Less is More: Significance-Based N-gram Selection for Smaller, Better Language Models

Robert C. Moore **Chris Quirk**
Microsoft Research
Redmond, WA 98052, USA
{bobmoore,chrisq}@microsoft.com

Abstract

The recent availability of large corpora for training N-gram language models has shown the utility of models of higher order than just trigrams. In this paper, we investigate methods to control the increase in model size resulting from applying standard methods at higher orders. We introduce significance-based N-gram selection, which not only reduces model size, but also improves perplexity for several smoothing methods, including Katz backoff and absolute discounting. We also show that, when combined with a new smoothing method and a novel variant of weighted-difference pruning, our selection method performs better in the trade-off between model size and perplexity than the best pruning method we found for modified Kneser-Ney smoothing.

1 Introduction

Statistical language models are potentially useful for any language technology task that produces natural-language text as a final (or intermediate) output. In particular, they are extensively used in speech recognition and machine translation. Despite the criticism that they ignore the structure of natural language, simple N-gram models, which estimate the probability of each word in a text string based on the $N-1$ preceding words, remain the most widely-used type of model.

Until the late 1990s, N-gram language models of order higher than trigrams were seldom used. This was due, at least in part, to the fact the amounts of training data available did not produce significantly better results from higher-order models. Since that time, however, increasingly large amounts of language model training data have become available ranging from approximately one billion words (the Gigaword corpora from the Linguistic Data Consortium) to trillions of words (Brants et al., 2007). With these larger resources, the use of language models based on 5-grams to 7-grams is becoming increasingly common.

The problem we address here is that, even when relatively modest amounts of training data are used, high-order N-gram language models estimated by standard techniques can be impractically large. Hence, we investigate ways of building high-order N-gram language models without dramatically increasing model size. This is, of course, the same goal behind much previous work on language model pruning, including that of Seymore and Rosenfeld (1996), Stolcke (1998), and Goodman and Gao (2000). We take a novel approach, however, which we refer to as significance-based N-gram selection. We reject a higher-order estimate of the probability of a particular word in a particular context whenever the distribution of observations for the higher-order estimate provides no evidence that the higher-order estimate is better than our backoff estimate.

Perhaps our most surprising result is that significance-based N-gram selection not only reduces language model size, but it also improves perplexity when applied to a number of widely-used smoothing methods, including Katz backoff and several variants of absolute discounting.[1] In contrast, experiments applying previous pruning methods to Katz backoff (Seymore and Rosenfeld, 1996; Stolcke, 1998) and absolute discounting (Goodman and Gao, 2000) always found the lowest perplexity model to be the unpruned model.

We tested significance-based selection on only one smoothing method without obtaining improved perplexity: modified Kneser-Ney (KN)

[1] For most of the standard smoothing methods mentioned here, we refer the reader to the excellent comparative study of smoothing methods by Chen and Goodman (1998). References to the original sources may be found there.

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 746–755,
Singapore, 6-7 August 2009.

smoothing (Chen and Goodman, 1998). This is unfortunate, because modified KN smoothing generally seems to have the lowest perplexity of any known smoothing method for N-gram language models; in our tests it had a lower perplexity than any of the other models, with or without significance-based N-gram selection. However, when we compared modified KN smoothing to our best results applying N-gram selection to other smoothing methods for multiple N-gram orders, two of our models outperformed modified KN in terms of perplexity for a given model size.

Of course, the trade-off between perplexity and model size for modified KN can also be improved by pruning. So, in a final set of experiments we found the best combinations we could for pruned modified KN models, and we did the same for our best model using significance-based selection. The best pruning method for the latter turned out to be a novel modification of weighted-difference pruning (Seymore and Rosenfeld, 1996) that was especially convenient to compute given our method for performing significance-based N-gram selection. The final result is that our best model using significance-based selection and modified weighted difference pruning always had a better size/perplexity trade-off than pruned modified KN, with up to about 8% perplexity reduction for a given model size.

2 Significance-Based N-gram Selection

The idea of using a statistical test to decide whether to use a higher- or lower-order estimate of an N-gram probablity is not new. It was perhaps first proposed by Ron, et al. (1996), who suggested using a threshold on relative entropy (Kullback-Liebler divergence) as an appropriate test to decide whether to extend the context used to predict the next token in a sequence. Stolcke (1998) used the same metric in his work on language model pruning, and he also pointed out that weighted difference pruning is, in fact, an approximation of relative entropy pruning. However, while relative entropy pruning is based on a statistical test, it is not a *significance* test. The difference in probability represented by a certain relative entropy value can be statistically significant when measured on a large corpus, but not significant when measured on a small corpus.

