THE USE OF WORD N-GRAMS AND PARTS OF SPEECH FOR HIERARCHICAL CLUSTER LANGUAGE MODELING

Wen Wang and Dimitra Vergyri

Speech Technology and Research Lab
SRI International, Menlo Park, CA 94025, USA
{wwang,dverg}@speech.sri.com

ABSTRACT

We present extensions to the work of backoff hierarchical class n-gram language modeling of Zitouni et al. [1] by studying the efficacy of exploring the use of parts of speech (POS) information in hierarchical word clustering. We propose two approaches. One is to use POS n-gram contextual distributions of a target word for clustering. The other is to generate a class tree for each group of words sharing the same POS. The resulting class tree and a set of class trees, from the two approaches, respectively, are then employed in the hierarchical cluster language modeling. We evaluate the two approaches on SRI Arabic conversational telephone speech recognition system and show that the approach of building a set of POS-specific class trees achieves a 3% relative improvement on perplexity compared to the model of Zitouni et al. and a 8% relative improvement on perplexity over the baseline standard word n-grams. When used for N-best rescoring, our approach also outperforms the model of Zitouni et al. and the baseline and achieves significant word error rate (WER) reductions.

1. INTRODUCTION

Language models (LMs) must cope with the problem of estimating probabilities from a limited size of training data. The estimations of probabilities of low-frequency and unseen n-grams are inherently difficult, and the tasks suffer more serious data sparsity when the vocabulary size increases for large vocabulary automatic speech recognition systems. Data sparsity is particularly problematic for morphologically rich languages, for example, Turkish, Russian, or Arabic. Such languages have a high vocabulary growth rate, which results in high language model perplexity and a large number of out-of-vocabulary words. Word clustering addresses questions of data sparseness and generalization in statistical language modeling. It is expected that classes of words are useful in that statistics on classes can replace statistics on individual words whenever they are unavailable or unreliable. A traditional class-based language model is built by partitioning the vocabulary into classes and approximating transition probabilities from word to word to transition probabilities from word class to word class [2]. Clustering could be more useful when it can provide a variety of cluster granularities. Zitouni et al. [1] developed the approach of hierarchically clustering the vocabulary into a word tree in which the root node represents the whole vocabulary and a leaf node represents a word in the vocabulary. When estimating the conditional probability of a word based on its n-gram prefix, the hierarchical backoff strategy first backs off to its context with the most distant word replaced by its class, from the most specific to the most general (i.e., traversing bottom-up along the tree), and if none of these backoffs could guarantee a minimum number of occurrences then backs off to the normal lower-order (n-1)-gram prefix. In this way, it is likely to achieve a more accurate n-gram estimation, in particular on unseen words.

In this paper, we are interested in how to improve the clustering algorithm in the work of Zitouni et al., especially, how to effectively employ syntax information (POS in particular). We study two ways of employing POS information in word clustering. The results show that the approach of building a set of POS-specific class trees achieves a significant improvement on performance over the reported model of Zitouni et al. [1]. In the remainder of the paper we briefly review the model of Zitouni et al. and describe the details of our approaches. We test our models on SRI Arabic conversational telephone speech recognition system and compare to the baseline standard word n-gram LMs and our implementation of Zitouni et al.'s model, denoted *Z Model*.

2. HIERARCHICAL CLUSTER LM

An extensive presentation of Z Model can be found in [1]. Let C_i^j denote the j^{th} ancestor of the word w_i in the class hierarchy (the 1^{st} ancestor is its immediate parent). The probability of $p(w_i|w_{i-n+1}^{i-1})$ will first back off to $p(w_i|C_{i-n+1}^1, w_{i-n+2}^{i-1})$, and then recursively along the chain of ancestors, back off to $p(w_i|C_{i-n+1}^{j+1}, w_{i-n+2}^{i-1})$ when C_{i-n+1}^{j+1} is not the root of the tree and $p(w_i|w_{i-n+2}^{i-1})$ otherwise.

