A CLASS-BASED LANGUAGE MODEL FOR LARGE-VOCABULARY SPEECH RECOGNITION EXTRACTED FROM PART-OF-SPEECH STATISTICS

Christer Samuelsson and Wolfgang Reichl

Bell Laboratories, Lucent Technologies
600 Mountain Ave, Murray Hill, NJ 07974, USA

ABSTRACT

A novel approach is presented to class-based language modeling based on part-of-speech statistics. It uses a deterministic word-to-class mapping, which handles words with alternative part-of-speech assignments through the use of ambiguity classes. The predictive power of word-based language models and the generalization capability of class-based language models are combined using both linear interpolation and word-to-class backoff, and both methods are evaluated. Since each word belongs to one precisely ambiguity class, an exact word-to-class backoff model can easily be constructed. Empirical evaluations on large-vocabulary speech-recognition tasks show perplexity improvements and significant reductions in word error-rate.

1. INTRODUCTION

Data sparseness is a perennial problem when constructing language models for large-vocabulary speech recognition. Although word N-gram models have proved extremely useful when enough data is available to accurately estimate the N-gram probabilities, estimating these for low-frequency words is inherently difficult. And as vocabulary sizes increase, low-frequency words constitute an increasingly larger portion of them. One solution that immediately springs to mind is to group low-frequency words together into equivalence classes, and use the N-gram probabilities of the word classes, instead of those of the individual words. Language models using word classes are more compact and generalize better to unseen word sequences than purely word-based language models.

Simple models based on deterministic word-to-class relations are usually unable to capture the ambiguous nature of many words. On the other hand, overlapping word classes require difficult procedures to handle the one-to-many word-to-class mappings when annotating the training corpus and when using the language model for speech recognition [9, 10]. To avoid the latter problem, it is desirable that the word classes be true equivalence classes in that they partition the vocabulary, i.e., that each word belongs to exactly one word class.

There are many different ways to assign words to classes based on syntactic or semantic categories. One popular approach is based on minimizing the training set perplexity by automatically clustering the words [11, 10]. We here propose an automatic method for partitioning the vocabulary into word classes, based on syntactic behavior, that uses as its only information source a method for specifying for each word in the vocabulary the set of part-of-speech (PoS) tags that it can be attributed with, and the conditional probability of each tag given the word. The lexical component of a statistical PoS tagger constitutes such a resource.

Class-based language models have proved effective for training on small datasets and for fast language model adaptation. For large training datasets, word N-grams are still superior in capturing collocational relations between words. To utilize the power of word N-grams, the constructed class-based model can be combined with word N-gram models, and we evaluate two different methods for doing this on a large-vocabulary speech recognition task; both linear interpolation and a word-to-class backoff scheme retain the advantages of the two language-model types.

The integration of the class-based language model into the decoder is done directly in the forward search without generating a word lattice and rescoring it. Although the reduction in perplexity was modest in the experiments, the reduction in word error-rate was statistically significant. This shows the benefit of combining different information sources for the language model component of a speech recognition system.

2. PART-OF-SPEECH BASED LANGUAGE MODEL

In order to construct meaningful word classes, we wish to group words together that function similarly from a syntactic point of view. One way of doing this is to look at the part-of-speech (PoS) information associated with each word. This would typically include information about the word's main syntactic category, such as noun, verb, adjective, etc., and also additional information, like number (singular or plural), tense, degree of comparison, etc.

A first attempt might be to simply have one word class

for each part-of-speech tag. The problem with this approach is that many words can be assigned different PoS tags in different contexts, compare the two occurrences of the word "can" in *I can drink a can of coke*. This in turn means that we will either have overlapping word classes, which will complicate the application of the model in a speech recognition system, or we have to guess for each occurrence of a word what PoS tag to assign to it in the current context.

We will take another approach, which looks at ambiguity classes of words. This means that we will group those words together which can be assigned the same set of possible PoS tags. We will actually refine this a bit further to include the likelihoods of the possible PoS tags. An ambiguity class will be known by a (finite) sequence of PoS tags, the first one being the most likely one, the second one the second most likely one, etc. Since each word in the vocabulary will belong to exactly one ambiguity class, it will belong to exactly one word class. Very similar approaches to word clustering have been taken by [1] and [6].