The primary test we use to choose between higher- or lower-order estimates of an N-gram probablity is inspired by an insight of Jedynak and Khudanpur (2005). They note that, given a set of y observations of a multinomial distribution, the observed counts will have the highest probabilty of any possible set of y observations for the maximum likelihood estimate (MLE) model derived from the relative frequencies of those observations. In general, however, the MLE model will not be the only model for which this set of observations is the most probable set of y observations. Jedynak and Khudanpur call the set of such models the maximum likelihood set (MLS) for the observations.

Jedynak and Khudanpur argue that the observations alone do not support choosing the MLE over other members of the MLS. The MLE may assign the observations a higher probability than other members of the MLS, but that may be an accident of what outcomes are possible given the number of observations. If we flip a coin 9 times and get 5 heads, is there any reason to believe that the probability of heads is closer to the MLE $5/9$ than it is to $5/10$? No, because $5/9$ is as close as we can come to $5/10$, given 9 observations.

We apply this insight to the problem of N-gram selection as follows: For each word w_n in a context $w_1...w_{n-1}$ with a backoff estimate for the probability of that word in that context $\beta\, p(w_n|w_2...w_{n-1})$,[2] we do not include an explicit estimate of $p(w_n|w_1...w_{n-1})$ in our model, if the backoff estimate is within the MLS of the counts for $w_1...w_n$ and $w_1...w_{n-1}$.

This requires finding the MLS of a set of observations only for binomial distributions (rather than the general multinomial distributions studied by Jedynak and Khudanpur), which has a very simple solution:

$$MLS(x,y) = \left\{ p \;\middle|\; \frac{x}{y+1} \leq p \leq \frac{x+1}{y+1} \right\}$$

where x is the count for $w_1...w_n$, y is the count for $w_1...w_{n-1}$, and p is a possible backoff probabilty estimate for $p(w_n|w_1...w_{n-1})$. In this case, the MLS is the set of binomial distributions that have x successes as their mode given y trials, which is well-known to be specified by this formula.

We describe this method as "significance-based" because we can consider our criterion as a significance test in which we take the backoff

[2] $p(w_n|w_2...w_{n-1})$ being the next lower-order estimate, and β being the backoff weight for the context $w_1...w_{n-1}$.

probability estimate as the null hypothesis for the estimate in the higher-order model, and we set the rejection threshold to the lowest possible value; we reject the null hypothesis (the backoff probability) if there are *any* outcomes for the given number of trials that are more likely, according to the null hypothesis, than the one we observed.

We make a few refinements to this basic idea. First, we never add an explicit higher-order estimate to our model, if the next lower-order estimate is not explicitly stored in the model. This enables us to keep only the next lower-order model available while performing N-gram selection.

Next, we observe that in some cases the higher-order estimate for $p(w_n|w_1...w_{n-1})$ may not fall within the MLS for the observed counts, due to smoothing. In this case, we prefer the backoff probability estimate if it lies within the MLS or between the smoothed higher-order estimate and the MLS. Otherwise, we would reject the backoff estimate for being outside the MLS, only to replace it with a higher-order estimate even farther outside the MLS.

Finally, we note that the backoff probability estimate for an N-gram not observed in the training data sometimes falls outside the corresponding MLS, which in the 0-count case simplifies to

$$MLS(0, y) = \left\{ p \;\middle|\; 0 \leq p \leq \frac{1}{y+1} \right\}$$

When this happens, we include an explicit higher-order estimate $p(w_n|w_1...w_{n-1}) = 1/(y+1)$, which is the upper limit of the MLS. This is similar to Rosenfeld and Huang's (1993) "confidence interval capping" method for reducing unreasonably high backoff estimates for unobserved N-grams.

In order to apply this treatment of 0-count N-grams, we sort the explicitly-stored N-grams for each backoff context by decreasing probability. For each higher-order context, to find the 0-count N-grams subject to the $1/(y+1)$ limit, we traverse the sorted list of explicitly-stored N-grams for its backoff context. When we encounter an N-gram whose extension to the higher-order context was not observed in the training data, we give it an explicit probability of $1/(y+1)$, if its weighted backoff probability is greater than that. We stop the traversal as soon as we encounter an N-gram for the backoff context that has a weighted backoff probability less than or equal to $1/(y+1)$, which in practice means we actually examine only a small number of backoff probabilities for each context.

3 Finding Backoff Weights by Iterative Search

The approach described above is very attractive from a theoretical perspective, but it has one practical complication. To decide which N-grams for each context to explicitly include in the higher-order model, we need to know the backoff weight for the context, but we cannot compute the backoff weight until we know exactly which higher-order N-grams are included in the model.

We address this problem by iteratively solving for a backoff weight that yields a normalized probability distribution. For each context, we guess an initial value for the backoff weight and keep track of the sum of the probabilites resulting from applying our N-gram selection method with that backoff weight. If the sum is greater than 1.0, by more than a convergence threshold, we reduce the estimated backoff weight and iterate. If the sum is less than 1.0, by more than the threshold, we increase the estimated weight and iterate.

