HIERARCHICAL LINEAR DISCOUNTING CLASS N-GRAM LANGUAGE MODELS: A MULTILEVEL CLASS HIERARCHY APPROACH

Imed Zitouni

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
izitouni@us.ibm.com

Qiru Zhou

Bell Labs, Alcatel-Lucent
Murray Hill, NJ 07974
qzhou@alcatel-lucent.com

ABSTRACT

We introduce in this paper a hierarchical linear discounting class n-gram language modeling technique that has the advantage of combining several language models, trained at different nodes in a class hierarchy. The approach hierarchically clusters the word vocabulary into a word-tree. The closer a tree node is to the leaves, the more specific the corresponding word class is. The tree is used to balance generalization ability and word specificity when estimating the likelihood of an n-gram event. Experiments are conducted on Wall Street Journal corpus using a vocabulary of 20,000 words. Results show a reduction on the test perplexity over the standard n-gram approaches by 10%. We also report considerable improvement in the accuracy of the speech recognition task.

Index Terms— Language Modeling, n-gram, Class Hierarchy, Linear Distortion

1. INTRODUCTION

Estimating the probability of low-frequency and unseen n-grams is still inherently difficult when using the popular n-gram language models (LMs). The problem becomes more acute as the vocabulary size increases since the number of low-frequency and unseen n-grams events increases considerably. The class n-gram LMs [1] is one approach to overcome this problem. Class n-gram LMs are more compact and generalize better on unseen n-grams compared to standard word-based LMs. Nevertheless, for large training corpora, word n-gram LMs are still better than class-based LMs. A better approach is to build a language model that is general enough to better model unseen events, but specific enough to capture the ambiguous nature of words. Our solution is the hierarchical linear discounting class n-gram LMs. This approach combines the power of word n-grams for frequent events and the predictive power of class n-grams for unseen and rare events. It linearly interpolates different n-gram LMs each one of them is trained on one level of a word tree, obtained by clustering hierarchically the word vocabulary. The leaves represent individual words, while the nodes define clusters, or word classes: a node contains all the words of its descendant nodes. The closer a node is to the leaves, the more specific the corresponding class is. The tree is used to balance generalization ability and word specificity when estimating the probability of n-gram events. The model trained on the leaves level (level 0) is the standard word n-gram LMs. Those LMs trained on a level in the class hierarchy greater than 0 are in fact the class n-gram LMs. The higher the number of levels in the class hierarchy is, the more compact and general the class n-gram LMs become.

We recently presented the backoff hierarchical class n-gram LMs (HCLMs) that also uses a class hierarchy [2] . HCLMs are a generalization of the backoff word n-gram LMs [3]. When using HCLMs, the probability of an unseen n-gram (w_{i-n+1}^{i}) is computed according to a more specific context than the (n-1)-gram: we use the class of the most distant word w_{i-n+1} followed by the other words, $F(w_{i-n+1}), w_{i-n+2}^{i-1}$; $F()$ represents the class (parent) of x within the class hierarchical. The approach presented in this paper proceeds differently on the class hierarchy. It is based on the linear interpolation smoothing technique [4], rather than the backoff technique used by the HCLMs [2]. It has the advantage of letting several LMs, trained at different nodes in the class hierarchy, to contribute to estimate the likelihood of an event, whether it is frequent, rare or unknown. This is different from HCLMs that rely on a LM in a higher node in the hierarchy *only* if the n-gram event is unknown.

The use of word tree for LMs was proposed by several scientists [5, 6, 7, 8]. L. Bahl *et al.* proposed a tree-based LM where a linear interpolation is used to smooth the relative frequency at each node of the tree [5]: the likelihood of an n-grams is computed as the linear interpolation of several word class LMs. This approach has some similarities with the technique we propose here. The main difference is in the manner the interpolation scheme is used, the way active nodes in the tree are selected, and the technique we use to build the class hierarchy.

