Natural Language Understanding Assignment 1

Udit Gupta
M.tech CDS
14454
udit2008.gupta@gmail.com

Task 1

The data has been divided into train, development and test sets with 60-20-20 splits. Tokens occurring fewer than or equal to 3 times have been replaced with a <unk> token to cater for Out-of-Vocabulary words. The dictionaries for unigram, bigram and trigram have been made from sentences (after prepending <s> and appending </s> tokens) using the train set. Any word in the test set that isn't present in the vocabulary is assigned <unk> token.

Metric used for comparison: Perplexity of the test set.

Language model: I first implemented the linearly interpolated trigram model and found optimal weights (.5, .3, .2) for trigram and (.75, .25) for bigram using the dev set. The perplexity on the brown corpus was around 170. Next, I implemented the Kneser-Ney algorithm using **interpolated Kneser-Ney smoothing for trigrams**. The discount value of 0.7 gave better results. This gave lower perplexity (around 130) than the linear interpolation method. Hence, I used Kneser-Ney model for final evaluation of subtasks S1, S2, S3, S4.

Subtasks	Perplexity	Vocab size
S1	128	12232
S2	144	14625
S3	162	18132
S4	95	21781

Observations: For S1, S2 and S3 the perplexity increases as the vocabulary size increases, but for S4 where we train on Brown and Gutenberg, and test on Gutenberg corpus (train/dev/test splits considered) the perplexity turns out to be lower even though the vocab size is larger. This might be because the books in Gutenberg corpus also capture some information from the text in Brown corpus. The converse isn't true (consider S3), i.e. testing on Brown corpus while training on Brown and Gutenberg corpora won't lead to lower perplexity since sequences in Brown corpus are less likely to have been generated from the sequences in Gutenberg corpus.

Conclusion: So, it is imperative that we understand that simply putting more data (words in this case) into training from any source wont reduce the perplexity. In order to reduce perplexity, we should properly choose the domain for training data and then try to obtain maximum amount of data from that domain.

Task 2

The procedure used to generate sentence is:

1. Start from [<s>, <s>]
2. Uniformly sample a token from the training data.
3. Calculate the (continuation) probability for that token as given by the Kneser-Ney interpolated model.
4. Repeat this process 7500 times, each time sampling a random token and calculating its probability.
5. The token which has the highest probability is chosen as the next word in the sequence.
6. All the above points are repeated 10 times to generate a 10 token sentence.

Generated sentence from Brown Corpus:

'and now the original sin to make some real money'