Comparison of N-gram Smoothing Methods for Speech Recognition DT2119 Project Report

Henrik Karlsson
henrik10@kth.se

Abstract

We compared a couple of n-gram smoothing methods: Katz smoothing, Witten-Bell smoothing (interpolation and backoff) and modified Kneser-Ney smoothing (interpolation and backoff). These methods were compared intrinsically and extrinsically with cross-entropy and word error rate respectively. The smoothing methods were compared with regards to different data sizes and found that it had can have a significant impact on the relative performance of the different methods. In general the difference in cross-entropy decreased with increased data size while the difference in word error can shift rapidly. Additionally, we showed that modified Kneser-Ney smoothing is better consistently better then other smoothing methods and that the backoff version of the Kneser-Ney smoothing might be better than the interpolated version for speech recognition. Finally, results showed smoothing methods have a significant impact on word error rate, improving it with up to 2%.

1 Introduction

An n-gram language model tries to estimate the probability of a each word given the previous $n-1$ words. This is denoted as the conditional probability $P(w_i|w_{i-n+1}^{i-1})$ and the maximum likelihood estimate is

$$P_{\text{MLE}}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{c(w_{i-n+1}^{i-1})} \tag{1}$$

where $c(w_j^k)$ is the frequency of the word sequence in the training data.

Equation 1 is not used in practice due to the out-of-vocabulary (OOV) problem. The OOV problem is that equation 1 assigns zero probability to unseen n-grams and because our training data is fraction of all data, we will have many OOV n-grams. To avoid the OOV problem, we distribute some of the probability mass from seen n-grams to unseen n-grams, this technique is called smoothing or discounting.

This redistribution is called smoothing or discounting. We compared five different smoothing models based on Katz smoothing, Kneser-Ney discounting or Witten-Bell discounting. For all methods we tested backoff models and in addition to that we tested interpolated models for Kneser-Ney and Witten-Bell discounting.

Backoff and interpolation is essential for good n-gram language models [5]. Backoff is a smoothing technique where you use lower order n-grams if the data of higher order n-grams are not good enough. This is usually determined with a cutoff value k where n-grams with counts lower than k are backed off. Interpolation is a smoothing technique where you interpolate higher order n-gram with lower order n-grams.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

1.1 Katz smoothing

The Good-Turing frequency estimates [2] states that an n-gram seen r times should be treated as being seen r^* times where

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{2}$$

where n_r is the number of n-grams seen exactly r times.

Katz smoothing [3] is based on the Good-Turing frequency estimate. Katz smoothing discounts the MLE with the Good-Turing frequency estimate for all n-grams where r is less than a cutoff value. The left over probability is then distributed to lower order n-grams using backoff.

Katz smoothing was one of the most widely used smoothing techniques and has been shown to be competitive [5].

1.2 Witten-Bell discounting

The intuition behind Witten-Bell discounting is that the probability of encountering unseen n-grams is approximately the probability of encountering an n-gram for the first time in the training set. To calculate that probability we want the number of unique n-grams with a specific history w_{i-n+1}^{i-1}, let that be

$$N_{1+}(w_{i-n+1}^{i-1}) = |\{w_i : c(w_{i-n+1}^{i}) > 0\}| \tag{3}$$

We then let the Witten-Bull discounting be:

$$p_{\mathrm{WB}}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) + N_{1+}(w_{i-n+1}^{i-1})p_{\mathrm{WB}}(w_i|w_{i-n+2}^{i-1})}{\sum_{w'} c(w_{i-n+1}^{i-1}, w') + N_{1+}(w_{i-n+1}^{i-1})} \tag{4}$$

Previous studies has shown that Witten-Bell does not perform very well, being worse than Kneser-Ney discounting and Katz smoothing [5].