The principle of the MDI based word clustering algorithm is that the similarity between words can be measured based on their contextual statistics. Zitouni et al. [1] choose the *relative entropy* or *Kullback-Leibler (KL) distance* between distributions p_{w_1} and p_{w_2} of two words w_1 and w_2 to calculate how likely they are to be instances of the same cluster centroid. In Z Model, the contextual statistics of a word w

is estimated by the maximum likelihood estimated probabilities of its left and right neighboring words at distance d, given word w. Given V as the vocabulary, in the case of $d = 1$, the contextual statistics of a word w is defined as a $2V$ dimension vector as $p_1(w) = \{p_l\{w\}, p_r\{w\}\}$, where $p_l\{w\} = \{p_l(w_1/w), \ldots, p_l(w_V/w)\}$ and $p_r\{w\} = \{p_r(w_1/w), \ldots, p_r(w_V/w)\}$. Given the vocabulary V is partitioned into C clusters, $\{O_c\}, c = 1, C$, with their centroids denoted $o_c, c = 1, C$ respectively, the global discriminative information GDI can be calculated as $GDI = \sum_{c=1}^{C} \sum_{i \in O_c} D(w_i \parallel o_c)$. The goal of clustering is to minimize GDI. Note each cluster is represented by the centroid o_c of the cluster c. Given class $O_c = \{w_i, i = 1, \ldots, v_c\}$, the centroid of $O_c, o_c = \{o(k|o_c), k = 1, \ldots, 2V\}$ is defined as the mean of all context vectors $p_(w_i)$ of the words w_i belonging to O_c. Since a word is interpreted as a vector in terms of contextual distributions, the word clustering procedure becomes an encoding problem in VQ design. In order to cluster a sequence of sample data V into C classes, an LBG procedure [3] based on iterative improvement on GDI over the initial codebook generally produces decent clustering. The LBG procedure stops when GDI $\leq$ threshold. Note that there is a slight modification on the LBG procedure. We force that at least K words should appear in each cluster $O_c : N(O_c) > K$; otherwise, $(C \leftarrow C - 1)$ and the LBG procedure is repeated again. Once the C classes are determined for the current sample space, the above algorithm is recursively conducted on each of the classes to grow a hierarchical classification tree. The top-down clustering algorithm is described in detail in [1]. There are two parameters to control tree growing, namely, the maximum number of direct descendants for one class node in a tree, C, and the number of levels in the tree, L (Level 0 is the root of the tree containing the whole vocabulary and Level L contains leaves).

3. EMPLOYING POS INFORMATION

The motivation of our work is that one might expect that the clustering algorithm for a task could guarantee mutual substitutability between words in a class, syntactically and semantically. The word n-gram contextual statistics captures semantic similarity between words, but no information strong enough to discriminate distinct syntactic behaviors. We hypothesize that by incorporating some types of syntactic information into word clustering, we may manage to refine word clusters and hence improve the performance of the hierarchical cluster LM. We start with POS since it can be thought of as a classification based on syntactic behavior. Zitouni et al. [1] briefly talked about using the KL distortion measure "to define the similarity between two words w1 and w2 in terms of their POS function or their contextual information", but it is not described in their paper how they implemented it and whether they used POS information in their evaluations.

3.1. Model I. POS N-gram Context

In this approach, we replace each word in the context of a target word in Z Model with its POS. Given T as the space of linguistically defined POS tags for the task, in the case of $d = 1$, the contextual statistics of a word w is defined as $p_1(w) = \{p_l\{w\}, p_r\{w\}\}$, where $p_l\{w\} = \{p_l(t_1/w), \ldots, p_l(t_T/w)\}$ and $p_r\{w\} = \{p_r(t_1/w), \ldots, p_r(t_T/w)\}$. The discriminative information between two words w_1 and w_2 is computed as the KL distortion between $p_1(w_1)$ and $p_1(w_2)$, the same clustering procedure described in Section 2 is conducted, and the resulting class tree is then used in the hierarchical cluster LM described in Section 2.

