# Study on Interaction between Entropy Pruning and Kneser-Ney Smoothing

*Ciprian Chelba, Thorsten Brants, Will Neveitt, Peng Xu*

Google, Inc., 1600 Amphiteatre Pkwy, Mountain View, CA 94043, USA

## Abstract

The paper presents an in-depth analysis of a less known interaction between Kneser-Ney smoothing and entropy pruning that leads to severe degradation in language model performance under aggressive pruning regimes. Experiments in a data-rich setup such as `google.com` voice search show a significant impact in WER as well: pruning Kneser-Ney and Katz models to 0.1% of their original impacts speech recognition accuracy significantly, approx. 10% relative.

## 1. Introduction

A typical voice search language model used in our system for the US English query stream is trained as follows:

- vocabulary size: 1M words, OoV rate 0.57%
- training data: 230B words; we use only correct queries, i.e. those that did not trigger spelling correction

The resulting size, as well as its performance on unseen query data (10k queries) when using Katz smoothing is shown in Table 1. We note a few key aspects:

- the first pass LM (15 million $n$-grams) requires very aggressive pruning—to about 0.1% of its unpruned size—in order to make it usable in standard ASR decoders
- the perplexity hit taken by pruning the LM is significant, 50% relative; similarly, the 3-gram hit ratio is halved
- the unpruned model has excellent $n$-gram hit ratios on unseen test data: 77% for $n = 5$, and 97% for $n = 3$.

| $n$ | Size | Pruning | PPL | hit ratios |
|---|---|---|---|---|
| 3 | 15M | entropy (Stolcke) | 190 | 47/93/100 |
| 3 | 7.7B | none | 132 | 97/99/100 |
| 5 | 12.7B | cut-off (1-1-2-2-2) | 108 | 77/88/97/99/100 |

Table 1: Typical voice search LM, Katz smoothing: the LM is trained on 230 billion words using a vocabulary of 1 million words, achieving out-of-vocabulary rate of 0.57% on test data.

The use of Katz smoothing in our language model is not accidental: we examined the interaction between Stolcke pruning and various $n$-gram LM smoothing techniques for aggressive pruning regimes which cut the original LM to under 1% of its original size. The main finding is that the increasingly popular family of Kneser-Ney [1] smoothing methods is in fact poorly suited for such aggressive pruning regimes, as explained in Section 2. When evaluated in terms of both perplexity and ASR word error rate, the more traditional ones, e.g. Katz/Good-Turing [2] perform significantly better after pruning, as the experiments in Section 2.3 show.

We wish to emphasize that this is not a new result, [3] also pointed out this behavior of Kneser-Ney models and proposed a solution that alleviates the problem by growing the LM, instead of pruning it. [4] also suggests a variation of Kneser-Ney smoothing more suitable for pruning.

## 2. Pruning Interaction with Smoothing

As mentioned in the introduction, the use of Katz smoothing in our language model is not accidental. We examined the interaction between Stolcke pruning and various $n$-gram LM smoothing techniques for aggressive pruning regimes which cut the original LM to under 1% of its original size. The main finding is that the increasingly popular family of Kneser-Ney [1] smoothing methods is in fact poorly suited for such aggressive pruning regimes. This section attempts to explain what causes this.

### 2.1. Language Model Smoothing

The goal of language model smoothing is to make sure that the $n$-gram language model probability $p(w|h)$ satisfies the constraint: $p(w|h) > \epsilon, \forall(h, w) \in V^n$, and some $\epsilon > 0$, and thus will assign a non-zero probability to any string of words $W = w_1 \dots w_l$[1] belonging to a given vocabulary, $W \in V^*$.

There has been a large amount of work on deriving various smoothing methods for $n$-gram language modeling. An excellent review is presented in [5]. Most smoothing methods—in this paper we compare Katz/Good-Turing [2], Ney [6], Witten-Bell [7], Ristad [8] as implemented in the SRILM toolkit [9]—do not diverge too much from the relative frequency estimates, with one notable exception: Kneser-Ney [1] and its variants, which uses the left diversity count for a given $n$-gram as the count for lower order $n$-grams, instead of the regular maximum likelihood count.

Section 2.7 of [5] (pp. 15-16) derives Kneser-Ney smoothing by imposing a consistency constraint between the unigram marginal on the predicted word and its relative frequency estimate, under the assumption that the $n$-gram context probability is equal to the relative frequency estimate.