It is easy to see that, for all standard smoothing methods, the function from backoff weights to probability sums is piece-wise linear. Within a region where no decision changes about which N-grams to include in the model, the probability sum is a linear function of the backoff weight. At values of the backoff weight where the set of selected N-grams changes, the function can be discontinous. With a little more effort, one can see that the linear segments overlap with respect to the probability sum in such a way that there will always be one or more values of the backoff weight that make the probability sum equal 1.0, with one specific exception.

The exception arises because of the capping of backoff probabilites for unobserved N-grams. It is possible for there to be a context for which all observed N-grams are included in the higher-order model, the probabilities for all unobserved N-grams are either capped at $1/(y+1)$ or effectively 0 due to arithmetic underflow, and the probability sum is less than 1.0. For some smoothing methods, the probability sum cannot be increased in this situation by increasing the backoff weight. We check for this situation, and if it arises, we increase the cap on the 0-count probability just enough to make the probability sum equal 1.0.

That exception aside, we iteratively find backoff weights as follows: For an initial estimate of the backoff weight for a context, we compute

what the backoff weight would be for the base smoothing method without N-gram selection. If that value is less than 1.0, we use it as our initial estimate, otherwise we use 1.0, which annecdotally seems to produce better models than initial estimates greater than 1.0, in situations where there are multiple solutions. If the first iteration of N-gram selection produces a probability sum less than 1.0, we repeatedly double the estimated backoff weight until we obtain a sum greater than or equal to 1.0, or we encounter the special situation previously described. If the initial probability sum is greater than 1.0, we repeatedly halve the estimated backoff weight until we obtain a sum less than or equal to 1.0.

Once we have values for the backoff weight that produce probability sums on both sides of 1.0, we have a solution bracketed, and we can use standard numerical search techniques to find that solution. At every subsequent iteration, we try a value for the backoff weight between the largest value we have tried that produces a sum less than 1.0 and the smallest value we have tried that produces a sum greater than 1.0. We stop when the difference between these values of the backoff weight is less than a convergence threshold.

We use a combination of simple techniques to choose the next value of the backoff weight to try. The primary technique we use is called the "false position method", which basically solves the linear equation defined by the two current bracketing values and corresponding probability sums. The advantage of this method is that, if our bracketing points lie on the same linear segment of our function, we obtain a solution in one step. The disadvantage of the method is that it sometimes approaches the solution by a long sequence of tiny steps from the same side.

We try to detect the latter situation by keeping track of the number of consecutive iterations that make a step in the same direction. If this number reaches 10, we take the next step by the bisection method, which simply tries the value of the backoff weight halfway between our two current bracketing values. In practice, this combined search method works very well, taking an average of less than four iterations per backoff weight.

4 Modified Weighted-Difference Pruning

While the N-gram selection method described above considerably reduces the number of parameters in a high-order language model, we may wish to reduce language model size even more. The concept of significance-based N-gram selection to produce smaller models could be extended by relaxing our criterion for using backoff distributions in place of explicit higher-order probability estimates, but true significance tests at more relaxed thresholds that are accurate for small counts are expensive to compute; so we resort to more conventional language model pruning methods.

In our experiments, we tried four methods for additional pruning: simple count cutoffs, relative entropy pruning (REP) (Stolcke, 1998), and two modified versions of Seymore and Rosenfeld's (1996) weighted-difference pruning (WDP). In the notation we have been using, Seymore and Rosenfeld's WDP criterion for using a backoff estimate, in place of an explicit higher-order estimate, is that the quantity

$$K \times \left(\begin{array}{l} \log(p(w_n|w_1...w_{n-1})) - \\ \log(\beta_u \, p(w_n|w_2...w_{n-1})) \end{array} \right)$$

be less than a pruning threshold, where K is the Good-Turing-discounted training set count for $w_1...w_n$, and β_u is the backoff weight for the unpruned model.

The first of our modified version of WDP uses the following quantity instead:

$$p(w_1...w_n) \times \left| \begin{array}{l} \log(p(w_n|w_1...w_{n-1})) - \\ \log(\beta_p \, p(w_n|w_2...w_{n-1})) \end{array} \right|$$

where $p(w_1...w_n)$ is an estimate of the probability of $w_1...w_n$ and β_p is the backoff weight for the pruned model.

We make three modifications to WDP in this formula. First, we follow a suggestion of Stolcke (1998) by replacing the discounted training set count K of $w_1...w_n$ with an estimate the joint probability of $w_1...w_n$, computed by chaining the explicit probability estimates, according to our model, for all N-gram lengths up to n.