2. HIERARCHICAL LINEAR DISCOUNTING CLASS N-GRAM LANGUAGE MODELS

When using classical linearly interpolated word n-gram models, more general (n-1)-gram distribution is used to estimate the n-gram distribution. Linear interpolation can be seen as a

smoothing approach that allows the combination of an arbitrary number of distribution or even LMs. Using a recursive representation, the classical linearly interpolated word n-gram LMs estimate the conditional probability of a word w_i given a history $w_{i-n+1}^{i-1} = (w_{i-n+1}, ..., w_{i-1})$, according to the (n-1)-gram distribution as follows:

$$P(w_i|w_{i-n+1}^{i-1}) = (1 - \lambda(w_{i-n+1}^{i-1}))fr(w_i|w_{i-n+1}^{i-1}) + \lambda(w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1}) \quad (1)$$

where $\lambda()$ is positive and denotes the zero-frequency probability. The term $fr()$ denotes the relative frequency of the n-gram between parenthesis. The most known and original version of the linear interpolated trigram LM [9] was not defined recursively as described in equation 1. It was presented as a linear combination of all order empirical distributions, but it is still the same technique.

To better explore the power of word n-grams for frequent events and the predictive power of class n-grams for unseen or rare events, we propose the hierarchical linear discounting class n-gram LMs, denoted HLDC LMs. HLDC LMs combine discounting and redistribution according to the linear interpolation smoothing technique [4]. These models estimate the conditional probability of an n-gram w_{i-n+1}^{i}, $P(w_i|w_{i-n+1}^{i-1})$ according to more general distribution extracted from the class hierarchy; we use the class of the most distant word w_{i-n+1} followed by the other words: $F(w_{i-n+1}), w_{i-n+2}^{i-1}$. 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 its location in the tree (cf. Section 3).

Let F_i^j denote the j^{th} parent of word w_i: $F_i^j = F^{(j)}(w_i)$. The probability $P(w_i|w_{i-n+1}^{i-1})$ is estimated as follows:

$$P(w_i|w_{i-n+1}^{i-1}) = (1 - \lambda(w_{i-n+1}^{i-1}))fr(w_i|w_{i-n+1}^{i-1}) + \lambda(w_{i-n+1}^{i-1})P(w_i|F_{i-n+1}^{1}, w_{i-n+2}^{i-1}) \quad (2)$$

where $P(w_i|F_{i-n+1}^{j}, w_{i-n+2}^{i-1})$ is recursively estimated according to more general distribution by going up one level at a time in the hierarchical word clustering tree:

$$P(w_i|F_{i-n+1}^{j}, w_{i-n+2}^{i-1}) = \begin{cases} (1 - \lambda(F_{i-n+1}^{j}, w_{i-n+2}^{i-1}))fr(w_i|F_{i-n+1}^{j}, w_{i-n+2}^{i-1}) + \\ \lambda(F_{i-n+1}^{j}, w_{i-n+2}^{i-1})P(w_i|w_{i-n+2}^{i-1}) \\ \qquad \textbf{if } F_{i-n+1}^{j+1} \textbf{ is the root} \\ (1 - \lambda(F_{i-n+1}^{j}, w_{i-n+2}^{i-1}))fr(w_i|F_{i-n+1}^{j}, w_{i-n+2}^{i-1}) + \\ \lambda(F_{i-n+1}^{j}, w_{i-n+2}^{i-1})P(w_i|F_{i-n+1}^{j+1}, w_{i-n+2}^{i-1}) \\ \qquad \textbf{otherwise} \end{cases} \quad (3)$$

As a result, the whole procedure provides a consistent way to compute the probability of any n-gram event by exploring the classes that are in the hierarchical word tree. If the parent of the class F_{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 linearly interpolated word n-gram models as described in equation 1. Based on this definition, HLDC LMs are a generalization of the classical linearly interpolated word n-gram LMs: linear interpolated word n-gram LMs can be seen as HLDC LMs with a single level (leaves) in the hierarchical word tree.

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.* [10] 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:

$$GDI = \sum_{c=1}^{C} \sum_{i \in O_c} D(w_i, o_c) \quad (4)$$

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 [11, 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, 11] for more details regarding the estimation of these parameters.