However, as the next section explains in more detail, Stolcke pruning replaces the relative frequency estimates for the context probability $f(h)$ with the estimates built by chaining the lower order estimates provided by the model, which by construction diverge significantly from their maximum likelihood counterparts. This is one source of mismatch between Stolcke pruning and Kneser-Ney smoothing.

The same section (2.7 of [5], pp. 15-16) also makes the argument that since the lower order estimates are used only when backing-off from the maximum order—the latter being typically a smoothed version of the maximum likelihood/relative frequency estimate—a better choice for them is to *complement it* rather than stay close to the maximum likelihood estimate.

[1] The actual words in the sentence are embedded in sentence start/end symbols which are present as distinguished words in the language model vocabulary $V$.

However, with aggressive pruning many of the highest order $n$-grams may be discarded. The model will then back-off, possibly at no cost, to the lower order estimates which are far from the maximum likelihood ones and will thus perform poorly in perplexity. This is a second source of mismatch between entropy pruning and Kneser-Ney smoothing.

Our experiments confirm that for models in the Kneser-Ney smoothing family, aggressive Stolcke pruning severely damages the model's performance in perplexity as well as word-error-rate. The deterioration can be attributed to the following factors:

1. poor model estimates for the context frequency $f(h)$ when chaining lower order estimates:
$$f(h) \not\simeq p_{KN}(h_1) \cdot \ldots \cdot p_{KN}(h_n|h_1 \ldots h_{n-1})$$
2. after pruning, a significant number of predictions on test data are likely to be made using lower order estimates, which diverge significantly from maximum-likelihood ones by construction.

## 2.2. Language Model Pruning

A very simple form of reducing the LM size is count cut-off pruning—removing the n-grams whose count is below a certain threshold. The method doesn't allow for fine grained control over the size of the resulting LM, and so entropy pruning techniques are by far the most popular.

Stolcke pruning [10] is probably the most popular technique used for trimming the size of back-off $n$-gram language models in speech recognition. Its most appealing advantage over Seymore-Rosenfeld pruning [11] is *self-containedness*, namely that it operates on the back-off model alone, not needing to store the $n$-gram counts. A more recent attempt at improved LM pruning is presented in [12].

### 2.2.1. Stolcke Pruning

Section 3 of [10] explains the algorithm in detail. We reproduce a high level description here for completeness' sake: let $(h, w)$ be an $n$-gram under consideration for pruning. Its current probability estimate in the model is $p(w|h)$; were we to remove it from the model, this would become $p'(w|h) = \alpha'(h)p(w|h')$ where $h'$ is the context obtained by dropping the left-most word in $h$. The decision on whether we should prune or retain the $n$-gram $(h, w)$ in the model is based on the relative entropy

$$D[p(\cdot|h) \| p'(\cdot|h)] = p(h) \sum_w p(w|h) \log \frac{p(w|h)}{p'(w|h)}$$

We note the presence of the term $p(h)$ for the context probability which is computed using the chain rule and lower order $n$-gram estimates in [10], whereas [11] uses the relative frequency in the training data: $p(h) = f(h)$ .

For Kneser-Ney models the lower order estimates for a given $n$-gram are based on left-diversity counts, and thus diverge significantly from the relative frequency estimate.

A blatant example is the sentence beginning context <s>: its left-diversity count is 1 (always preceded by </s>) and thus the Kneser-Ney estimate for its unigram probability in a 2-gram model will be very different from the relative frequency estimate. It can be easily verified that $\log p_{KN}(\texttt{<s>})$ is upper bounded by $-\log |V|$, where $|V|$ is the vocabulary size[2], instead of being roughly equal to $-\log L$, where $L$ is the average sentence length in the training data. Typical values are $|V| = 10^5$, $L = 20$ which will result in many more 2-grams $(\texttt{<s>}, w)$ being pruned from a Kneser-Ney model than its Good-Turing counterpart, for example.