3.2. Model II: POS-specific Class Tree

In this approach, for each linguistically defined POS tag $t \in \mathcal{T}, |\mathcal{T}| = T$ for the task, we partition the vocabulary into T subsets where all words in one subset share the same POS t and the corresponding subset is denoted $\mathcal{S}_t$. If there are words in the vocabulary with undefined POS, they are grouped together into a cluster labeled as "POS unknown". Hence, we generate T clusters (or $T+1$ in case we have the cluster "POS unknown"). For each of these clusters, a similar top-down clustering procedure using word n-gram as contextual statistics as described in Section 2 is applied, resulting in a set of POS-specific class trees. This approach is depicted in Figure 1, where Level 1 contains the set of POS tags (and probably the unknown tag). Since the subsets $\mathcal{S}_t, t \in \mathcal{T}$ bear different distributions from each other, we found that when growing a class tree for $\mathcal{S}_t$, if we change the threshold in the clustering algorithm in Section 2 from a constant threshold for all POS to $r \cdot \sum_{w \in \mathcal{S}_t} \text{Count}(w)$, where r is a constant for all POS tags, we can obtain better LM performance. Furthermore, there are two variants of implementing hierarchical backoff for this approach. If the POS membership is hard (i.e., each word belongs to one POS only) by using a predefined lexicon with hard POS membership or using an automatic POS tagger to assign the most likely tag to each word in the vocabulary, then the hierarchical backoff is the same as in Section 2. Otherwise, if the POS membership is soft, then the backoff estimation in Section 2 changes to a weighted average of conditional probabilities with the most distant word replaced by all of its ancestors in the same level L of the tree, where the weights correspond to the emission probability of the word belonging to its possible POS tags. We implemented both variants but in this paper, since our task provides a predefined lexicon with hard POS membership, we used the simpler hierarchical backoff strategy.

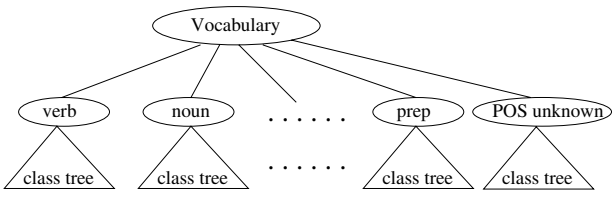

Fig. 1. Generating class trees for words sharing the same POS and words with unknown POS (in our Arabic LM experiments, these are all word fragments).

4. EXPERIMENTS AND RESULTS

4.1. Data and Baseline System

The results presented in this paper were produced on the LDC CallHome (CH) corpus of Egyptian Colloquial Arabic (ECA). The language model training set consists of the acoustic model training transcripts, hub5_new and eval96 subsets and contains 120 conversations (approximately 180K words) in total. LMs are evaluated on the 32K word dev96 test set and the 18K word eval97 test set. The recognizer was trained on the "romanized" transcriptions of the data. 9% of all word tokens are disfluencies and 1.6% are foreign words. The recognition vocabulary includes 18K words. The decoder uses a multipass approach. After a few steps, the single search flow splits into two sub-systems and we obtain two sets of N-best hypotheses, each of which could be rescored by a more powerful LM. The final hypotheses are generated by 2-way N-best ROVER. This is denoted *final-pass* N-best rescoring. By contrast, we can also apply the more powerful LM in every N-best rescoring step in order to generate better hypotheses for acoustic model adaptation and further improve WER. This is denoted *all-pass* N-best rescoring. Details of the recognition system can be found in [4]. On this task, a multi-stream factored language model (FLM) using morphological word representations has achieved significant perplexity and WER reductions [4] and is included in our comparison.