3.2. Word Clustering Algorithm

The standard word clustering we use in this paper is similar to the one used previously in [2]. We assume that a word

set is to be split into at most C classes, and 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 [10]. 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 [10].
- step 3: update the codebook using the minimum distance or nearest neighbor rule [2, 11];
- step 4: **if** $GDI > t$ **then** go to step 2
  - where t is an experimentally tuned threshold controlling the convergence of the process; the current set of clusters may leads to the minimum global discriminative information (cf. equation 4).
- step 5: **if** $\exists\ O_c\ /\ N(O_c) < K$ **then** $(C \leftarrow C - 1)$ and go to 1, **else** stop.

4. EXPERIMENTS

Experiments are performed on the Wall Street Journal 94-96 text corpus with a vocabulary that includes 20,000 words ($20K$). The $20K$ vocabulary has a 1.1% out-of-vocabulary rate on the test data. This database is divided into training, development and test sets. For language modeling purposes, the training set contains 56 million words, and the test set contains approximately 6 million words. A development set of 5 million words is also used to tune the different parameters of the model, including the depth of the clustering tree. Performance is evaluated in terms of test perplexity and word error rate (WER) produced by our speech recognizer [12].

Figure 1 presents the performance of word n-gram LMs and HLDC LMs with different number of levels in the class hierarchy. The maximum number of direct descendant of a class is fixed to $C = 6$ (cf. section 3). We remind that word n-gram LMs estimated with the linear distortion technique [4] are the HLDC LMs with a number of levels in the class hierarchy equal to 0. Hence, we believe that it is fair to consider word n-gram LMs as baselines for comparison purpose. Figure 1 shows also the performance of the HCLMs approach as described in [2]. As a reminder, HCLMs are a generalization of the backoff word n-gram LMs that uses a class hierarchy. When the n-gram is unknown, HCLMs backoff to the class of the most distant word w_{i-n+1} followed by the other words. One of the main differences with HLDC LMs is that HCLMs uses the class hierarchy *only* if the n-gram is unknown. This is different from HLDC LMs that benefit from the different nodes in the class hierarchy when estimating the probability of an event. Results show that we do not need a large number of levels in the class hierarchy to converge and improve upon the baseline: three or four levels are enough to achieve a good performance. Figure 1 shows a 7% improvement for bigrams (202.8 vs. 216.7) and 10% improvement for trigrams (127.5 vs. 140.8) over the word n-gram models. It also shows that HLDC LMs outperform HCLMs by 4% (202.8 vs. 210.6 for bigram and 127.5 vs. 132.2 for trigram). We notice that HLDC LMs are less sensitive to the number of levels in the class hierarchy: the perplexity value of the HCLMs decreased for the first three levels and then it starts to increase. However, the HLDC LMs converge to its optimum with four levels in the class hierarchy and doesn't increase afterward.

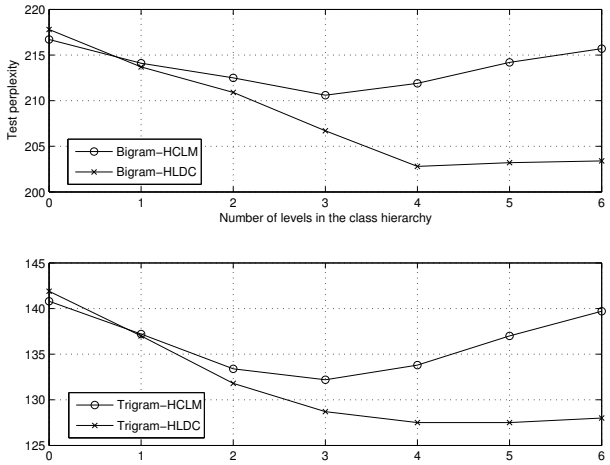

Fig. 1. Trigram and bigram test perplexity with different number of levels in the class hierarchy.

We also compared HLDC LMs to word class n-gram (n-class) LMs as well as a linear interpolation between word n-gram and n-class LMs [9]. In the n-class models, the conditional probability of the n-gram $w_{i-n+1}^{i} = w_{i-n+1}, \ldots w_i$, is estimated as follows:

$$P(w_i|w_{i-n+1}^{i-1}) = P(w_i|F(w_{i-n+1}), \ldots F(w_{i-1})). \quad (5)$$

where the function $F(x)$ represents the class of x. Notice that the HLDC LMs is able to integrate any classification approach for building the class hierarchy. In order to make a fair comparison between the proposed hierarchical approach and the n-class LMs, we *should* use the same classification technique. Hence, to build the class set, we use the MDI approach (§section 3.2) that assigns each word to a unique class. We initialize the MDI classifier with a maximum number of classes equal to 1200 assuming that one class should contains at least 5 words. We present in table 1, the perplexity values obtained by the different LMs on the entire test set.

