CSE 143 Assignment 2

Liam Xu, Jaisuraj Kaleeswaran, Vishwa Vijayasankar

May 2024

1 Programming: n-gram language modeling

Training Set

Model	Perplexity
Unigram	976.54
Bigram	77.07
Trigram	7.87

Dev Set

Model	Perplexity
Unigram	892.25
Bigram	28.29
Trigram	2.99

Test Set

Model	Perplexity
Unigram	896.49
Bigram	28.31
Trigram	2.99

The perplexities of the unigram model on the different data sets are the highest, while the bigram perplexities are much lower, and the trigram perplexities are the lowest. Also, the perplexities of the models on the debug file with HDTV . were $658, 63.7, 39.5$, respectively. This is pretty much as expected.

2 Programming: additive smoothing

$\alpha = 1$

Training Set

Model	Perplexity
Unigram	977.51
Bigram	1442.31
Trigram	6244.42

Dev Set

Model	Perplexity
Unigram	894.39
Bigram	1669.66
Trigram	9676.65

$\alpha = 0.5$

Training Set

Model	Perplexity
Unigram	976.80
Bigram	971.66
Trigram	3964.85

Dev Set

Model	Perplexity
Unigram	893.15
Bigram	1241.95
Trigram	7905.41

$\alpha = 0.1$

Training Set

Model	Perplexity
Unigram	976.55
Bigram	407.84
Trigram	1115.69

Dev Set

Model	Perplexity
Unigram	892.39
Bigram	701.73
Trigram	4899.49

Using additive smoothing resulted in worse perplexities than before, so the best hyperparameter would be $\alpha = 0$, which is the same as not using any smoothing. This might be because additive smoothing shifts too much of the probability mass away from commonly seen tokens, resulting in a less accurate model overall.

3 Programming: smoothing with linear interpolation

3.1

Training Set

Hyperparameters	Perplexity
$\lambda_1 = 0.1, \lambda_2 = 0.3, \lambda_3 = 0.6$	11.15
$\lambda_1 = 0.2, \lambda_2 = 0.2, \lambda_3 = 0.6$	11.53
$\lambda_1 = 0.1, \lambda_2 = 0.4, \lambda_3 = 0.5$	12.44
$\lambda_1 = 0.05, \lambda_2 = 0.4, \lambda_3 = 0.55$	11.57
$\lambda_1 = 0.01, \lambda_2 = 0.4, \lambda_3 = 0.59$	10.95
$\lambda_1 = 0.6, \lambda_2 = 0.3, \lambda_3 = 0.1$	38.33

Dev Set

Hyperparameters	Perplexity
$\lambda_1 = 0.1, \lambda_2 = 0.3, \lambda_3 = 0.6$	3.25
$\lambda_1 = 0.2, \lambda_2 = 0.2, \lambda_3 = 0.6$	3.33
$\lambda_1 = 0.1, \lambda_2 = 0.4, \lambda_3 = 0.5$	3.32
$\lambda_1 = 0.05, \lambda_2 = 0.4, \lambda_3 = 0.55$	3.24
$\lambda_1 = 0.01, \lambda_2 = 0.4, \lambda_3 = 0.59$	3.19
$\lambda_1 = 0.6, \lambda_2 = 0.3, \lambda_3 = 0.1$	4.57

3.2

Test Set

Hyperparameters	Perplexity
$\lambda_1 = 0.01, \lambda_2 = 0.4, \lambda_3 = 0.59$	3.19

3.3

If you used half the training data, the perplexity on unseen data would increase, because the model would have less data to learn from and become less capable of predicting the next word correctly.

3.4

If you converted all tokens that appeared less than 5 times to UNK, it would decrease the perplexity compared to if you converted tokens that appeared only once, since it would decrease the size of the vocabulary and force the model to focus on more common words, which would help it generalize to unseen data.