The second modification to WDP is that we use the absolute value of the difference of the log probabilities. By using the signed difference of the log probabilities, Seymore and Rosenfeld will always prune a higher-order probability estimate if it is less than the backoff estimate. But the backoff estimate may well be too high. Using the absolute value of the difference avoids this problem.

$$p(w_n|w_1 \dots w_{n-1}) = \begin{cases} \alpha_{w_1 \dots w_{n-1}} \frac{C(w_1 \dots w_n) - D_{n,C(w_1 \dots w_n)}}{C(w_1 \dots w_{n-1})} & \\ \quad + \beta_{w_1 \dots w_{n-1}} p(w_n|w_2 \dots w_{n-1}) & \text{if } C(w_1 \dots w_n) > 0 \\ \\ \gamma_{w_1 \dots w_{n-1}} p(w_n|w_2 \dots w_{n-1}) & \text{if } C(w_1 \dots w_n) = 0 \end{cases}$$

$$\beta_{w_1 \dots w_{n-1}} = \delta \frac{|\{w' | C(w_1 \dots w_{n-1} w') > 0\}|}{C(w_1 \dots w_{n-1})}$$

$$\alpha_{w_1 \dots w_{n-1}} = 1 - \beta_{w_1 \dots w_{n-1}}$$

Figure 1: New language model smoothing method

The final modification is that we compute the difference in log probability with respect to the backoff weight for the pruned model rather than the unpruned model, which we are able to do by performing the pruning inside our iterative search for the value of the backoff weight. We do this because, if the backoff weight is changed significantly by pruning, backoff estimates that meet the pruning criterion with the old backoff weight may no longer meet the criterion with the new backoff weight, and vice versa. Since the new backoff weight is the one that will be used in the pruned model, that seems to be the one that should be used to make pruning decisions.

Our second variant of modified WDP is like the first, but it estimates $p(w_1 ... w_n)$ simply by dividing Seymore and Rosenfeld's discounted N-gram count K by the total number of highest-order N-grams in the training corpus. This is equivalent to smoothing only the highest-order conditional N-gram model in estimating $p(w_1 ... w_n)$, estimating all the lower-order probabilities in the chain by the corresponding MLE model. We refer to this joint probability estimate as "partially-smoothed", and the one suggested by Stolcke as "fully-smoothed".

5 Evaluation

We carried out three sets of evaluations to test the new techniques described above. First we compared the perplexity of full models and models reduced by significance-based N-gram selection for seven language model smoothing methods. For the best three results in that comparison, we looked at the trade-off between perplexity and model size over a range of N-gram orders. Finally, we tried various pruning methods to further reduce model size, and then compared the best result we obtained using previous techniques with the best result we obtained using our new techniques.

5.1 Data and Base Smoothing Methods

For training, parameter optimzation, and test data we used English text from the WMT-06 Europarl corpus (Koehn and Monz, 2006). We trained on the designated 1,003,349 sentences (27,493,499 words) of English language model training data, and used 2000 sentences each for testing and parameter optimization, from the English half of the English-French dev and devtest data sets.

We conducted our experiments on seven language model smoothing methods. Five of these are well-known: (1) interpolated absolute discounting with one discount per N-gram length, estimated according to the formula derived by Ney et al. (1994); (2) Katz backoff with Good-Turing discounts for N-grams occurring 5 times or less; (3) backoff absolute discounting with Ney et al. formula discounts; (4) backoff absolute discounting with one discount used for all N-gram lengths, optimized on held-out data; (5) modified interpolated Kneser-Ney smoothing with three discounts per N-gram length, estimated according to the formulas suggested by Chen and Goodman (1998).

We also experimented with two variants of a new smoothing method that we have recently developed. Full details of the new method are given elsewhere (Moore and Quirk, 2009), but since it is not well-known, we summarize the method here. Smoothed N-gram probabilities are defined by the formulas shown in Figure 1, for all n such that $N \geq n \geq 2$,[3] where N is the greatest N-gram length used in the model. The novelty of this model is that, while it is an interpolated model, the interpolation weights β for the lower-order model

[3]For $n = 2$, we take the expression $p(w_n|w_2 \dots w_{n-1})$ to denote a unigram probability estimate $p(w_2)$.

	Method	base PP	select PP	percent change
1	interp-AD-fix	62.6	61.6	-1.6
2	Katz backoff	59.8	56.1	-7.9
3	backoff-AD-fix	59.9	54.3	-9.3
4	backoff-AD-opt	58.8	54.4	-7.5
5	KN-mod-fix	51.6	54.6	+5.8
6	new-fix	56.1	52.1	-7.1
7	new-opt	53.7	52.0	-3.3

Table 1: Perplexity results for N-gram selection

are not constrained to match the backoff weights γ for the lower-order model. This allows the interpolation weights to be set independently of the discounts D, with the backoff weights being adjusted to normalize the resulting distributions.