Section 4 in [10] contrasted Stolcke pruning against Seymore-Rosenfeld pruning for Katz models and found no significant differences in either perplexity or ASR word error rate on Katz smoothed models, even when pruned to 2% of their original size. However, the author clearly states that no study of the interaction between various smoothing techniques and pruning has been carried out. As our experiments reported below show, the perplexity of models in the Kneser-Ney family of smoothing techniques degrades very fast with pruning. This is partly due to the mis-estimate for $p(h)$ outlined previously: using the relative frequency $f(h)$ instead, e.g. Seymore-Rosenfeld pruning [11], improves the behavior with pruning but does not fully fix the problem, as anticipated in Section 2.1, and confirmed experimentally in Section 2.3.2.

### 2.2.2. Seymore-Rosenfeld Pruning

Seymore-Rosenfeld pruning is an alternative to Stolcke pruning that relies on the relative frequency in the training data for a given context $f(h)$ instead of the probability $P(h)$ computed from lower order estimates. For Kneser-Ney models this eliminates one source of potential problems in pruning: since the $P(h)$ calculation involves only lower order $n$-gram estimates it will use the diversity based estimates, which are quite different from the relative frequency ones.

## 2.3. Pruning Experiments

Whenever possible, we ran our experiments investigating the interaction of smoothing with pruning on Broadcast News data and used the SRILM toolkit for easy reproduce ability. For training we used 128M words, and for testing 692k words. The 143k word vocabulary was estimated by retaining only the 1-grams whose count in the training data was higher than 1.

### 2.3.1. Stolcke Pruning

A first batch of experiments compared the PPL of various models—we varied both the $n$-gram order $n$ and the smoothing technique involved—before and after pruning to about the same size, measured in number of $n$-grams. Tables 2-3 shows the results for $n = 3, 4$, respectively; no tuning was performed for any smoothing technique, we took the default values as implemented by the SRILM toolkit.

As anticipated, the Kneser-Ney family of models is hurt significantly by aggressive Stolcke pruning. Notably, the relative increase in perplexity for Kneser-Ney models is twice as large as for the other smoothing techniques evaluated—100% vs. 54% and 135% vs. 65% rel. increase for $n = 3, 4$, respectively.

A secondary observation is that the particular choice of smoothing technique is not so important: with the exception of Ristad's smoothing, the unpruned models are within 5% relative of the best PPL value—attained by the Kneser-Ney model. The same holds for the pruned models, with the exception of the Kneser-Ney models.

In a second batch of experiments we took a closer look at the change in perplexity with the number of $n$-grams in the pruned model. Figure 1 shows the results for 4-gram models. Although the unpruned Kneser-Ney models start from a slightly lower perplexity than the Katz model, they degrade faster with

[2] In practice it ends up being much lower since the left diversity for many 1-grams is much larger than 1

| **3-gram**<br>LM smoothing | Perplexity<br>un/pruned | Perplexity<br>rel. increase | No. n-grams<br>un/pruned |
|---|---|---|---|
| Ney | 130.1/201.3 | 54.76% | 18,483,341/371,683 |
| Ney, Interpolated | 129.5/201.9 | 55.86% | 18,483,341/375,849 |
| Witten-Bell | 129.5/200.1 | 54.54% | 18,483,341/370,923 |
| Witten-Bell, Interpolated | 131.3/205.7 | 56.69% | 18,483,341/389,102 |
| Ristad | 135.9/206.7 | 52.08% | 18,483,341/389,439 |
| Katz (Good-Turing) | 129.5/199.1 | 53.75% | 18,483,341/394,026 |
| Kneser-Ney | 125.5/256.4 | 104.27% | 18,483,341/395,733 |
| Kneser-Ney, Interpolated | 126.6/252.9 | 99.75% | 18,483,341/390,116 |
| Kneser-Ney (Chen-Goodman) | 126.6/257.1 | 103.09% | 18,483,341/384,279 |
| Kneser-Ney (Chen-Goodman), Interpolated | 124.5/250.0 | 100.80% | 18,483,341/389,103 |