1.3 Kneser-Ney discounting

Kneser-Ney discounting [4] is based on absolute discounting. In absolute discounting, we subtract a fixed discount D from each non-zero count, the left over probability is then distributed with lower order n-grams as follow:

$$p_{\mathrm{abs}}(w_i|w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - D, 0)}{\sum_{w_i} c(w_{i-n+1}^{i})} + (1 - \lambda_{w_{i-n+1}^{i-1}})p_{\mathrm{abs}}(w_i|w_{i-n+2}^{i-1}) \tag{5}$$

where

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i-1}) \tag{6}$$

The difference between absolute discounting and Kneser-Ney discounting lies in the distribution of the left over probability. Kneser-Ney discounting replaces the unigram probability distribution with the probability of w_i being a continuation of an n-gram. This is done by counting the number of n-grams that ends with w_i in the training set.

$$p_{\mathrm{KN}}(w_i) = \frac{|\{w_{i-1} : c(w_{i-1}^{i}) > 0\}|}{|\{w_{i-n+1}^{i} : c(w_{i-1}^{i}) > 0\}|} \tag{7}$$

The equation for Kneser-Ney discounting is thus:

$$p_{\mathrm{KN}}(w_i|w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - D, 0)}{\sum_{w_i} c(w_{i-n+1}^{i-1}, w')} + (1 - \lambda_{w_{i-n+1}^{i-1}})p_{\mathrm{KN}}(w_i|w_{i-n+2}^{i-1}) \tag{8}$$

Kneser-Ney discounting has been shown to be one of the best smoothing methods, consistently outperforming other methods [5].

2 Experimental methodology

2.1 Smoothing implementation

We used the SRI Language Model (SRILM) toolkit version 1.5.6 to build the language models. All the n-gram language models we built and tested were of order three.

The left over probability of unigram distribution were distributed to unseen words from our dictionary.

2.1.1 Absolute discount

We implemented absolute discount as a baseline model. We implemented absolute discount with a discount coefficient $D = 0.5$ and backoff. The backoff was done when there was no history for the n-gram, i.e. when $c(w_{i-n+1}^{i}) = 0$.

2.1.2 Katz smoothing

We implemented Katz smoothing with the maximum cutoff $k = 7$ or lower depending on the discount coefficients. The cutoff value k was set to 7 unless there was a discount coefficient d_r where $r \leq k$ violating the condition $0 < d_r < 1$, if the condition was violated k was lowered until no d_r violates the condition.

2.1.3 Kneser-Ney smoothing

We built both a backoff and an interpolated version of Kneser-Ney smoothing with modified discount coefficients. In the modified Kneser-Ney smoothing [5], the discount depends on the frequency of the n-gram as follows:

$$D_r = \begin{cases} 1 - 2Y\frac{n_2}{n_1} & r = 1 \\ 2 - 3Y\frac{n_3}{n_2} & r = 2 \\ 3 - 4Y\frac{n_4}{n_3} & r \geq 3 \end{cases} \tag{9}$$

where

$$Y = \frac{n_1}{n_1 + 2n_2} \tag{10}$$

and n_r is the number of n-grams seen r times.

2.1.4 Witten-Bell smoothing

We built both a backoff and an interpolated version of the trigram Witten-Bell smoothing.

2.2 Data

We used data from Google's 1 Billion Word Language Model Benchmark corpus and Voxforge dataset. The Google corpus consists of 0.8 billion words of English text from WMT11 website [1]. From the Google corpus, we used 10 million sentences as training data and 3 million sentences for intrinsic evaluation. The Voxforge dataset consists of 12 thousand audio recordings and prompts[6]. From the Voxforge dataset we used a thousand audio recordings and prompts as a test set for extrinsic evaluation. In addition to the audio recordings and prompts, we used the Voxforge acoustic model for our ASR and Voxforge dictionary of 124,313 words as our dictionary. All of our datasets were built so that all words appears in the dictionary.

2.3 Intrinsic Evaluation

For measuring intrinsic performance we used cross-entropy. For a test set T composed of sentences $(t_1, t_2, \ldots, t_N)$ we calculate the probability of the test set as:

$$P(T) = \prod_{i=1}^{N} p(t_i) \tag{11}$$

The cross-entropy of the test set $H(T)$ is then defined as:

$$H(T) = -\frac{1}{W_T} \log_2 p(T) \tag{12}$$

where W_T is the number of words in the test set. Due to that the cross-entropy decreases with higher test set probability, lower cross-entropy is better.

2.4 Extrinsic Evaluation

An improvement in intrinsic performance does not guaranteed an improvement in the performance of a speech recognition system. Extrinsic evaluations is therefore necessary to compare language models in speech recognition.