The motivation for this is to let the D parameters correct for potential overestimation of the probabilities for observed N-grams, while the δ parameter (which determines the α and β interpolation parameters) somewhat independently corrects for quantization errors caused by the fact that only certain probabilities can be derived from integer observed counts, even after discounting. δ is interpretable as the estimated mean quantization error for each distinct count for a given context.

We tested two variants of the new method, (6) one in which the D parameters and the δ parameter are set by fixed criteria, and (7) one in which a single value for all D parameters and the value of the δ parameter are optimized on held-out data. For the fixed value of δ, we assume that, since the distance between possible N-gram counts, after discounting, is approximately 1.0, their mean quantization error would be approximately 0.5. For the fixed discount parameters, we use three values for each N-gram length: D_1 for N-grams whose count is 1, D_2 for N-grams whose count is 2, and D_3 for N-grams whose count is 3 or more. We set these values to be the discounts for 1-counts, 2-counts, and 3-counts estimated by the Good-Turing method. This yields the formula

$$D_r = r - (r+1)\frac{N_{r+1}}{N_r},$$

for $1 \leq r \leq 3$, where N_r is the number of distinct N-grams of the length in question occuring r times in the training set.

In all experiments, the unigram language model is an un-smoothed, closed-vocabulary MLE model. We use this unigram model, because there is no simple, principled way of assigning probabilities to individual out-of-vocabulary (OOV) words. The only principled solution to this problem that we are aware of is to use a character-based model, but this seems overly complicated for something that is orthogonal to the main points of this study, and of minor practical importance. Since we make no provision for OOV words in the models, OOV words are also omitted from all perplexity measurements. Thus, the perplexity numbers are systematically lower than they would be if OOVs were taken into account, but they are all comparable in this regard.

5.2 Results for Significance-Based N-gram Selection

Table 1 shows the minimum perplexity (with respect to N-gram order) of language models up to 7-grams for each of the seven smoothing methods discussed above, with and without significance-based N-gram selection. N-gram selection improved the perplexity of all models, except for modified KN. The lowest overall perplexity remains that of the base modified KN method, but with N-gram selection, the two variants of the new smoothing method come very close to it.

If we cared only about perplexity, that would be the end of the story, but we also care about language model size. The results in Table 1 were obtained on models estimated using just the counts needed to cover the parameter optimization and test sets; so to accurately measure model size, we trained full language models using base modifed KN, and the two variants of the new method with N-gram selection. The resulting sizes of the models represented in backoff form (in terms of total number of probability and backoff parameters) are shown in Figure 2 as function of N-gram length, from trigrams up to 7-grams for KN and up to 10-grams for the two new models. We see that beyond 4-grams the model sizes diverge dramatically, with the new models incorporating N-gram selection leveling off, but the modified KN model (or any standard model) continuing to grow in size, apparently linearly in the N-gram order.

In Figure 3, we show the relationship between perplexity and model size for the same three models, varying N-gram order. We see that between about 20 million and 45 million parameters, both of the new models incorporating significance-