Table 2: 3-gram model perplexity degradation after aggressive *Stolcke pruning* (2% of original size), for various smoothing methods

| **4-gram**<br>LM smoothing | Perplexity<br>un/pruned | Perplexity<br>rel. increase | No. n-grams<br>un/pruned |
|---|---|---|---|
| Ney | 120.5/197.3 | 63.75% | 31,095,260/383,387 |
| Ney, Interpolated | 119.8/198.1 | 65.33% | 31,095,260/386,214 |
| Witten-Bell | 118.8/196.3 | 65.16% | 31,095,260/380,372 |
| Witten-Bell, Interpolated | 121.6/202.3 | 66.31% | 31,095,260/396,424 |
| Ristad | 126.4/203.6 | 61.09% | 31,095,260/395,639 |
| Katz (Good-Turing) | 119.8/198.1 | 65.33% | 31,095,260/386,214 |
| Kneser-Ney | 114.5/285.1 | 148.98% | 30,360,224/388,184 |
| Kneser-Ney, Interpolated | 115.8/274.3 | 136.93% | 30,360,224/398,750 |
| Kneser-Ney (Chen-Goodman) | 116.3/280.6 | 141.23% | 30,360,224/396,217 |
| Kneser-Ney (Chen-Goodman), Interpolated | 112.8/270.7 | 139.98% | 30,360,224/399,129 |

Table 3: 4-gram model perplexity degradation after aggressive *Stolcke pruning* (1.3% of original size) for various smoothing methods

pruning.

In experiments for voice search, we trained much larger language models on different data than that used in the experiments reported in this section. We observed similar relative differences in perplexity between much bigger Kneser-Ney/Katz models—after pruning them to 0.1% of their original size. The differences were also found to impact speech recognition accuracy significantly, approx. 10% relative.

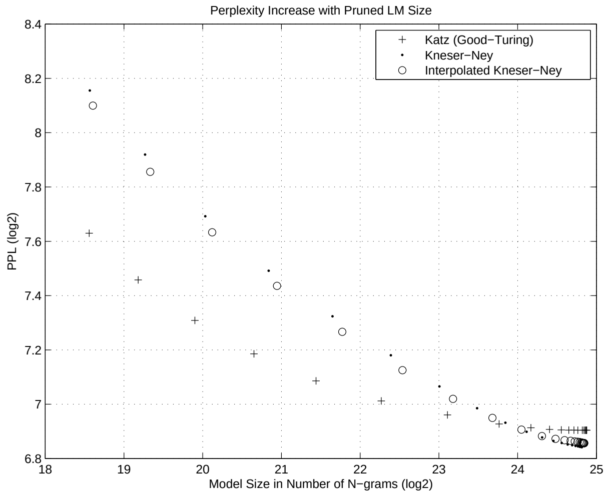

Figure 1: Stolcke pruned 4-gram model perplexity as a function of model size (no. $n$-grams) for Katz, Kneser-Ney and Interpolated Kneser-Ney models

### 2.3.2. Seymore-Rosenfeld Pruning

We contrasted Stolcke and Seymore-Rosenfeld pruning as implemented in our own language modeling tools [13] since the SRILM toolkit doesn't implement the latter. A second batch of experiments used our language model training infrastructure to estimate both Katz (Good-Turing) and Kneser-Ney 4-gram models, and prune them using both Stolcke and Seymore-Rosenfeld pruning. The LM training data and vocabulary are fixed to the same ones used in the previous experiments. Table 4 presents the results. For such aggressive pruning regimes, Seymore-Rosenfeld pruning is significantly better than Stolcke for Kneser-Ney models. However, it does not fully bridge the gap relative to Katz smoothing: as the last two rows of the table illustrate, there is still a significant difference between Katz and Kneser-Ney smoothing after pruning, even though the context frequency estimation problem has been factored out by using Seymore-Rosenfeld pruning. As outlined at the end of Section 2.1, we can explain the poor PPL performance of the pruned model on the direct use of diversity-based estimates in the Kneser-Ney model, without any adjustment of the lower order estimates used after back-off. Since the difference between Katz and Kneser-Ney is very small on unpruned models, and significant on pruned models we chose to use Katz smoothing when building a LM for voice search.

## 3. Acknowledgments

Thanks to everybody on the Google speech team who helped with the voice search experiments, in particular Johan Schalkwyk, Vida Ha, Boulos Harb and Carolina Parada for their in-

