Introduction

Using the Wikitext-2 and PTB datasets, the goal of this study was to develop a 2-gram language model for text prediction. The project's goal was to evaluate a simple language model's performance and identify any potential drawbacks. We performed data preprocessing, trained the models, and evaluated their performance using the perplexity metric. A back-off strategy was also used to enhance the language models' quality, and the perplexity of the models with and without back-off was contrasted.

Preprocessing of Data

Building language models requires a critical step called data preparation. We created a preprocessing function that lowercases all characters and tokenizes the incoming text into words. Separate datasets for training, validation, and testing were produced as a result of the preprocessing of the text for the Wikitext-2 and PTB datasets. This process guarantees that our model is trained and assessed on correct and dependable data, which is essential for producing accurate results.

Language Model in 2 Grams

We created a 2-gram language model (M1) that serves as a foundation for both the Wikitext-2 and PTB datasets. The 2-gram model is a straightforward and computationally cheap language model that takes into account the most recent word while predicting the following word. The model was put into practice by counting the occurrences of each bigram in the training data using Python's defaultdict and Counter classes. The model was then put to the test using the corresponding validation and test datasets, with perplexity serving as the evaluation metric.

Model Assessment

We used the perplexity metric, which assesses how well a model predicts a sample, to assess the performance of our 2-gram language models (M1) without back-off. Better performance is indicated by lower perplexity values. The baseline models yielded the following perplexity findings:

Wikitext-2 (without back-off):

- Train Perplexity: 95.33
- Valid Perplexity: 19.79
- Test Perplexity: 18.67

PTB (without back-off):

- Train Perplexity: 72.13
- Valid Perplexity: 25.88
- Test Perplexity: 28.12

Back-off implementation

We used a back-off strategy to potentially enhance the performance of our 2-gram language models (creating M2). By giving a weighted probability to the unigram probability of the second word in the bigram, this method can handle unseen bigrams. In our studies, we applied a smoothing parameter alpha of 1. The models with back-off (M2) produced the following perplexity results:

Wikitext-2 (with back-off):

- Train Perplexity: 77.88
- Valid Perplexity: 684.63
- Test Perplexity: 786.89

PTB (with back-off):

- Train Perplexity: 55.69
- Valid Perplexity: 134.92
- Test Perplexity: 122.78

Discussion

The outcomes demonstrate that the back-off strategy produced a range of outcomes. Perplexity increased for the Wikitext-2 dataset, indicating worse performance. Perplexity increased for the validation and test sets for the PTB dataset, which at first glance seems to indicate better results. On closer inspection, the perplexity difference between the models with and without back-off is quite minor, and it might not really be an improvement.

The restricted context in the PTB dataset, as well as the higher perplexity values in the Wikitext-2 dataset, could be some of the contributing factors.

The increased perplexity values in the Wikitext-2 dataset and the modest improvement in the PTB dataset could also be attributed to a variety of variables. These elements consist of:

1. **Limited context**: When predicting the next word, the 2-gram model only takes the most recent word into account. When dealing with long-term dependencies or intricate phrase structures, this restricted context can lead to predictions that are incorrect. Higher-order n-gram models or more sophisticated models like RNNs, LSTMs, or transformers may perform better since they are better at capturing longer-term context.

2. **Dataset complexity**: The Wikitext-2 dataset has a wider vocabulary and more varied sentence patterns than the PTB dataset, making it more complicated. Higher perplexity values may result from the 2-gram model's inability to fully represent the Wikitext-2 dataset's complexity. Performance on complicated datasets might be enhanced by experimenting with more sophisticated models.

3. **Suboptimal smoothing parameter**: In the back-off approach, the alpha parameter controls the weight of the probability of a single bigram in determining the likelihood of a single unigram. The selection of alpha could not be the best one for the datasets available, resulting in subpar performance. The back-off technique may work better if the smoothing parameter is chosen in a more methodical manner, such as by utilizing grid search or cross-validation.

4. **Model drawbacks**: The back-off method and the 2-gram model are straightforward methods for modeling language. On both datasets, more sophisticated methods like interpolation, Kneser-Ney smoothing, or neural language models might do better.

5. **Tokenization technique**: The performance of a language model can be considerably impacted by the tokenization technique employed in the preprocessing stage. Because we solely took word-level tokenization into account in our approach, we may not have adequately captured punctuation marks or contractions. Better performance might be attained by using subword tokenization or better tokenization techniques.

   For both the Wikitext-2 and PTB datasets, addressing these issues by experimenting with higher-order n-gram models, putting more sophisticated language models into practice, and enhancing tokenization and preprocessing techniques may result in better performance and more trustworthy findings.

References

1. Aouragh, Si, Yousfi, Abdellah, & Laaroussi, Saida. (2021). A new estimate of the n-gram language model.
2. Lecouteux, Benjamin & Rubino, Raphaël & Linarès, Georges. (2010). Improving back-off models with bag of words and hollow-grams. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010. 2418-2421. 10.21437/Interspeech.2010-524.
3. Popel, Martin & Mareček, David. (2010). Perplexity of n-Gram and Dependency Language Models. 6231. 173-180. 10.1007/978-3-642-15760-8_23.