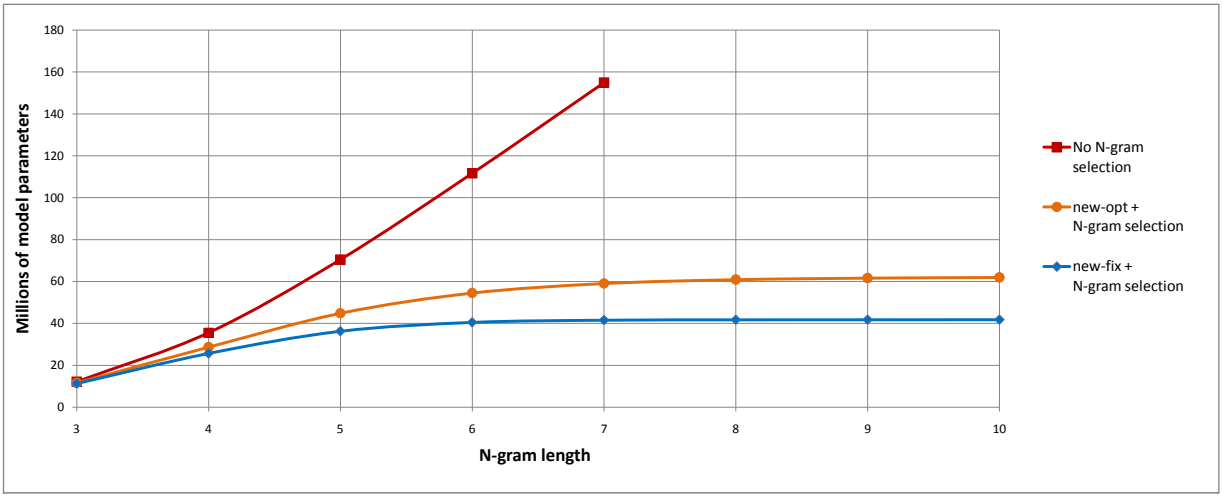

Figure 2: Model size vs. N-gram length

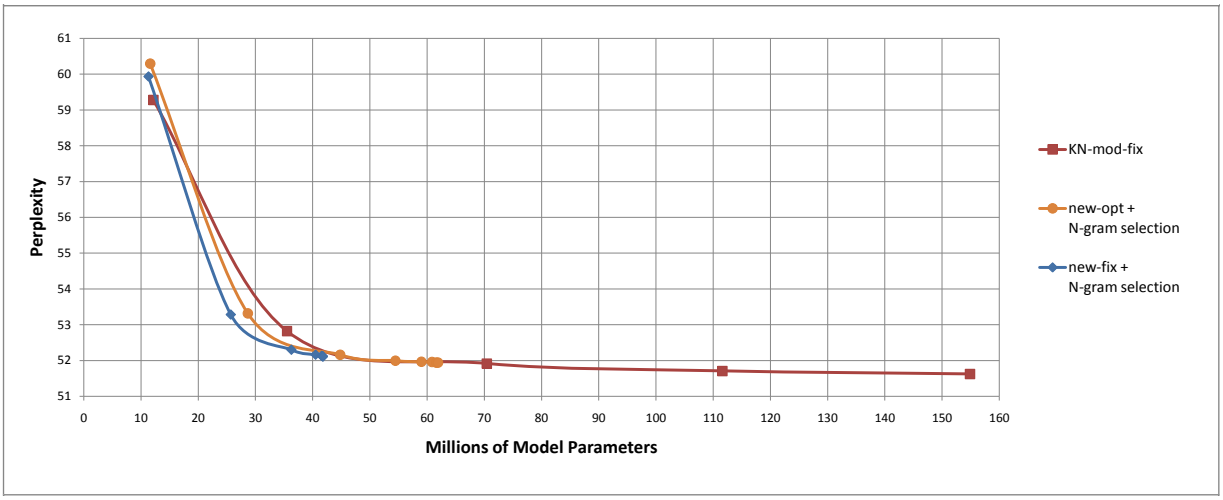

Figure 3: Perplexity vs. model size

based N-gram selection seem to outperform modified KN, and that the best of the three is, in fact, the new model with fixed parameter values.

5.3 Results for Additional Pruning

We further tested modified KN smoothing, and our new smoothing method with fixed parameter values and significance-based N-gram selection, with additional pruning. We compared several pruning methods on trigram models: count cutoffs, REP,[4] and our two modified versions of WDP.

Figure 4 shows the resulting combinations of perplexity and model size for REP and modified WDP at various pruning thresholds, and for count cutoffs of 1, 2, and 3 for both bigrams and trigrams ($n > 1$) and for trigrams only ($n > 2$), applied to our new smoothing method with fixed parameter values, together with significance-based N-gram selection. Overall, modified WDP with fully-smoothed joint probability estimates performs the best. It is has lower perplexity than count cutoffs at all model sizes tested, and is about equal to REP at very severe pruning levels and superior to REP with less pruning. Modified WDP with fully-smoothed joint probabilities is about equal to modified WDP with partially-smoothed joint probabilities at the highest and lowest pruning levels tested, but superior in between.

Figure 4 also shows the result of applying modified WDP with fully-smoothed joint probabilities to our new smoothing method *without* significance-based N-gram selection, to test whether the former subsumes the gains from the latter. We see that modified WDP does not render

[4]Thanks to Asela Gunawardana for the use of his REP tool.