The metrics used for extrinsic evaluation was the word error rate (WER). WER is based on the Levenshtein distance in a word level and is calculated with,

$$\text{WER} = \frac{S + I + D}{N} \tag{13}$$

where

- S is the number of substitutions,
- I is the number of insertions,
- D is the number of deletions, and
- N is the number of words in the reference.

The ASR we used was CMU Sphinx, an open source speech recognizer.

3 Results

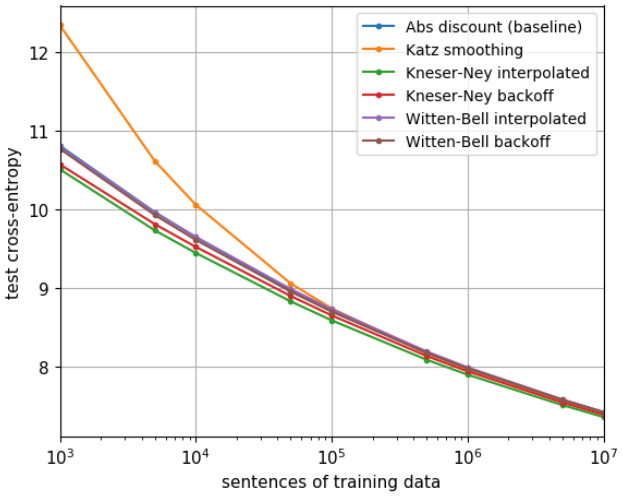

Figure 1: Intrinsic performance of various methods.

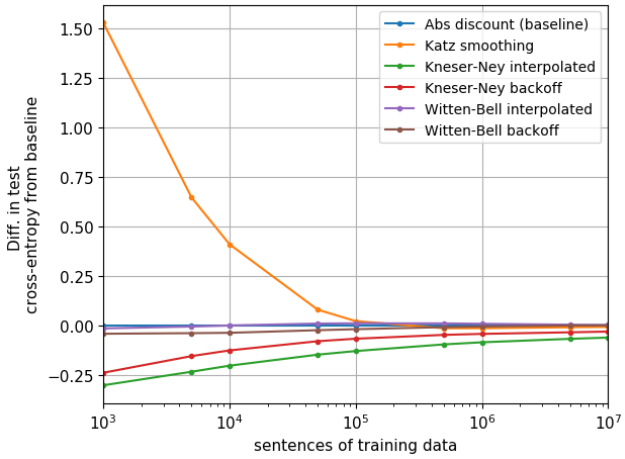

Figure 2: Intrinsic performance of various methods relative to the baseline.

[H]

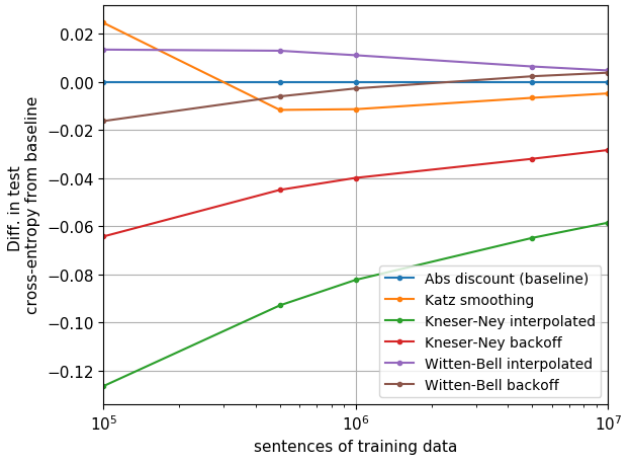

Figure 3: Intrinsic performance of various methods relative to the baseline, range 10^5 to 10^7 sentences.

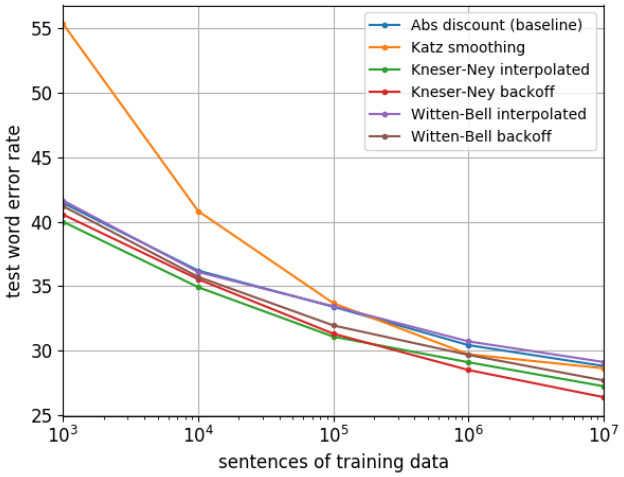

Figure 4: Extrinsic performance of various methods.

