VARIABLE N-GRAM LANGUAGE MODELING AND EXTENSIONS FOR CONVERSATIONAL SPEECH

Manhung Siu† *and Mari Ostendorf*

Boston University, 730 Commonwealth Ave, Boston, MA 02215
† Currently working for BBN Inc.

ABSTRACT

Recent progress in variable n-gram language modeling provides an efficient representation of n-gram models and makes training of higher order n-grams possible. In this paper, we apply the variable n-gram design algorithm to conversational speech, extending the algorithm to learn skips and classes in context to handle conversational speech characteristics such as repetitions and disfluency markers. We show that using the extended variable n-gram, we can build a language model that uses fewer parameters for longer context and improves the test perplexity and recognition accuracy.

1. INTRODUCTION

Language modeling is an integral part of a speech recognition system and the most successful technique to date is n-gram modeling. Recently, progress has been made using a variable n-gram model [1, 2], which characterizes word history by a variable number of preceding words, as a more efficient way of using the data than fixed-length n-grams. However, the variable n-gram techniques have only been explored on written text, such as the North American Business News corpus. In conversational speech, there are many occurrences of disfluency markers and repetitions, as in: "That that was uh you know a fairly large building." Previous studies on conversational speech and disfluencies have shown that skipping all disfluency markers actually hurts performance, while skipping some may help [3, 4]. This motivates us to extend the variable n-gram algorithm to selectively skip words in the word history. We also extend the variable n-gram to handle word classes in context. Instead of grouping words together in all contexts, as in a class grammar, our class-in-context model groups words together only in a particular context. For example, "sort" and "kind" are typically not associated together, but in the context of "of", ("sort of" and "kind of") they are very similar. Our algorithm creates a variable n-gram that both requires a smaller number of parameters and performs better than a regular n-gram model.

In this paper, we first describe the basic variable n-gram training algorithm. Second, we describe our extension to handle skips. Third, we describe our algorithm to search for word classes in context. Lastly, we report some experimental results on the Switchboard corpus.

2. BASIC VARIABLE N-GRAM TRAINING

The goal of the variable n-gram design algorithm is to increase test set likelihood and reduce the number of parameters in the model. Taking an approach similar to [1], we begin with an n-gram of a large n, say, $n = 5$ and arrange the model as a tree such that each node represents a conditional distribution of a particular context length as shown in Figure 1. Denote an n-gram conditional distribution as $p(.|h_k)$ where k is the length of the history. For a trigram distribution, it is denoted as $p(.|h_2)$. For each node in this tree, we can compute the change in goodness or the distance between the child node and its parent node, which corresponds to using a shorter history. We can view the variable n-gram as determining the optimal conditioning event by selectively removing a child node, which reduces the complexity of the model and the number of parameters in the model.

One measure of the goodness of the model is maximizing training likelihood. Since replacing a node always reduces training likelihood and does not necessarily indicate how much this can generalize in test, we use the change in leave-one-out training log likelihood [5]. The leave-one-out probability of a word w conditioned on context h when using the Witten and Bell back-off [6] can be express as

$$p_{loo}(w|h) = \frac{c(w,h) - 1 + r_h p(w)}{c(h) - 1 + r_h},$$

where $c(w,h)$ is the occurrence count of w following h, $c(h)$ is the number of occurrences of h, and r_h is the number of unique words following h. Alternatively, we can use the minimum description length (MDL) criterion which is an adjusted log likelihood for comparison between models with different number of parameters. The MDL distance of replacing distribution j by distribution i can be expressed as

$$d(i,j) = l_i - l_j - \frac{k_i - k_j}{2} log(n_j),$$

where l_x is the log likelihood evaluated using distribution x, k_x is the number of parameters in model x, and n_j is the number of tokens observed in distribution j. We rank order all nodes based on the goodness metric and select those based on a pre-set threshold. Our experimental results show that using the leave-one-out log likelihood as the criterion performs better that using MDL, giving a Switchboard test-set perplexity of 77.5 verse 81.0.

