NLP HW2

Chi-yeh Chen
chiyehc

Friday 25th October, 2024

4. Task2 Written

4.1 ngram counts: Train the n-gram language model on the *data/bbc/business.txt* dataset for n = 2 and n = 3. Then do the same for *data/bbc/sports.txt* dataset.

1. How many unique 2-grams are present in the business dataset?

Solution: 83819

2. How many unique 3-grams are present in the business dataset?

Solution: 141221

3. How many unique 2-grams are present in the sports dataset?

Solution: 77398

4. How many unique 3-grams are present in the sports dataset?

Solution: 135645

5. How many possible 2- and 3-grams could there be, given the same vocabulary? How do the empirical counts given above compare to the number of possible 2- and 3-grams?

Solution: For the business dataset, V is equal to 19990. Hence, the number of possible 2-grams should be 19990^2, and the number of possible 3-grams should be 19990^3.
For the sport dataset, V is equal to 9611. Hence, the number of possible 2-grams should be 9611^2, and the number of possible 3-grams should be 9611^3.
We can see that the empirical counts are much less than the possible counts. It is because of data sparsity.

4.2 Song Attribution: Train tri-gram (n=3, smoothing= 0.1) language models on collections of song lyrics from three popular artists (‘data/lyrics/‘) and use the model to score a new unattributed song.

1. What are the perplexity scores of the test lyrics against each of the language models? (a) Taylor Swift (b) Green Day (c) Ed Sheeran:

Solution:
(a) Taylor Swift PPL with tri-gram: 138.00663307990817
(b) Green Day PPL with tri-gram: 522.5401188730924
(c) Ed Sheeran PPL with tri-gram: 521.2574891234094

2. Who is most likely to be the lyricist?

Solution:
Taylor Swift

4.3 Introduction to Decoding and Text Generation:

It is said, Ed Sheeran will have a new album next year. Whether it is true or not, let us try to predict some lyrics that might show up. To predict the content of the tracks based on their titles, you plan to use an RNN language model trained on his lyrics (stored in *ed_sheeran.txt*). You are particularly interested in the following three tracks:

- **s1:** Yellow
- **s2:** Fell the Fall
- **s3:** Down Bad

1. For each of these track titles (s1 to s3), list the top five word candidates predicted to follow each sequence. Ensure you exclude special tokens added during training to indicate the end-of-sentence and start-of-sentence. You may need to modify or print specific details from the **generate_sentence** function to achieve this.

Solution:

(a) **s1**: Yellow pages `</s>` `<s>` i m
(b) **s2**: Fell the Fall in the with you`</s>`
(c) **s3**: Down Bad fruit`</s>` `<s>` i m

2. Report any one of the generated sentences here. Which generation mode do you think is better and why?

Solution: I think the max mode is better since it is more straight and predictable.

4.4 Comparison to a GPT:

1. What is the perplexity of your LM models (n-gram and RNN)?

Solution:

(a) **n-gram**: 179.0901520291377
(b) **RNN**: 1.0895694571618804

2. What is the perplexity of the GPT-2 model?

Solution: 50.644840240478516

3. How might you reason about the differences in perplexity between these three models? Think about the parameters, size of the vocabulary, training data, etc. used to build your language models, compared to that of GPT-2, and how might this impact their respective performance?

Solution:

- N-gram's high perplexity reflects its limitation in capturing long-term dependencies and struggles with data sparsity.

- RNN's low perplexity might suggest it has overfitted the training data or is performing very well on a simple dataset.

- GPT-2's perplexity, while still low, might reflect that it is handling a broader context but isn't as fine-tuned for this specific dataset as the RNN.

4. What are some of the trade-offs between using a simpler model like your language models versus a more complex model like GPT-2, and how might these trade-offs affect their performance on different tasks?

Solution:

- N-grams may work well for smaller tasks with limited vocabulary and short-range dependencies, but they will likely perform poorly on complex tasks like natural language generation or translation.

- RNNs are more suitable for tasks that require understanding sequential data, such as language modeling, but may struggle with long-term dependencies.

- GPT-2 excels in tasks requiring a deep understanding of context, such as question answering or creative writing, due to its large-scale training and the ability to handle long-range dependencies.