This lexical PoS tag statistics will then be applied to a large training corpus, i.e., each word in it will be assigned an ambiguity class. The most frequent words in the training corpus will however be assigned singleton word classes, as there should be enough training data for this, and as we expect their syntactic behavior to be idiosyncratic. Ambiguity classes consisting of very many PoS tags will be truncated, either through a cumulative probability threshold, or by using an upper bound on the number of admissible tags. Likewise, ambiguity classes with very few member words will be avoided, again by truncating the tails of the tag sequences. Those ambiguity classes that remain after these pruning measures, together with the word classes assigned to high-frequency words, will constitute the final word classes.

Having thus annotated the training corpus with word classes, the word-class N-gram probabilities

$$P_C(w_k \mid w_i, w_k) \quad = \quad P(w_k \mid C_k) \cdot P(C_k \mid C_i, C_j)$$

can easily be calculated for the classes C_i, C_j and C_k of the words w_i, w_j and w_k. It is the probability of the current word, given the current word class, multiplied by the probability of the current word class given the two previous ones. The estimation of the conditional class probabilities $P(C_k \mid C_i, C_j)$ employs Katz's backoff scheme, in turned based on Good-Turing discounting, see [3].

Adding new words to the vocabulary is easily accomplished since only lexical information about a potential new word is required. The ambiguity class of a new word can easily be determined by the lexical component of an existing PoS tagger, as it will most likely have the capability to handle previously unseen words. This is an advantage over data-driven word-classes generated by word clustering, since adding words to the vocabulary typically requires reclustering the words, and it is impossible to achieve robustness for words unobserved in the training data.

3. COMBINING CLASS AND WORD MODELS

3.1. Linear Interpolation

While class-based language models generalize better to unseen word sequences, word-based language models in general have better performance, when enough training data is available. It is desirable to retain the advantages of each of these models by combining their word predictions. The most popular approach to combine different sources of information is by linear interpolation [2]. It consists of the weighted sum of the different prediction probabilities

$$\begin{aligned} P_I(w_k \mid w_i, w_j) \; &= \\ &= \; \lambda_W \cdot P_W(w_k \mid w_i, w_k) + \lambda_C \cdot P_C(w_k \mid w_i, w_j) \end{aligned}$$

where $P_W(w_k \mid w_i, w_k)$ is the word-based language model and $P_C(w_k \mid w_i, w_k)$ denotes the class-based model. Both models are based on Katz's backoff scheme and are internally normalized and consistent.

The estimation of the interpolation parameters $\lambda_{W,C}$ is based on perplexity minimization using the well-known EM-algorithm. Linear interpolation makes it possible to handle any number of different language models separately and to combine them afterwards. By minimizing the perplexity, it is guaranteed that the interpolated model is not worse than any of its components. The optimization of the interpolation parameters can be performed on the training data or online as an adaptive procedure on the test utterances [12].

3.2. Word-to-Class Backoff Model

Analyzing the behavior of the individual models leads us to a different way of combining the prediction probabilities. It is well known that non-backed-off word-trigram and word-bigram models perform better than class-based models. On the other hand, class-based models often produce better probability estimates than word-unigram backoffs, due to their ability to generalize (see Table 1). A word-to-category backoff model was proposed in [9] to retain the advantages of each of these approaches by backing off from the word-based to category-based probability estimates. Because of the stochastic mapping from words to categories in [9], an exact calculation of the backoff weights is not feasible and an approximate model is required. In our approach, the word classes are true equivalence classes, i.e., each word belongs to exactly one word class, and it is therefore feasible to construct an exact word-to-class backoff without any approximation, and to pre-calculate the backoff weights for each word and class history.

The proposed word-to-class backoff model utilizes word trigrams and word bigrams whenever possible and backs off

to the class-based model only when necessary, thus avoiding the non-informative word unigram probabilities:

$$P_{BO}(w_k \mid w_i, w_j) = \begin{cases} P'_W(w_k \mid w_i, w_j) & \textbf{if } |w_j, w_k| \neq 0 \\ \beta_{BO}(w_i, w_j) \cdot P_C(w_k \mid w_i, w_j) & \textbf{else,} \end{cases}$$

where $|w_j, w_k|$ denotes the number of word pairs (w_j, w_k) observed in the training data. $P'_W(w_k \mid w_i, w_j)$ is a truncated version of $P_W(w_k \mid w_i, w_j)$, with the backoff to word unigrams removed, and is not a proper probability distribution. The unigram probability mass is instead redistributed to the class-based model according to this equation, and the class-based model only kicks in after the word-bigram backoff has failed. The word-to-class backoff weights

