events increases to 27% for the $5K$ vocabulary and to 34% for the $20K$ vocabulary, resulting in a considerable relative improvement of the WER: 12% and 10% respectively. This confirms our thought that the hierarchical class n-gram language models act better when the number of unseen events increases.

	5K		**20K**	
	bigram	trigram	bigram	trigram
Baseline	9.3%	7.6%	14.2%	12.4%
HLM	9.0%	6.7%	13.9%	11.2%

Table 2: WER on $5K$ and $20K$ vocabularies using word bigram, word trigram, hierarchical class bigram and hierarchical class trigram (HLM) respectively

5. Conclusion

We have discussed the potential of the backoff hierarchical class n-gram models to estimate the probability of unseen events. Compared to the traditional backoff, the originality of this approach is in the use of a class hierarchy that leads to a better estimation of the likelihood of unseen events. We also concluded that our approach is not sensitive to the Turing's discount coefficient, like the traditional backoff n-gram models. On the other hand, the depth of the hierarchy is an important point to be considered when building the model. A tree with few levels should give better result compared to a deep one. Experiments on WSJ database with two different vocabulary sizes ($5K$ and $20K$) show that the improvement of the test perplexity over the standard backoff approach is directly related to the total number of unseen events: 24% improvement with the $20K$ vocabulary compared to 10% improvement with the $5K$ vocabulary. Speech recognition results are also sensitive to the number of unseen events: up to 12% improvement in the WER when the hierarchical class trigram model is used, due to the large number of unseen events in the ASR test set. As a future work, we will explore this new approach with a much more accurate tree and on a much more sparser corpus such as Switchboard and CALLHOME databases.

6. References

[1] B. Suhm and W. Waibel, "Towards better language models for spontaneous speech," in *Proc. ICSLP-1994*, 1994.

[2] I. Zitouni, O. Siohan, H-K.J. Kuo, and C-H. Lee, "Backoff hierarchical class n-gram language modelling for automatic speech recognition systems," in *Proc. ICSLP-2002*, Denver, USA, 2002.

[3] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 35, no. 3, 1987.

[4] J.W. Miller and F. Alleva, "Evaluation of a language model using a clustered model backoff," in *Proc. ICSLP-1996*, 1996.

[5] C. Samuelsson and W. Reichl, "A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics," in *Proc. ICASSP-1999*, 1999.

[6] S. Bai, H. Li, Z. Lin, and B. Yuan, "Building class-based language models with contextual statistics," in *Proc. ICASSP-1998*, 1998.

[7] H. Li, J.P Haton, J. Su, and Y. Gong, "Speaker recognition with temperal transition models," in *Eurospeech-95*, Madrid, Spain, 1995.

[8] Q. Zhou and W. Chou, "An approach to continuous speech recognition based on self-adjusting decoding graph," in *Proc. ICASSP-1997*, 1997, pp. 1779–1782.