A Comparative Study of Smoothing Techniques for N-Gram Models

Orkun Kınay, Murat Barkın Kınay

July 2024

Abstract

Smoothing techniques are critical in Natural Language Processing (NLP) for addressing the issue of zero probabilities in language models. This research paper presents a comparative study of various smoothing techniques applied to n-gram models of different orders. The evaluation metric used is perplexity, which measures the effectiveness of these techniques. The study investigates the performance of Additive (Laplace) Smoothing, Good-Turing Smoothing, Jelinek-Mercer Smoothing, Kneser-Ney Smoothing, and their respective improvements across unigram, bigram, and trigram models.

Introduction

Smoothing techniques are essential in language modeling to handle the problem of zero probabilities for unseen events. This paper provides a comprehensive evaluation of several smoothing techniques across different n-gram models, using perplexity as the evaluation metric. The techniques examined include Additive (Laplace) Smoothing, Good-Turing Smoothing, Jelinek-Mercer Smoothing, and Kneser-Ney Smoothing, as well as modifications and improvements to these methods.

Background

Language models assign probabilities to sequences of words, with n-gram models being a common approach. However, n-gram models suffer from the sparsity problem, where many possible word sequences are not observed in the training data, leading to zero probabilities. Smoothing techniques mitigate this issue by redistributing some probability mass to unseen events.

Smoothing Techniques

Smoothing techniques have evolved significantly over time, each addressing specific weaknesses of their predecessors while introducing new concepts. Additive

(Laplace) Smoothing is one of the simplest methods, where a small constant is added to all counts to prevent zero probabilities. This method was initially described by Jeffreys [1]. Despite its simplicity, it often disproportionately affects the probabilities of rare events, leading to overestimation.

Good-Turing Smoothing, introduced by Good [2], offered a more refined approach by adjusting probabilities based on the frequency of frequencies. This technique better estimates the probability of unseen events but requires reliable frequency estimates and can be computationally intensive for large datasets.

Jelinek-Mercer Smoothing, proposed by Jelinek and Mercer [3], introduced linear interpolation between the maximum likelihood estimate and a background model. This method effectively balances observed data with a general model, though it requires careful tuning of the interpolation parameter to achieve optimal results.

Kneser-Ney Smoothing, introduced by Kneser and Ney [4], further advanced the field by considering the diversity of contexts in which words appear. This method provided superior performance for n-gram models by ensuring that higher-order n-grams are smoothed more effectively. However, its complexity and computational demands were significant challenges.

Subsequent improvements, such as the Modified Kneser-Ney Smoothing by Chen and Goodman [5], addressed some of these challenges by refining discounting methods and handling lower-order distributions more effectively, making it one of the best-performing techniques in current NLP applications.

Methodology

To evaluate the performance of these smoothing techniques, we trained unigram, bigram, and trigram models on a large text corpus. The models were evaluated using perplexity, defined as:

$$\text{Perplexity}(W) = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} \tag{1}$$

where W is the sequence of words and N is the number of words in the sequence.

Results and Discussion

The results of our experiments are summarized in Table 1. We compare the perplexity scores of different smoothing techniques across unigram, bigram, and trigram models.

The results show that the perplexity increases drastically with higher-order n-grams when no smoothing is applied. This behavior is expected due to the sparsity problem, where many n-grams have zero counts, leading to extremely high perplexity values. This highlights the necessity for smoothing techniques to handle unseen n-grams effectively.

Smoothing Technique	**Unigram (N=1)**	**Bigram (N=2)**	**Trigram (N=3)**	**4-gram (N=4)**
None	756.29	4372.79	2734609.41	204017704.88
Laplace	601.90	577.92	2702.74	4666.67
Good-Turing	756.29	344743.28	82396877.92	982967514.05
Jelinek-Mercer	756.29	5757.05	3159456.67	218473542.77
Kneser-Ney	0.00005357	0.32	6149.32	12988649.82
Modified Kneser-Ney	765.61	5051.75	3195934.74	220785444.09

Table 1: Perplexity scores of different smoothing techniques across n-gram models.

Laplace smoothing demonstrates moderate improvements over the unsmoothed model by reducing perplexity for higher-order n-grams. While Laplace smoothing helps mitigate the sparsity problem, the perplexity still increases with the order of n-grams, indicating that this method, while beneficial, is not sufficient to fully address data sparsity on its own.

Good-Turing smoothing, on the other hand, performs poorly with higher-order n-grams, resulting in extremely high perplexity values. This suggests that while Good-Turing smoothing is theoretically robust, it struggles in practice due to its complexity and the requirement for reliable frequency estimates, especially for larger n-grams.

Jelinek-Mercer smoothing shows some improvement over the unsmoothed and Good-Turing models for bigrams and trigrams but still results in very high perplexity for higher-order n-grams. Although the linear interpolation used in Jelinek-Mercer smoothing helps to some extent, it is not sufficient to effectively handle the sparsity problem in higher-order models.

Kneser-Ney smoothing presents an unusual trend with very low perplexity for unigram and bigram models, suggesting an anomaly in the implementation or evaluation. For trigram and 4-gram models, however, the perplexity values spike drastically. This indicates that while Kneser-Ney smoothing is theoretically effective, practical implementation issues may affect its performance.

Modified Kneser-Ney smoothing performs similarly to Jelinek-Mercer smoothing for bigrams and trigrams but does not show significant improvement. The high perplexity values for higher-order n-grams suggest that the modifications, although theoretically beneficial, do not translate into substantial practical gains in this experiment. Several factors could explain why Modified Kneser-Ney did not perform as expected. First, the complexity of Modified Kneser-Ney smoothing means that any errors or oversights in the implementation could lead to suboptimal performance. Second, the technique introduces additional parameters that require careful tuning, and inadequate tuning of these parameters could result in worse performance. Third, the specific characteristics of the Brown corpus, such as its size, diversity, and domain-specific properties, might affect the performance of the smoothing techniques. The modifications in Kneser-Ney might not align well with these characteristics. Lastly, perplexity as an evaluation metric is sensitive to how well the model handles rare events. If the modifications in Modified Kneser-Ney do not significantly improve the handling of rare n-grams compared to the original, the perplexity scores might not reflect an enhancement.

Conclusion

This study highlights the evolution and comparative performance of smoothing techniques in n-gram models. The results highlight the challenges of higher-order n-gram models and the limitations of various smoothing techniques. Laplace smoothing provides moderate improvements over unsmoothed models, while Good-Turing and Jelinek-Mercer smoothing show significant issues with higher-order n-grams. The results for Kneser-Ney smoothing suggest potential implementation or evaluation issues, and Modified Kneser-Ney smoothing, despite its theoretical improvements, does not show significant practical benefits over other techniques in this experiment. Future research should focus on refining these techniques and exploring alternative methods to address data sparsity in n-gram models.

References

[1] Harold Jeffreys, *An invariant form for the prior probability in estimation problems.* Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 1946.

[2] I. J. Good, *The Population Frequencies of Species and the Estimation of Population Parameters.* Biometrika, 1953.

[3] F. Jelinek, R. L. Mercer, *Interpolated estimation of Markov source parameters from sparse data.* In Proceedings of the Workshop on Pattern Recognition in Practice, 1980.

[4] Reinhard Kneser, Hermann Ney, *Improved backing-off for m-gram language modeling.* In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1995.

[5] Stanley F. Chen, Joshua Goodman, *An empirical study of smoothing techniques for language modeling.* In Proceedings of the 34th annual meeting on Association for Computational Linguistics, 1996.