CSCI159 - Natural Language Processing
Pine, Anjali
09/17/24

1. **What is the best lambda smoothing parameter?**

Lambda	**Dev Perplexity**
0.1	549.9962029796966
0.001	335.93474790214407
0.0001	301.62200338284356
0.00001	362.11928537367163
0.000001	501.4428797666049
0.0000001	724.4364239600482
0.00000001	1053.7374491687344
0.000000001	1533.869950770377

Lambda	**Test Perplexity**
0.1	532.5822728085109
0.01	325.06184687015264
0.001	291.81264200311375
0.0001	349.50179406872223
0.00001	481.64589411785875
0.000001	691.7462639816223
0.0000001	1053.7374491687344
0.00000001	1533.869950770377

a. It seems that the best lambda is 0.001 for both dev and testing with a perplexity of 292. If I hadn't used the development set to select the lambda and simply picked a lambda arbitrarily, the performance difference would have been around 30. By using the dev set for lambda selection, I was able to minimize perplexity on unseen data, as the dev set helped identify the most appropriate balance between higher-order and lower-order n-grams. The optimal lambda on the dev set was close to the optimal lambda on the test set, which suggests that the dev set was representative of the overall data distribution. The results highlight the importance of tuning hyperparameters using a dev set rather than directly on the test set, as this prevents overfitting and leads to more generalizable models. While the lambda tuning improved results, some smoothing methods like the discount model still outperformed lambda interpolation, particularly in handling unseen n-grams.

2. **What is the best discount?**

Discount	**Dev Perplexity**
0.99	46.397554543213744
0.9	39.35456845854172
0.75	36.542048020172196
0.5	34.24655988341142
0.25	32.787759796040405
0.1	32.09666693265303

Discount	**Test Perplexity**
0.99	46.51017942873482
0.9	39.429284284448755
0.75	36.6061883865512
0.5	34.30518827499347
0.25	32.84434429609141
0.1	32.15263029240035

The best discount was 0.1 for both development and testing. Had we not used the development set to tune the discount parameter, we might have chosen a larger discount value arbitrarily, such as 0.5 or 0.9. This would have resulted in significantly higher perplexities on both the development and test sets. This shows that careful tuning on the development set was crucial for identifying the most effective discount value. The performance improvement is consistent across both the development and test sets, with the lowest discount value (0.1) providing the best results. This suggests that giving more weight to lower-order n-grams helps capture the structure of the data better. The minimal difference in perplexity between the development and test sets indicates that the discount model generalizes may be overfitting or is smoothing incorrectly.

3. **Performance**
   Which model is better? Provide some quantitative and/or qualitative arguments (including data or examples) of which approach is better. Make sure to clearly explain your evaluation approach and your arguments.
   - The discount model performed significantly better, as evidenced by the lower average perplexities compared to the lambda model. This is because the discount model more effectively reallocates probability mass, especially to rare and unseen events, by reducing the probability of frequent n-grams and redistributing it to less frequent ones. In contrast, the lambda model tends to assign too much probability mass to unseen events without considering their context or frequency, which leads to suboptimal performance.
4. **Wrap-up**
   Very briefly answer the following questions: how long did you spend on this assignment? What was the most fun part? least fun part? how would you improve it if I had to give it again?
   - Around 15-20 hours were spent on this assignment. The most fun part was getting to work in a partnership and learn from each other. It was also a cool experience to brush up on our Java knowledge and have concepts come back to us that we'd first learned in CS 62. The least fun part was debugging the code that we thought had sound logic while still getting wrong perplexities. To improve this process, we would recommend having more mentor sessions available as well as more cohesive instructions. I think a more chronological/procedural writeup would've been more helpful in following along with what we were supposed to do.

Ethics

5. Of the six scenarios, which do you see as the most problematic? Give a couple of sentences justifying your answer.

- I think the "Tyranny of AI Design" is the most problematic scenario. Since AIs are trained by biased data, the more we use AI, the more bias we feed into training. This potentially will create even larger disparities in an already segregated society, denying marginalized people of job opportunities, mortgage loans, health care, among others.

6. What is one scenario that is also concerning that is not listed here? It can be NLP-specific or more broadly to AI

- Related to the Deepfake example, AI could also impersonate a person’s voice and writing style if trained enough. This could lead to scams and fraud, where, for example, scammers can use your voice to ask for ransom from someone you know, or giving written consent to something you yourself did not agree to.