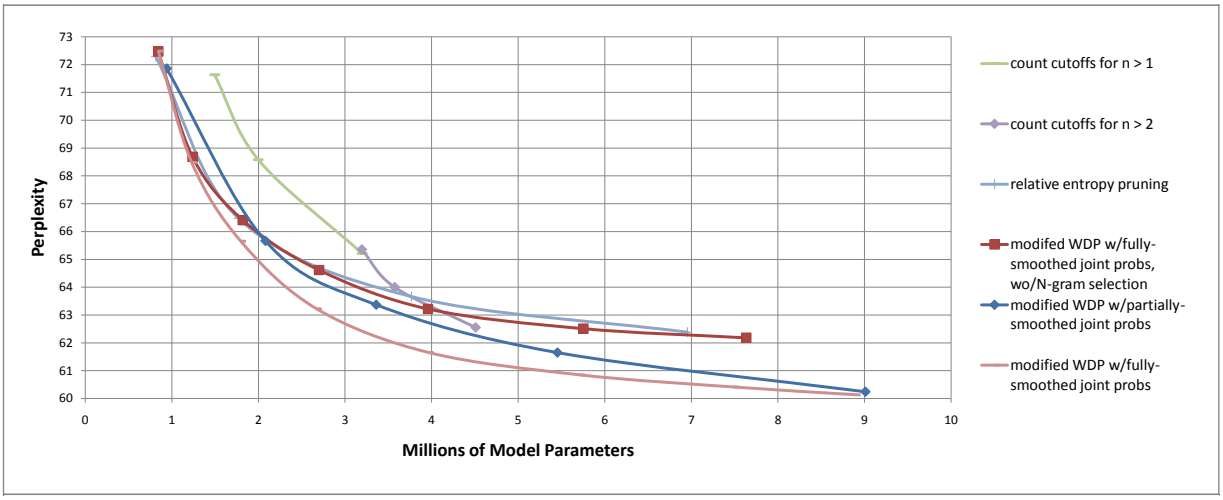

Figure 4: Pruning methods for new smoothing technique with N-gram selection

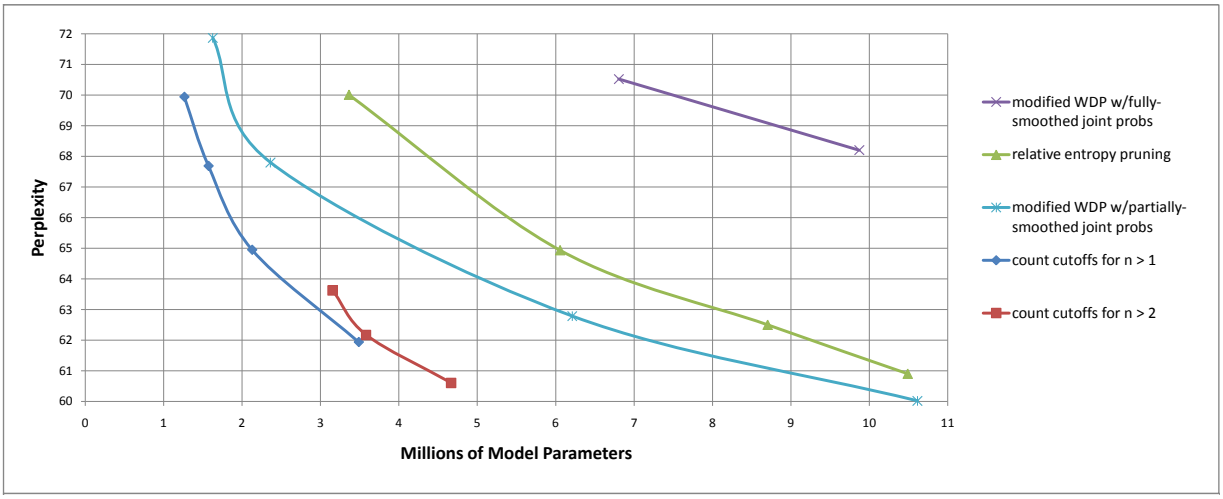

Figure 5: Pruning methods for modified KN smoothing

N-gram selection redundant except at very severe pruning levels, much like REP.

Figure 5 shows the results of applying the same four pruning methods to KN smoothing. Count cutoffs clearly perform the best with KN smoothing. It is interesting to note, however, that—contrary to the results for our new smoothing method—with KN smoothing, modified WDP with partially-smoothed joint probabilities is significantly better than either REP or modified WDP with fully-smoothed joint probabilities. We believe this is due to the fact that the latter two methods both estimate the joint probabilities by chaining the lower-order conditional probabilities from the fully-smoothed model, which in the case of KN smoothing are designed specifically to cover N-grams that have not been observed, and are poor estimates for the probabilities of lower-order N-grams that do occcur in the training data.

Finally, we compared the new smoothing method with N-gram selection and modified WDP with fully-smoothed joint probabilities against modified KN smoothing with count cutoffs, using combinations of pruning parameter values and N-gram order that yielded the best size/perplexity trade-offs. The results are shown in Figure 6. At all model sizes within the range of these experiments, the new method with significance-based N-gram selection and modified WDP had lower perplexity than modifed KN with count cutoffs—up to about 8% lower at greater pruning levels.

This experiment also suggests that the size/perplexity trade-off is easier to optimize for our new combination of smoothing, N-gram selection, and modified WDP, than for KN smoothing with count cut-offs. Table 2 shows the