$$\beta_{BO}(w_i, w_j) = \beta(w_i, w_j) \cdot \beta_C(w_i, w_j)$$

consist of the trigram-to-bigram backoff-weight $\beta(w_i, w_j)$ and the additional backoff weight

$$\beta_C(w_i, w_j) = \frac{1.0 - \sum_{\forall w_k \,:\, |w_j, w_k| \neq 0} P(w_k \mid w_j)}{1.0 - \sum_{\forall w_k \,:\, |w_j, w_k| \neq 0} P_C(w_k \mid w_i, w_j)}$$

which is necessary to meet the normalization requirement and can be pre-calculated for each context.

Contrary to linear interpolation, the word-to-class backoff model uses the class-based probabilities only if no trigram or bigram is available. It does not interpolate the class-based model with word-based trigrams and bigrams. This has the advantage of being selective and making optimal use of available information, whereas linear interpolation combines the probability estimates "blindly". The interpolation weights are optimized globally to reduce perplexity and cannot differentiate between different contexts [1].

4. EXPERIMENTAL RESULTS

4.1. Language Model Evaluation

The evaluation of the class-based language models was performed on the 20 000 word, open vocabulary Wall Street Journal (WSJ) task. The PoS statistics was extracted from the Penn Treebank annotation of 1 million words from the WSJ, [4], using the lexical component of the statistical PoS tagger described in [5]. This was subsequently applied to the training corpus of the language model, which consist of approximately 37 million words of text from the WSJ over the period 1987-89. In this phase, each word in the 20k vocabulary was assigned a sequence of PoS tags, ranked according to their conditional probabilities.

[1] The interpolation weights can only depend on the history: $\lambda = \lambda(w_i, w_j)$, but not on the current word w_k.

High-frequency words, i.e., words that occurred more than 100 000 times in the training corpus, were assigned individual word classes. To avoid ambiguity classes with many different PoS tags, the sequences of PoS tags was truncated after 90% probability mass, or after four tags, whichever occurred first. Also, any ambiguity class with less than five member words (types, rather than tokens) was repeatedly deprived of its least frequent PoS tag until it had at least five members, or exactly one PoS tag. The resulting ambiguity classes and the singleton word classes thus assigned to the words of the training corpus formed the final word-class annotation of the training corpus. In total 305 ambiguity classes were generated from 28 different PoS tags, plus the additional 50 singleton word classes.

From this, the word and class N-gram probabilities were calculated using Katz's backoff scheme. The perplexity for the 20k vocabulary 1992-93 development and evaluation test sentences (about 1400 utterances, 2% out-of-vocabulary words) are listed in Table 1.

	Perplexity				
	WTG	CTG	INTP	WCBO	#words
TOTAL	187	550	179	180	22 386
TRI	36	202	39	36	13 076
BI	838	1 496	770	838	7 758
UNI	109 970	17 338	48 825	65 020	1 552

Table 1: Perplexities for different language models.

The first row in Table 1 shows the total test set perplexity for the word-based trigram (WTG), the class-based trigram (CTG), the interpolated model (INTP) and the word-to-class backoff model (WCBO). In the next rows the perplexity of the test data is separately listed for different backoff cases of the word-based model. For 58.4% of the words a trigram (TRI) is used in the word-based model and the lowest perplexity is achieved. In 34.6% the word-based model backs off to bigrams (BI) and is still much better than the class-based model. For about 7% of the data further backing off to word unigram probabilities is required. In this case the class-based model generalizes better and provides a lower perplexity. Both, the interpolated and the word-to-class backoff model, reduce the perplexity about 5% compared to the baseline WTG model. Analyzing the detailed results from the interpolated model, we realize the global optimally interpolation weights ($\lambda_W \approx 0.85, \lambda_C \approx 0.15$) actually increase the perplexity for the TRI-data, while decrease the perplexity of the other data. This shows how linear interpolation of the highly specific word trigrams with the more general class trigrams can partially increase perplexity, if only on set of general interpolation parameters is available. In the word-to-class backoff model the perplexities for the TRI- and BI-data are identical to the WTG model. Avoiding the usage of unigram probabilities reduces perplexity for this part of the test utterances (UNI).