3. ALGORITHM EXTENSIONS

Variable N-gram with Skips. For the word sequence "that was uh you know", the distribution of $p(.|\text{was uh you})$ and $p(.|\text{was you})$ can be quite similar. In the tree representation, instead of comparing a child node with its parent, we also compare it to its "uncles" and "aunts", which are skipped versions of its original context, as indicated in Figure 1 with arrows. In the example above, we not only compare $p(.|\text{was uh you})$ to $p(.|\text{uh you})$ (which is its parent), we also compare it to $p(.|\text{was you})$ by skipping the word *uh*. The best replacement of the child node is selected among its parent, uncles and aunts based on the maximal change in leave-one-out log likelihood.

We have several options as to how to tie the two nodes by modifying their distributions and the subtree underneath the nodes. Denote c the child node and u the uncle node, t_x and $\hat{t}_x$ the original and final subtree underneath a node x, p_x and $\hat{p}_x$ the original and final distribution associated with x. We considered two options for modifying the node distributions: 1) copying the uncle's distribution to the child, $\hat{p}_c = \hat{p}_u = p_u$, and 2) pooling the data from both nodes to estimate a shared distribution, $\hat{p}_c = \hat{p}_u = p_c + p_u$. Similarly, there are different options for handling the subtrees: 1) copy the subtree of the uncle, $\hat{t}_c = \hat{t}_u = t_u$, and 2) combine the two subtrees, $\hat{t}_c = \hat{t}_u = t_c + t_u$. Which option to use depends on the relationship between the child and uncle node. If they are truly representative of not only the same distribution but same context, then merging the distribution and the subtree should be preferable. However, if the two nodes are simply similar distributions of different contexts, then combining the subtrees may hurt performance. Since the goodness metrics evaluate nodes and not subtrees, this question must be answered empiricially. Our experiments show that merging the data associated with nodes and copying the subtree of the uncle node performs the best giving a switchboard test-set perplexity of 77.4.

Variable N-gram with Classes. Instead of comparing a child node with its parent or uncles, we can compare a child node with its siblings, that is, other nodes that share the same parent. If two nodes are found to be similar, we can combine their data to form a new node. Since all nodes that share the same parent also have in common $k-1$ of the k-word context, grouping these nodes together creates word-classes within a particular context. This differs from traditional class grammars in two respects. First, class definition is context dependent. For example, "sort" and "kind" are grouped together only when they are followed by "of"; they are not grouped together when they are followed by other words. Second, classes are defined on word histories but not the observations. For example, $p(.|\text{we})$ and $p(.|\text{you})$ may be grouped together and estimated jointly, but $p(\text{we}|h)$ and $p(\text{you}|h)$ are always estimated separately. Similar to the the skip variable n-gram, we have different options of combining the nodes. In our class-in-context work, the best combination is merging the distributions as well as the subtrees.

Figure 2 shows node tying in the class-in-context variable n-gram. Suppose the word "is" and "was" followed by "uh you" are similar, node 4 and node 9 are combined to form one single node. Their subtrees are also combined; a single subtree is formed underneath instead of two separate ones under each node.

Denote r be a context node and $\{c_i\}$ be the set of child nodes of r. Ideally, the clustering of contexts into classes should be done after distances between all child nodes, $d(c_i, c_j)$, are computed. However, given the number of nodes in the tree, that is computationally expensive. Instead, we approximate the algorithm by iteratively computing the distance of all nodes to one node and grouping that node together with the closest other if the distance is below a threshold. Again, we use the change in leave-one-out log likelihood as the distance measure.

4. EXPERIMENTS

Experiment paradigm. We test our algorithm on the Switchboard corpus, where the maximum n in the variable n-gram is 5. Perplexity results are based on training with the linguistically segmented data using 1.4M words and testing on 10k words. The recognition experiments train models using 3M words of acoustically segmented Switchboard and callhome English data with a dictionary of 26K words. Recognition tests are performed on 7 Switchboard conversations that are part of the 1995 Darpa Hub-5e evaluation set. Recognition is performed by rescoring the recognition n-best of 100, which are generated using a trigram grammar.

Qualitative results. It is informative to look at the words that are affected by the basic variable n-gram, and the skip and class extensions. In tables 1-3, we tabulate the most significant contexts that are observed in our training as measured by gain in leave-one-out log likelihood. Because of limited space, we show only the top 20 contexts, plus one or two additional interesting contexts.

