CS159 Assignment 2B

Nolan McCafferty and Daniel Rosenbaum

February 15, 2019

Evaluation

We randomly shuffled the sentences and split it into 90,000 sentences for training, 10,000 as our development and another 10,000 for testing. We trained on just the training data to get the development set perplexity and then trained on the training set + development set to get the testing perplexity.

1. What is the best lambda smoothing parameter?
   $\lambda = 0.001$

	perplexity (*ppl*)	
lambda (λ)	Development Set	Test Set
1.0	1073.44	1101.3
0.1	440.91	449.24
0.01	265.99	268.45
0.001	236.88	236.72
0.0001	282.73	279.56
0.00001	389.37	381.13
0.000001	559.35	542.57
0.0000001	808.93	777.88

2. What is the best discount?
   $d = 0.5$

	perplexity (*ppl*)	
discount (D)	Development Set	Test Set
0.99	182.26	183.69
0.9	156.40	157.69
0.75	149.29	150.49
0.5	149.13	150.16
0.25	159.46	160.19
0.1	180.78	181.00

3. Performance

   We found the Discount model to be the better language model. We can see this based on the perplexity tables above, as in the best case it calculates a much lower perplexity and in the general sense has a much lower overall perplexity as well (the worst case for the Discount, $d = 0.99$, performed better than the best case for the Lambda model $\lambda = 0.001$). The Discount model is also a lot more consistent as the difference between the worst case and best case is 33.13, whereas for the Lambda model this difference is 864.58. This shows that it is much more important to get your λ value correct as it has a large impact on the model's behavior, but with the Discount it can perform better with much less tweaking of the discount (d) value. We consider perplexity to be a fair evaluation of these models since all test data are actual sentences and thus should be recognized as such. To really show this performance is better, we also tested these models on made-up bad sentences (expecting to get much higher perplexities). Using the $\lambda = 0.001$ Lambda model we got $ppl = 1192.63$ and using the $d = 0.5$ Discount model we got $ppl = 820.84$. Here although Lambda produced a higher perplexity it is more within the range of results and thus for these incorrect sentences. We think that our Discount model also is more likely to categorize them as bad sentences since it produced much higher (where the magnitude increase was larger than Lambda model magnitude increase) perplexity in this case than when tested on actual sentences.

   We also implemented sentence generation as a more qualitative measure of performance. We used two different generation techniques, greedy and sampling.

   (a) Greedy

   The first sentence generator uses a greedy model. This means that the next word in the sentence will be the word with the highest bigram probability with the word before it. Thus, the greedy sentence is unique. To start the sentence, we use $< s >$ as the first word and generate from there. Interestingly, the greedy sentence is the same for all models. The sentence that we got is: `the united states .`

   Note: We noticed that our shuffled data must have put a lot of sentences about Switzerland into our Development and Test set, and got different "greedy" sentences when using these to train.

   (b) Probability Based Sampling

   The second sentence generator uses a model that samples from a probability distribution of the second words in a bigram. For this model, words that have a higher bigram probability will be more likely to occur but the choice is still random. Thus, we get very different sentences every time this generator runs. Again, we begin the sentence with $< s >$ and go from there. We used ten words as the maximum cutoff length for a sentence. So either the sentence ends with a period or the seventh word. Here are five generated sentences

for our most relevant models:

Lambda = 1

```
his wistert chopstick tab citi escapes kirkham
the blasphemous gaia identifying transferred satya gharbi
our gennesaret het flatten alban tensile nathra
he waterhouse high-school capes tikal communism pete
bmi vhs known-plaintext isigny-sur-mer nitrogen generalised kaito
```

Lambda = .001

```
this new orleans write once said the
a large nests schirottke hijackers wednesday checkuser of christian church
an athenian ecclesiology axioms for her daughter
in hershey company , it is an aircraft joel davis
most important as being announced the region
```

Discount = .5

```
his family feature that guitar and yemen growth take pictures
later , in latvia with the represented by tennessee .
the custard the school for steaks langston university of wcw
bioluminescence in the distinguished from a simple is divided
as seen is a very special parameter μ day ,
```

It is interesting to see how much better both of our best models are compared to the $\lambda = 1.0$ model. We also think that the best Discount model ($d = 0.05$) was able to produce better looking sentences than the best Lambda model ($\lambda = 0.001$). Although this is not the best metric of performance it is very fun to calculate.

4. Wrap-up
   We spent around 8 hours on this assignment, and most of that time was spent debugging our discount model. The least fun part was debugging and not knowing if our results were correct. On this note, a nice improvement to the assignment could be giving a range of values that the perplexity should be within. As it was frustrating to see our model work on smaller, hand-tested corpora but then give various results on the `sentences` file and not know which one was better. The most fun part was the sentence generation and seeing what the different models came up with. We also liked implementing an abstract class in Java since we had never done it before.