	Bigram	Trigram
Class	228.9	154.0
Word (Baseline)	216.7	140.8
LI (Word + Class)	215.8	138.6
HLDC	202.8	127.5

Table 1. Perplexity of world class n-gram LMs (Class), word n-gram LMs (Word), linear interpolation of word n-gram and n-class LMs (LI), and HLDC LMs.

As expected, the perplexity of the baseline word n-gram LMs is better than the class word n-gram LMs: 216.7 vs. 228.9 for the bigram model and 140.8 vs. 154.0 for the trigram model with the 20K vocabulary. Also, compared to the baseline word n-gram LMs, we notice that a linear interpolation of word n-gram LMs and n-class LMs doesn't led to a considerable improvement (215.8 vs. 216.7 for bigram and 138.6 vs. 140.8 for trigram on 20K vocabulary). On both bigram and trigram, results show that the proposed hierarchical LMs outperform the other approaches.

4.1. Speech Recognition Experiments

For ASR experiments, the word error rate (WER) on the 20K WSJ has been evaluated on the 333 sentences of the si_et_20 evaluation set. The speech recognition experiments were performed using the ASR system described in [12]. We gave equivalent setting to the pruning parameters to make sure that the decoder search doesn't favor one model over another. Results presented in Table 2 show that there is no significant improvement in performance between the baseline bigram model, and HLDC bigram LM. 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: unseen bigrams constitute 8% of the total bigrams. However, when the trigram model is used, the number of unseen events increases to 34%, leading to 10% reduction of the WER.

	20K	
	bigram	trigram
Baseline	14.2%	12.4%
LI (word + class)	14.1%	12.0%
HCLM	13.9%	11.2%
HLDC	13.8%	11.0%

Table 2. WER using word n-gram, linear interpolation between word and class n-gram (LI), HCLMs, and HLDC LMs.

5. CONCLUSION

Compared to traditional n-gram LMs, the originality of the approach introduced in this paper is in the use of a class hierarchy that leads to a better estimation of the likelihood of n-gram events. Experiments show that the HLDC LMs improve the test perplexity over the standard language modeling approaches: 10% improvement on trigram events. Speech recognition results show up to 12% reduction of the WER when using HLDC LMs. The magnitude of the WER reduction is larger than what we would have expected given the observed reduction of the language model perplexity. A prelimenay investigation and analysis of errors shows that the HLDC LMs is more effective on unseen events where the acoustic model is not able to well discriminate between words. This leads us to an interesting assumption that the reduction of unseen event perplexity, where the acoustic model is not able to well discriminate between words, is more effective for improving ASR accuracy. More work is required to be able to confirm this.

6. REFERENCES

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

[2] I. Zitouni, "Backoff hierarchical class n-gram language models: Effectiveness to model unseen events in speech recognition," *Journal of Computer Speech and Language, Academic Press*, vol. 21, no. 1, pp. 88–104, 2007.

[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] Renato DeMori, Ed., *Spoken Dialogues with Computers*, Academic Press, 1998.

[5] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "A tree-based statistical language model for natural language speech recognition," in *IEEE Transaction on Acoustics, Speech and Signal Processing*, July 1987, vol. 37, pp. 1001–1008.

[6] P.A. Heeman, "Pos tags and decision trees for language modeling," in *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, Maryland, June 1999, pp. 129–137.

[7] J.A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proceeding of HLT/NAACL*, Canada, May 2003.

[8] 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.

[9] F. Jelinek, "Self-organized language modeling for speech recognition," *Readings in Speech Recognition, A. Waibel and K-F. Lee editors*, pp. 450–506, Morgan Kaufmann, San Mateo, Calif., 1990.

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

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

[12] 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.