4.2. Speech Recognition Results

The interpolated model and the word-to-class backoff model were evaluated and compared to the baseline word trigram model on the 20k WSJ 1992 and 1993 evaluation test sets. All language models were directly integrated in our one-pass N-gram decoder [8]. No lattice rescoring is necessary in this system, which handles the different language models in the forward beam search on a layered self-adjusting decoding graph.

The acoustic models are three-state, cross-word triphone models with tied states, trained on the standard SI-84 and SI-284 training data. The state tying is based on a robust phonetic decision tree approach to cluster equivalent sets of context dependent states [7]. No acoustic adaptation was performed in the experiments. The phonetic lexicon for the 20 000 word vocabulary was automatically generated using a general English text-to-speech system with 41 phones. Table 2 presents the word error-rates for two acoustic models (SI-84 and SI-284) on the NOV92 and NOV93 evaluation data for the different language models.

		WER %		
Acoustic Model	Test Data	WTG	INTP	WCBO
SI-84	NOV92	12.3	11.6	11.6
SI-284	NOV92	9.8	9.5	9.4
SI-284	NOV93	13.5	13.3	13.1

Table 2: Word error-rates for different language models.

In all cases, a small but statistically significant error rate reduction up to 6% can be recognized. The combination of word-based and class-based language models helps to improve speech recognition performance by retaining the advantages of both models. The word-to-class backoff model performs slightly better than the linearly interpolated model. Backing off to the class-based model prevents the huge perplexities caused by the word unigram probabilities and avoids recognition errors for these words.

5. SUMMARY

A new approach to generating a class-based language model based on part-of-speech ambiguity classes was investigated. Two different methods for combining the class-based and word-based language models were evaluated and showed some perplexity improvement. Used in large-vocabulary speech-recognition tasks, both approaches, linear interpolation and word-to-class backoff, led to significant reductions in word error-rate.

6. REFERENCES

[1] W.Daelemans, J.Zavrel, P.Berck, S.Gillis. "MBT: A Memory-Based Part of Speech Tagger-Generator". In *Procs. Forth Workshop on Very Large Corpora*, Copenhagen, 1996.

[2] F.Jelinek, R.L.Mercer. "Interpolated Estimation of Markov Source Parameters from Sparse Data". In *Pattern Recognition in Practice*, pp. 381–397. North Holland, 1980

[3] S.M.Katz. "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer". In *IEEE Transactions on Acoustics, Speech, and Signal Processing 35(3)*, pp. 400–401, 1987.

[4] M.P.Marcus, B.Santorini, M.-A.Marcinkiewicz. "Building a Large Annotated Corpus of English: the Penn Treebank". *Computational Linguistics 19(2)*, pp. 313–330. ACL, 1993.

[5] C.Samuelsson, A.Voutilainen. "Comparing a Linguistic and a Stochastic Tagger". In *Procs. 35th Annual Meeting of the Association for Computational Linguistics*, pp. 246–253. ACL, 1997.

[6] E.Tzoukermann, D.R.Radev. "Using Word Class for Part-of-speech Disambiguation". In *Procs. Forth Workshop on Very Large Corpora*, Copenhagen, 1996.

[7] W.Reichl, W.Chou. "A Decision Tree State Tying Based on Segmental Clustering for Acoustic Modeling". In *Procs. ICASSP'98*, pp. 801-804, 1998.

[8] Q.Zhou, W.Chou. "An Approach of Continuous Speech Recognition Based on Self-Adjusting Decoding Graph". In *Procs. ICASSP'97*, pp. 1779-1782, 1997.

[9] T.R.Niesler, P.C.Woodland. "Combination Of Word-Based And Category-Based Language Models". In *Procs. ICSLP'96*, pp. 220-223, 1996.

[10] T.R.Niesler, E.W.D.Whittaker, P.C.Woodland. "Comparison Of Part-Of-Speech And Automatically Derived Category-Based Language Models For Speech Recognition". In *Procs. ICASSP'98*, pp. 177-180, 1998.

[11] H.Ney, U.Essen, R.Kneser. "On Structuring Probabilistic Dependencies in Stochastic Language Modeling". In *Computer Speech and Language 8*, pp. 1-38, 1994.

[12] R.Kneser, V.Steinbiss. "On the Dynamic Adaptation of Stochastic Language Models". In *Procs. ICASSP'93*, pp. 586-589, 1993.