NLP Assignment 2 Report

Aryan GD Singh
2019459

Methodology

1. Preprocessing of data
2. Counting the counts of words and bigrams
3. Training language models
4. Testing the models

Preprocessing

1. Make sentences lowercase
2. Remove punctuation
3. Tokenize sentences using NLTK tokenizer

Counting

We store the counts of words and bigrams using defaultdict in Python, which behaves similarly to dictionary.

Storing counts of unigrams is done simply by looping through the words, and for creating bigrams the NLTK library is used.

Training the models

We use the bigram counts to calculate the probabilities for each bigram, following different methods for each type of model.

No smoothing: P(w2 | w1) = count(w1 w2) / count(w1)
Laplace smoothing: P(w2 | w1) = (count(w1 w2) + 1) / (count(w1) + V)
Add k smoothing: P(w2 | w1) = (count(w1 w2) + k) / (count(w1) + k*V)

k is taken as 0.2

Testing and results

We read the questions, options and answers from the validation file and start making predictions.

The word in options with the highest probability is selected.

Accuracy is calculated as the percentage of questions correctly answered.

All the 3 models perform the same and give slightly more than 50% accuracy, 53.25% to be exact. The smoothing does not affect performance as it preserves the order of the probabilities even after scaling.

Assumptions

apostrophes are not handled separately, so “s” is a valid token.
0 occurrence cases are handled by smart utilisation of dictionaries.