4.2. Model Optimization

We evaluated the Z Model, Model I, and II by computing their perplexities on the dev96 and eval97 test sets and by applying them for final-pass and all-pass rescoring. The final WERs are used for comparison. Before evaluations we tried to optimize the LMs. One issue about the MDI discriminative distance measure is that $D(p_{w_1} \parallel p_{w_2})$ is not defined when $p_{w_2} = 0$ but $p_{w_1} > 0$. However, we found that the problem is avoided by the clustering algorithm since it does not need to compute the KL distance between individual word distributions; instead, it only requires computing the KL distance between a word distribution and average distributions, the current cluster centroids, which are guaranteed to be nonzero whenever the word distributions are. In the implementation of Z model, We compared to the variant using Kneser-Ney smoothed [5] contextual statistics and unsurprisingly obtained slightly degraded performance. This is consistent with our intuition that smoothing will blur the discriminative cues from contextual statistics. We also optimized the two parameters C and L for word clustering on a subset held out from the training data. On this task, we found $C = 6$ and $L = 3$ is a local optimum on the heldout set. POS information required by Model I and Model II is extracted from the morphological class for each word from the CH ECA lexicon. The lexicon defines the stem, morph class, root, and pattern for each of the 54,545 word entries. The lexicon defines 1,360 morph classes in total, which can be viewed as "complex" POS tags (e.g., verb+subj-1st-sg+DO-3rd-masc-sg). Among these morph classes, 25 "simple" POS tags are defined (e.g., verb). For investigating the impact of using linguistically defined POS tags with different information granularities, we compared the performance of Model I and II using the 1,360 morph classes as the set of POS tags or just using the 25 POS tags. The perplexity results are shown in Table 1. As can be seen from the table, it is important to choose an appropriate information granularity for both models. For Model I, refined POS tags can compensate some lexical cues lost when replacing context words with their POS. By contrast, if the set of POS tags is large and quite refined in Model II, the data will be seriously segmented and the efficacy of hierarchical clustering and backoff will be reduced. We use the 1,360 morph class tags in Model I and 25 POS tags in Model II in the following experiments.

Table 1. Comparison of perplexity from trigram models of Model I and Model II on the Arabic dev96 and eval97 test sets, using a set of 1,360 morph classes and a set of 25 POS tags.

Tag Space	**dev96**		**eval97**	
	I	II	I	II
1,360 morph classes	217.2	214	215.1	210.5
25 POS tags	219.3	208.8	217.5	206.1

4.3. Comparison of LMs

For diagnostic comparisons, we constructed a traditional class LM [2] with words clustered using the Brown algorithm [6] and the number of classes is equivalent to the total number of classes in Level $L - 1$ of the tree in Z Model. This class LM is denoted *Brown-class LM* and on this task, it includes 196 classes. On the test sets, it produced much higher perplexity over the word ngrams, but could achieve a modest perplexity reduction when interpolated with the word ngrams. We observed a similar pattern on the two class LMs we built using the 25-POS tag set (denoted 25-class LM) and 1,360-morph-class tag set (denoted 1,360-class LM). Table 2 summarizes the perplexity results from the baseline word ngrams, the interpolated class LMs, Z Model, Model I and II, as well as the FLM results reported in [4]. Z Model achieves a 5% relative perplexity reduction over the baseline, while Model II outperforms it with a 3% relative reduction, even slightly lower than the FLM perplexities. By contrast, Model I is inferior to Z Model, which may be because word n-gram context could provide more lexical cues than POS context. Z Model, Model I and II all outperform the interpolated Brown-class LM, 25-class LM, and 1,360-class LM, demonstrating the effectiveness of hierarchical cluster language modeling. A final result that compares the WER after applying these LMs (all in trigrams) for final N-best rescoring and ROVER and some for all-pass rescoring appears in Table 3[1]. We can observe a similar pattern of performance ranking, with Model

[1]The final-pass rescoring WER from Model I are 57.4% and 56.7% on dev96 and eval97.

II significantly outperforms Z Model (˜ 1% and 0.7% absolute WER reduction, respectively) and Model I. The WER reduction from Model II over Z Model is significant at the 0.1 level using a difference of proportions significance test. WER from Model II is comparable to that from FLM. Moreover, we observed that combining the scores from FLM and Model II could achieve a further WER reduction. Overall, we obtained 1.5% and 1.2% absolute WER reduction over the baseline system by final-pass rescoring. As expected, we obtained a better WER reduction from all-pass rescoring over final-pass rescoring from Z Model and Model II[2].