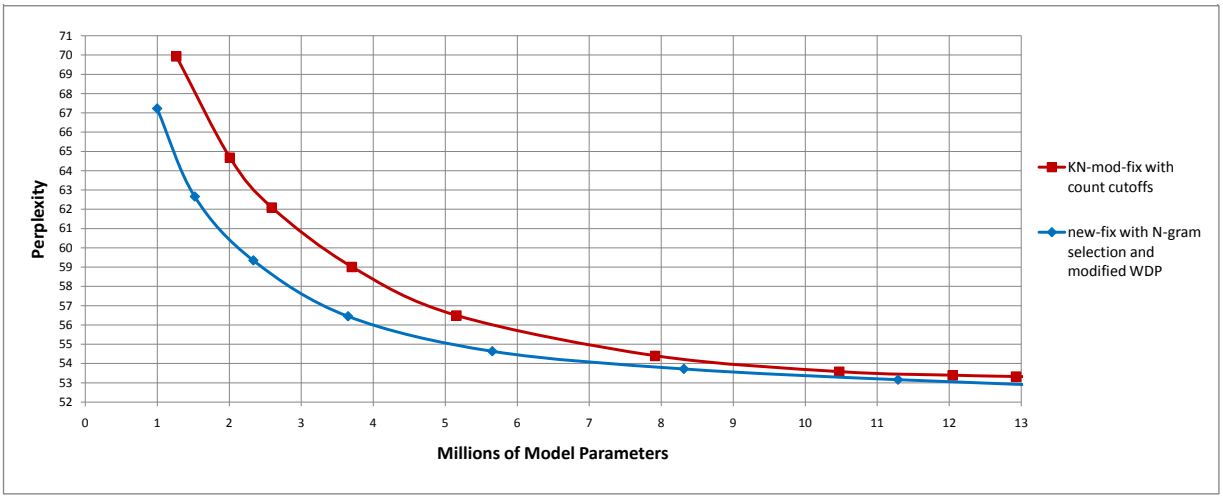

Figure 6: Comparison of two best pruned language models

PP	N	CC	$n >$
69.9	3	4	1
64.7	4	4	1
62.1	4	3	1
59.0	4	2	1
56.5	4	2	2
54.4	4	1	2
53.6	5	1	2
53.4	6	1	2
53.3	7	1	2

Table 2: Optimal pruning parameters for KN-mod-fix with count cutoffs

PP	N	threshold
67.2	10	$10^{-6.5}$
62.7	10	$10^{-6.75}$
59.3	10	$10^{-7.0}$
56.4	10	$10^{-7.25}$
54.6	10	$10^{-7.5}$
53.7	10	$10^{-7.75}$
53.2	10	$10^{-8.0}$

Table 3: Optimal pruning parameters for new-fix with N-gram selection and modified WDP

perplexity (PP), maximum N-gram length (N), count cutoff (CC), and N-gram lengths to which the count cutoffs are applied ($n >$) for the points on the curve for pruned KN in Figure 6. Although some tendencies are discernable, it seems clear that a significant part of the space of combinations of N, CC, and "$n >$" parameter values must be searched to find the best points for trading off perplexity against model size. Table 3 shows maximum N-gram length and pruning threshold values for the points on the corresponding curve for our new approach. Here the situation is much simpler. The best trade-off points are found by varying the pruning threshold, and including in the model all N-grams that pass the pruning threshold, regardless of N-gram length.

6 Conclusions

We have shown that significance-based N-gram selection can simultaneously reduce both model size and perplexity when applied to a number of language model smoothing methods, including the widely-used Katz backoff and absolute discounting methods. We are not aware of any other technique that does this. We also found that, when combined with a new smoothing method and a novel variant of weighted difference pruning, our N-gram selection method outperformed modified Kneser-Ney smoothing—using the best form of pruning we found for that approach—with respect to the trade-off between model size and model quality.

As our next steps, first, we need to verify that the results obtained on a moderate-sized training corpus are repeatable on much larger corpora. Second, we plan to extend this work to incorporate language model size reduction by word clustering, which has been shown by Goodman and Gao (2000) to produce additional gains when combined with previous methods of language model pruning.

References

Brants, Thorsten, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of EMNLP 2007*, 858–867.

Chen, Stanley F., and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

Goodman, Joshua, and Jianfeng Gao. 2000. Language model size reduction by pruning and clustering. In *Proceedings of ICSLP 2000*, 110–113.

Jedynak, Bruno M., and Sanjeev Khudanpur. 2005. Maximum likelihood set for estimating a probability mass function. *Neural Computation* 17, 1–23.

Koehn, Philipp, and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings of WMT 2006*, 102–121.

Moore, Robert C., and Chris Quirk. 2009. Improved smoothing for N-gram language models based on ordinary counts. In *Proceedings of ACL-IJCNLP 2009*.

Ney, Hermann, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8, 1–38.

Ron, Dana, Yoram Singer, and Naftali Tishby. 1996. The power of amnesia: learning probabilistic automata with variable memory length. *Machine Learning*, 25, 117–149.

Rosenfeld, Ronald, and Xuedong Huang. 1993. Improvements in stochastic language modeling. In *Proceedings of HLT 1993*, 107–111.

Seymore, Kristie, and Ronald Rosenfeld. 1996. Scalable Trigram Backoff Language Models. In *Proceedings of ICSLP 1996*. 232–235.

Stolcke, Andreas. 1998. Entropy-based pruning of backoff language models. In *Proceedings, DARPA News Transcription and Understanding Workshop 1998*, 270–274.