INLP Assignment-2 - Analysis

Name: Aaryan Kashyap
Roll Number: 2023114006

Perplexity Scores:

Perplexity Results for **FFNN** on **pride_and_prejudice**
Average Training Perplexity: 427.4428
Average Test Perplexity: 1200.8770

Perplexity Results for **FFNN** on **ulysses**
Average Training Perplexity: 1848.9583
Average Test Perplexity: 1992.5259

Perplexity Results for **RNN** on **pride_and_prejudice**
Average Training Perplexity: 255.7799
Average Test Perplexity: 1316.8406

Perplexity Results for **RNN** on **ulysses**
Average Training Perplexity: 649.1392
Average Test Perplexity: 1952.1131

Perplexity Results for **LSTM** on **pride_and_prejudice**
Average Training Perplexity: 166.3927
Average Test Perplexity: 1108.9579

Perplexity Results for **LSTM** on **ulysses**
Average Training Perplexity: 764.0636
Average Test Perplexity: 1845.8647

iNLP Assignment-1

Model: laplace
Corpus: Pride and Prejudice
Type: train
Average Perplexity: 512.715

Model: laplace
Corpus: Ulysses
Type: train
Average Perplexity: 712.492

Model: laplace
Corpus: Pride and Prejudice
Type: test
Average Perplexity: 441.497

Model: laplace
Corpus: Ulysses
Type: test
Average Perplexity: 634.293

Model: goodturing
Corpus: Pride and Prejudice
Type: train
Average Perplexity: 932.031

Model: goodturing
Corpus: Ulysses
Type: train
Average Perplexity: 893.967

Model: goodturing
Corpus: Pride and Prejudice
Type: test
Average Perplexity: 596.055

Model: goodturing
Corpus: Ulysses
Type: test
Average Perplexity: 799.117
Perplexity scores:

Corpus	Dataset	Laplace	Good-turing	FFNN	RNN	LSTM
Pride and Prejudice	Test	441.497	596.055	1200.8770	1316.8406	1108.9579
	Train	512.715	932.031	427.4428	255.7799	764.0636
Ulysses	Test	634.293	799.117	1992.5259	1952.1131	1108.9579

	Train	712.492	893.967	1848.9583	649.1392	764.0636

Comparison of Perplexity Scores

The perplexity scores from Assignment 2 (Neural Language Models) generally outperform those of Assignment 1 (Statistical Language Models) in terms of training perplexity, particularly for "Pride and Prejudice". However, the test perplexity scores for neural models in Assignment 2 are notably higher, suggesting challenges in generalization. Here's a detailed comparison:

1. **Training Perplexity:**
    - Neural models (FFNN, RNN, LSTM) show significantly lower perplexity on the training data compared to statistical models (Laplace, Good-Turing).
    - LSTM performs the best among the neural models, while Laplace smoothing performs the best among the statistical models.
2. **Test Perplexity:**
    - Test perplexity scores for neural models are higher than their training perplexity, indicating potential overfitting to the training data.
    - Statistical models exhibit smaller gaps between training and test perplexities, suggesting better generalization.
    - Among the statistical models, Laplace smoothing performs better than Good-Turing smoothing.

Ranking of Models

The models are ranked based on their perplexity scores (lower is better):

Training Perplexity Ranking

1. **LSTM on "Pride and Prejudice"** (166.39)
2. **LSTM on "Ulysses"** (764.06)
3. **RNN on "Pride and Prejudice"** (255.77)
4. **FFNN on "Pride and Prejudice"** (427.44)
5. **Laplace on "Pride and Prejudice"** (512.72)
6. **Good-Turing on "Ulysses"** (893.97)
7. **FFNN on "Ulysses"** (1848.96)

Test Perplexity Ranking

1. **LSTM on "Pride and Prejudice"** (1108.95)
2. **Laplace on "Pride and Prejudice"** (441.50)
3. **Good-Turing on "Pride and Prejudice"** (596.05)
4. **Laplace on "Ulysses"** (634.29)
5. **Good-Turing on "Ulysses"** (799.11)

6. **RNN on "Pride and Prejudice"** (1316.84)
7. **FFNN on "Pride and Prejudice"** (1200.87)

Detailed Analysis

1. Neural vs. Statistical Models

- **Neural Models (FFNN, RNN, LSTM):**
    - Neural models tend to overfit, evidenced by their high test perplexities despite low training perplexities.
    - Among neural models, LSTMs consistently perform the best, owing to their ability to capture long-term dependencies.
    - RNNs and FFNNs show limitations in handling the complexity of "Ulysses," possibly due to its higher vocabulary size and more complex sentence structures.
- **Statistical Models (Laplace, Good-Turing):**
    - Statistical models generalize better on test data as their test perplexities are closer to their training perplexities.
    - Good-Turing smoothing struggles more than Laplace smoothing, likely due to its sensitivity to unseen events.

2. Effect of Corpus Complexity

- The "Ulysses" corpus consistently results in higher perplexities across all models, reflecting its more complex sentence structures and vocabulary. Neural models particularly struggle with "Ulysses," showing higher perplexity gaps between training and test data.

3. Reasons for Variations

- **Model Complexity:** Neural models are more complex and can capture intricate patterns in data, but this complexity also leads to overfitting, especially when training data is limited.
- **Corpus Size and Diversity:** The larger vocabulary and higher complexity of "Ulysses" pose challenges for all models. Statistical models, which rely on smoothing techniques, can mitigate the effects of sparse data but cannot match the expressiveness of neural models.
- **Long-Term Dependencies:** LSTMs outperform other models because they address the vanishing gradient problem, making them better suited for capturing long-range dependencies in text.

4. Practical Implications

- For applications where training data is limited, statistical models like Laplace smoothing are more robust due to their generalization abilities.

- Neural models like LSTMs require more data and careful regularization to avoid overfitting, but they are superior in capturing complex relationships when properly tuned.

Conclusion

- **Best Performing Model:** LSTM on "Pride and Prejudice" achieves the lowest overall perplexity.
- **Most Generalizable Model:** Laplace smoothing exhibits the smallest training-test perplexity gap, demonstrating robustness.
- The choice of model should depend on the task requirements: LSTMs for tasks needing long-range context understanding and statistical models for scenarios with limited training data.

Which model performs better for longer sentences? Why?

Neural models, such as FFNNs, RNNs, and LSTMs, outperform statistical models (e.g., Laplace and Good-Turing smoothing) for longer sentences. This is because statistical models calculate perplexity as the inverse of the product of token probabilities. As statistical models generally assign less accurate probabilities to tokens, particularly for rare or unseen n-grams, the product of probabilities becomes very small for longer sentences. This results in an excessively high perplexity score. Neural models, on the other hand, capture sequential dependencies and patterns better, leading to more reliable probabilities, even for longer sequences.

How does the choice of n-gram size affect the performance of the FFNN model?

Smaller n-gram sizes generally result in better perplexity scores for FFNNs. This is because as the value of nn increases, the vocabulary size for n-grams grows exponentially (VnV^n), which negatively impacts the model's ability to assign accurate probabilities. Higher n-grams also lead to more sparse data and unseen combinations, further degrading performance. In contrast, smaller n-gram sizes (e.g., bigrams or trigrams) reduce the size of VnV^n, making the probability calculations more robust and less dependent on rare occurrences, thereby improving perplexity.

Click here to see all the generated files