One ongoing work of this research is that instead of using the backoff strategy in the hierarchical cluster LM in Section 2, word n-gram conditional probabilities are linearly interpolated with all of the conditional probabilities where the most distant word is replaced with its ancestor, from its immediate ancestor to the root. The interpolation weights are optimized using the EM algorithm on a heldout data set to maximize its likelihood, similar to *deleted interpolation*. We applied this technique to Z Model and Model II, denoted as *Z Model Interp* and *Model II Interp* in Table 3. As can be seen, they produced slightly better performance than their backoff counterparts, and the gain is held after combining with FLM, producing an overall 1.7% and 1.4% absolute WER reduction compared to the baseline on the two test sets, indicating this is a direction worth further exploring.

Table 2. Comparison of perplexity from bigram and trigram models of the standard word n-gram and a variety of LMs on the Arabic dev96 and eval97 test sets. The number 2 denotes bigram models and 3 trigram models.

LMs	dev96		eval97	
	2	3	2	3
Word Ngrams	230.3	227.1	227.9	223.7
Brown-class LM + word Ngrams	226.6	222.5	225.3	219.2
25-class LM + word Ngrams	229	224.8	226.1	221.5
1,360-class LM + word Ngrams	226.2	222	225.1	218.8
Z Model	225.8	215.7	224.7	212.6
Model I	227.4	217.2	227.3	215.1
Model II	219	208.8	217.9	206.1
FLM [4]	223	213	222	209

In conclusion, we studied two approaches for employing POS information for word clustering and hierarchical cluster LM. We observed that compared to inducing classes using only word-ngram contextual information, the approach of generating a set of POS-specific class trees produces significant improvement on language model performance, both on perplexity and N-best rescoring. A preliminary implementation of using interpolation instead of backoff in the hierarchical cluster LM also produces a modest improvement. In our future work, we will continue exploring effective algorithms to improve word clustering for class LMs by clustering words based on word n-gram context and syntactic constructions. For example, we can use headwords (and their POS tags) and syntactic dependency relation types to represent the syntactic behavior of words and then add this into the similarity measure for word clustering. We will investigate combinations of our approach with factored LMs. One approach is to use clusters with the variable granularity directly as factors in the factored language model framework, or to generate class hierarchies for factors used in a factored language model so that we can enhance the already flexible backoff framework of a factored language model. We will also port our approaches to other languages.

[2]The FLM all-pass WER included applying the FLM in the very first-pass recognition while our experiments on Z Model and Model II did not. In the future work, we will explore effective approaches to incorporate our approaches in the first-pass recognition as well.

Table 3. WERs (in %) and obtained by the baseline system and the system using the hierarchical cluster backoff/interpolation LMs for N-best list rescoring as described in Section 4.1. The * is explained in the footnote.

LMs	dev96		eval97	
	final-pass	all-pass	final-pass	all-pass
Word Ngrams	53.9		57.6	
Z Model	53.6	53.3	57.3	57
Z Model Interp	53.4	-	57.1	-
Model II	52.7	52.3	56.6	56.3
Model II Interp	52.5	52.2	56.4	56.2
FLM [4]	52.6	52.1*	56.6	56.1*
Model II + FLM	52.4	-	56.4	-
Model II Interp + FLM	52.2	-	56.2	-

Acknowledgments

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. NBCHD040058 and MDA972-02-C-0038. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Governorment. The authors also thank Andreas Stolcke and Kristin Precoda for useful discussions regarding its content.

5. REFERENCES

[1] I. Zitouni and H. J. Kuo, "Effectiveness of the backoff hierarchical class n-gram language models to model unseen events in speech recognition," in *Proceedings of ASRU*, 2003.

[2] F. Jelinek, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition*, Alex Waibel and Kai-Fu Lee, Eds. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[3] R. M. Gray, "Vector quantization," in *Readings in speech recognition*, Alex Waibel and Kai-Fu Lee, Eds. Morgan Kaufmann, 1990.

[4] D. Vergyri, K. Kirchhoff, K. Duh, and A. Stolcke, "Morphology-based language modeling for Arabic speech recognition," in *Proceedings of ICSLP*, 2004.

[5] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Rep., Harvard University, Computer Science Group, 1998.

[6] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.