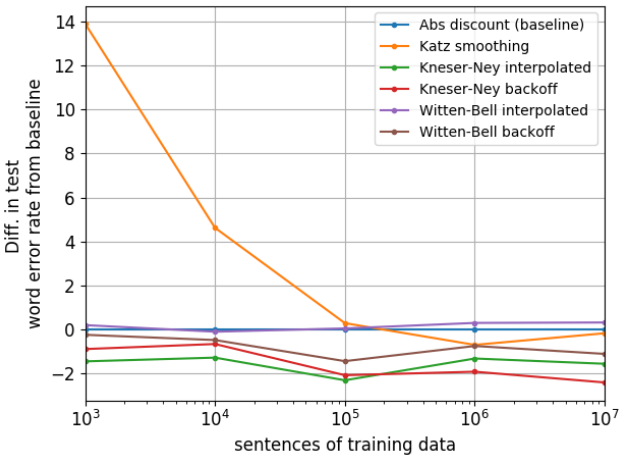

Figure 5: Extrinsic performance of various methods relative to the baseline.

4 Discussion and Conclusions

Previous work done by Chen and Goodman [5] is conflicting with our results. The work done by Chen and Goodman are very inspired us to do this study. They tested several n-gram smoothing techniques with several corpora and with many different parameters for the smoothing methods. They introduced the modified Kneser-Ney discount technique and showed that it outperformed all the other methods and we confirmed this result. However, their results for Witten-Bell smoothing and Katz smoothing is conflicting with our results.

Figures 1 to 3 shows that Katz-smoothing is better than Witten-Bell when the training data is large, this conflicts with the Chen and Goodman study. The Chen and Goodman study showed that Katz smoothing is always better than Witten-Bell smoothing and that Witten-Bell smoothing approaches the performance of Katz smoothing with larger training set. This is conflicting with figure 3 which shows that Katz smoothing is initially worse that Witten-Bell smoothing but is better when the size of the training set increases.

Figures 1 to 3 shows that Witten-Bell should be used with backoff and Kneser-Ney should be interpolated. Figure 2 shows that interpolation should be used with Kneser-Ney because the interpolated version is strictly better than the backoff version. The same figure shows that Witten-Bell is always better with backoff but in figure 3 we can see that the interpolated version is almost as good with

larger training sets. With more training data, the interpolated version might actually outperform the backoff version of Witten-Bell smoothing. These results suggest that backoff and interpolation should be chosen with regards to the underlying smoothing method and perhaps also the size of the training data.

Figure 4 to 5 shows that the cross-entropy reflects WER in general but not always. For lower training set sizes the cross-entropy reflects the WER but not for larger training sets. Backoff Kneser-Ney have better WER than the interpolated version, that is the opposite of the cross-entropy score. The Witten-Bell versions performs differently as well, backoff version being much better than the interpolated in WER while in cross-entropy they have similar performance. This might be due to the small test set of just 1000 prompts or that the prompts does not come from the same dataset as the training set.

Our experiments showed that the correlation between cross-entropy and WER is quite strong and that the interpolated Kneser-Ney smoothing is the best method in general but that the backoff version of Kneser-Ney might be better for speech recognition. To determine which is actually better for speech recognition, tests should be performed with a relevant dataset with the appropriate size. Our experiments showed that the performance of Katz smoothing might depend heavily on the datasets because it had the worst performance in our study but was among the better methods in the study made by Chen and Goodman.

References

[1] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[2] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, pages 237–264, 1953.

[3] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3): 400–401, 1987.

[4] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[5] C. Stanley and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394, 1999.

[6] Voxforge.org. Free speech... recognition (linux, windows and mac) - voxforge.org. `http://www.voxforge.org/`. accessed 06/25/2014.