Table 1 tabulates some of the most significant contexts that the basic variable n-gram selected. Column one shows the rank of the n-gram context change, column two shows the original contexts before the variable n-gram, and column three shows the contexts after the variable n-gram. From the table, we notice that word repetitions, sentence begin markers and the first word of a long phrase are often reduced by the variable n-gram. Removing repetition words in conversational speech is consistent with the findings in [4]. Sentence begin symbols (marked by the symbol _B_) are removed from the history for some of the conjunctions and conversational markers, such as "well", and "but" because these markers occur primarily in the sentence begin position. Initial words in a multi-word phrase (such as "a" in "a lot of" or the first "as" in "as far as") are removed, probably because these

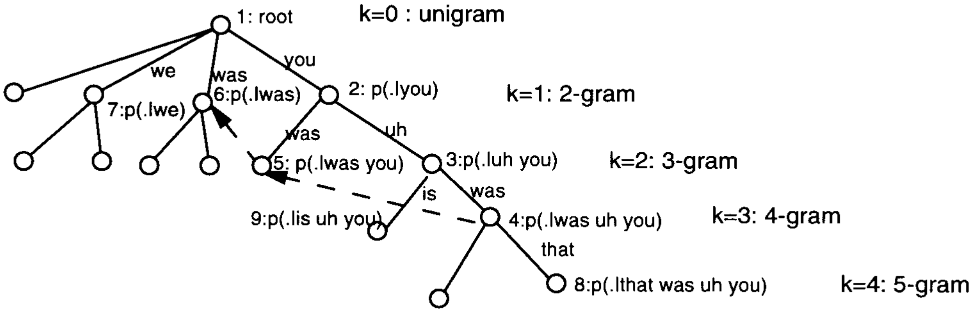

Figure 1. Example of a tree representation of a variable n-gram, with arrows indicating node replacement for a skip in the history.

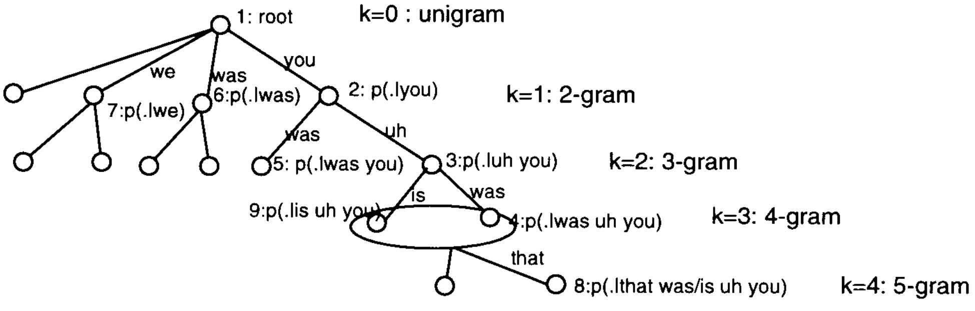

Figure 2. Example of a tree representation of a variable n-gram, with arrows indicating node replacement for class-in-context in the history.

words typically occur as a phrase and seeing the last words of the phrase is enough to identify the phrase.

Table 2 shows the most significant contexts where words are skipped. Column one shows the rank of the skips, column two shows the original context before the skip, and column three shows the context after the skip. Disfluency markers (such as "um", "uh", "you know") and modifiers and conjunctions (such as "actually", "really", and "but") are most frequently skipped, which seems reasonable since these words have little information content.

Table 3 gives some of the most significant class contexts. Column one shows the rank of the combination, column two and three show the contexts that are combined. Sentence begin and conjunctions, singular and plural forms of words, and pronouns (such as "they" and "we") in certain contexts are frequently grouped together, as well as phrases (like "so many" and "too many").

Quantitative results. The perplexity results are summarized in Table 4, where we compare standard trigram language models with variable n-grams (up to 5-gram), with and without skips and classes in context. Results are given for the test perplexity and the number of conditional distributions in the models. From our experiments, the use of skip and class-in-context together gives the best perplexity with fewer number of conditional distributions to estimate. Table 5, which tabulates the recognition performance of using the basic variable n-gram and the skip and class-in-context extensions, shows that this model also gives the lowest word error rate (WER).