| Toolkit | Pruning | Smoothing | PPL | No. n-grams |
|---|---|---|---|---|
| SRILM | none (1-1-2-2 cutoff) | Katz (Good-Turing) | 130 | 31,095,260 |
| SRILM | none (1-1-2-2 cutoff) | Kneser-Ney | 116 | 30,360,224 |
| SRILM | Stolcke (1-1-2-2 cutoff) | Katz (Good-Turing) | 198 | 386,214 |
| SRILM | Stolcke (1-1-2-2 cutoff) | Kneser-Ney | 274 | 398,750 |
| Google LM | none (1-1-1-1 cutoff) | Katz (Good-Turing) | 119 | 111,323,496 |
| Google LM | none (1-1-1-1 cutoff) | Kneser-Ney | 111 | 112,127,986 |
| Google LM | none (1-1-2-2 cutoff) | Katz (Good-Turing) | 121 | 31,095,260 |
| Google LM | none (1-1-2-2 cutoff) | Kneser-Ney | 148 | 31,429,487 |
| Google LM | Stolcke (1-1-2-2 cutoff) | Katz (Good-Turing) | 210 | 387,150 |
| Google LM | Stolcke (1-1-2-2 cutoff) | Kneser-Ney | 336 | 395,494 |
| Google LM | Seymore-Rosenfeld (1-1-2-2 cutoff) | Katz (Good-Turing) | 205 | 386,437 |
| Google LM | Seymore-Rosenfeld (1-1-2-2 cutoff) | Kneser-Ney | 247 | 389,536 |

Table 4: Comparison of Stolcke and Seymore-Rosenfeld pruning for Katz (Good-Turing) and Kneser-Ney $4$-gram models estimated using Google LM training tools.

strumental role on training the voice search language model and evaluating it.

## 4. Conclusions

We have confirmed in a different experimental setup the less known fact that aggressive entropy pruning (in particular Stolcke pruning) significantly degrades language models built using Kneser-Ney smoothing, whereas Katz smoothing performs much better. Part of the loss of Stolcke pruning can be regained by using Seymore-Rosenfeld pruning, which uses counts instead of chaining to obtain $p(h)$. The remaining difference between Katz and Kneser-Ney smoothing may be addressed in future work with a mechanism along the lines of Eq. (9) of [4]. It changes the backoff distribution to use a combination of left-diversity count and raw count, where the weighting of the mix depends on the amount of pruning done.

As a concluding remark, although no WER results were reported in this paper, we generally see excellent correlation with PPL under various pruning regimes, as long as the training set and vocabulary stays constant, see [14] (presentation slides).

## 5. References

[1] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 1995, pp. 181–184.

[2] S. 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*, vol. 35, no. 3, March 1987, pp. 400–01.

[3] V. Siivola, T. Hirsimaki, and S. Virpioja, “On Growing and Pruning Kneser–Ney Smoothed $N$-Gram Models,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1617–1624, 2007.

[4] R. Kneser, “Statistical language modeling using a variable context length,” in *Proc. ICSLP ’96*, vol. 1, Philadelphia, PA, 1996, pp. 494–497. [Online]. Available: citeseer.ist.psu.edu/kneser96statistical.html

[5] S. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Harvard University, Tech. Rep. TR-10-98, August 1998. [Online]. Available: citeseer.ist.psu.edu/733490.html

[6] H. Ney and U. Essen, “On smoothing techniques for bigram-based natural language modelling,” in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, 1991, pp. 825–828.

[7] I. H. Witten and T. C. Bell, “The zero-frequency problem: estimating the probabilities of novelevents in adaptive text compression,” *Information Theory, IEEE Transactions on*, vol. 37, no. 4, pp. 1085–1094, 1991.

[8] E. S. Ristad, “A natural law of succession,” Princeton University, Tech. Rep. CS-TR-495-95, May 1995.

[9] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, September 2002, pp. 901–904.

[10] ——, “Entropy-based pruning of back-off language models,” in *Proceedings of News Transcription and Understanding Workshop*. Lansdowne, VA: DARPA, 1998, pp. 270–274.

[11] K. Seymore and R. Rosenfeld, “Scalable back-off language models,” in *Proceedings ICSLP*, vol. 1, Philadelphia, 1996, pp. 232–235.

[12] J. Gao and M. Zhang, “Improving language model size reduction using better pruning criteria,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics Morristown, NJ, USA, 2001, pp. 176–182.

[13] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, “Large language models in machine translation,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 858–867. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1090

[14] B. Harb, C. Chelba, J. Dean, and S. Ghemawat, “Back-off language model compression,” in *Proceedings of Interspeech*, Brighton, UK, 2009, pp. 325–355. [Online]. Available: http://research.google.com/pubs/pub35612.html