Rank	Child Node	Parent Node
1	A LOT OF	LOT OF
2	_B_ BUT	BUT
3	TO BE	BE
4	I THINK	THINK
5	I GUESS	GUESS
6	AND AND	AND
7	THE THE	THE
8	_B_ WELL	WELL
9	UH UH	UH
10	UH YOU_KNOW	YOU_KNOW
11	_B_ OH	OH
12	_B_ IT'S	IT'S
13	I DON'T KNOW	DON'T KNOW
14	_B_ WELL I	WELL I
15	A A	A
16	UH THE	THE
17	_B_ BUT UH	BUT UH
18	A LOT	LOT
19	_B_ AND_THEN	AND_THEN
27	AS FAR AS	FAR AS

Table 1. Most important merges in variable n-gram

5. SUMMARY

In this paper, we show that using variable n-gram on conversational speech captures some of the character-

Rank	Child context	Uncle context
1	DON'T REALLY	DON'T
2	HAVEN'T REALLY	HAVEN'T
3	WELL UH	WELL
4	CAN UH	CAN
5	I'VE ALSO	I'VE
6	WE ALL	WE
7	COULD PROBABLY	COULD
8	SAY YOU_KNOW	SAY
9	MUCH UH	MUCH
10	BUT [BREATHING]	BUT
11	SHOULD UH	SHOULD
12	SAID UH	SAID
13	WOULD REALLY	WOULD
14	MY UH	MY
15	WEREN'T REALLY	WEREN'T
16	CAN NOT	CAN
17	COULD REALLY	COULD
18	WATCH UH	WATCH
19	CAN REALLY	CAN
20	THEY PROBABLY	THEY

Table 2. List of contexts that benefited from a skip

istics of spontaneous speech such as repetitions. By extending the algorithm to include skips and class-in-context, we create a more powerful model that can also skip interruptions or pause fillers and, at the same time, combine words that function similarly in context.

By combining variable n-gram with skips and classes-in-context, we improve test perplexity by 5% while reducing the number of conditional distributions needed by 60% over a standard trigram. The smaller, more compact model gives a reduction of 0.5% in word error rate when applied to rescoring recognition n-best hypotheses.

REFERENCES

[1] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia," in J. Cowan et al., editor, *Advances in Neural Information Processing Systems,* vol. 6, pp. 176-183, Morgan Kauffmann, 1994.

[2] T. Niesler and P. Woodland, "A variable-length category-based n-gram language model," *Proc. IEEE ICASSP,* pp. 164-167, 1996.

[3] M. Siu and M. Ostendorf, "Modeling Disfluencies in Conversational Speech," *Proc. ICSLP,* pp. 386-389, 1996.

[4] A. Stolcke and E. Shriberg, "Statistical Language Modeling for Speech Disfluency," *Proc. IEEE ICASSP,* pp. 405-408, 1996.

[5] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modeling," *Computer Speech and Language,* vol. 8, pp. 1-38, 1994.

[6] I. H. Witten and T. C. Bell. "The zero-frequency problem: Estimation the probabilities of novel events in adaptive text compression." *IEEE Trans. Inform. Theory,* IT-37:1085-1094, Jul 1991.

Rank	Sequence 1	Sequence 2
1	AND THE	THE THE
2	KIND OF	SORT OF
3	AND THE	UH THE
4	_B_ THEY	THEY THEY
5	OF THESE	ALL THESE
6	_B_ THEY'RE	AND THEY'RE
7	_B_ THEY	AND THEY
8	THEY ARE	THAT ARE
9	AND THE	YOU_KNOW THE
10	AND THE	THAT THE
11	AND UH	UH UH
12	A BIG	THE BIG
13	_B_ YOU_KNOW	AND YOU_KNOW
14	_B_ WE	WE WE
15	I HAD	THEY HAD
16	HAVE A	A A
17	_B_ IT'S	IT'S IT'S
18	_B_ IT'S	AND IT'S
19	SO MANY	TOO MANY
20	HAVE BEEN	HAS BEEN

Table 3. List of most important contextual merges

Experiment	Perpl.	Distribs
Baseline 3-gram	81.5	88921
Class in context for 3-gram	81.1	70675
Var. n-gram up to 5-gram	77.5	30112
Var. n-gram w/ skip	77.4	29381
Var. n-gram w/ skip and class	77.2	27527

Table 4. Perplexity of different language models and the corresponding number of conditional distributions.

Experiment	WER
Baseline 3-gram	35.58
Variable n-gram up to 5-gram	35.30
Variable n-gram w/ skip	35.29
Variable n-gram w/ skip and class	35.10

Table 5. Recognition results using the extended variable n-gram.