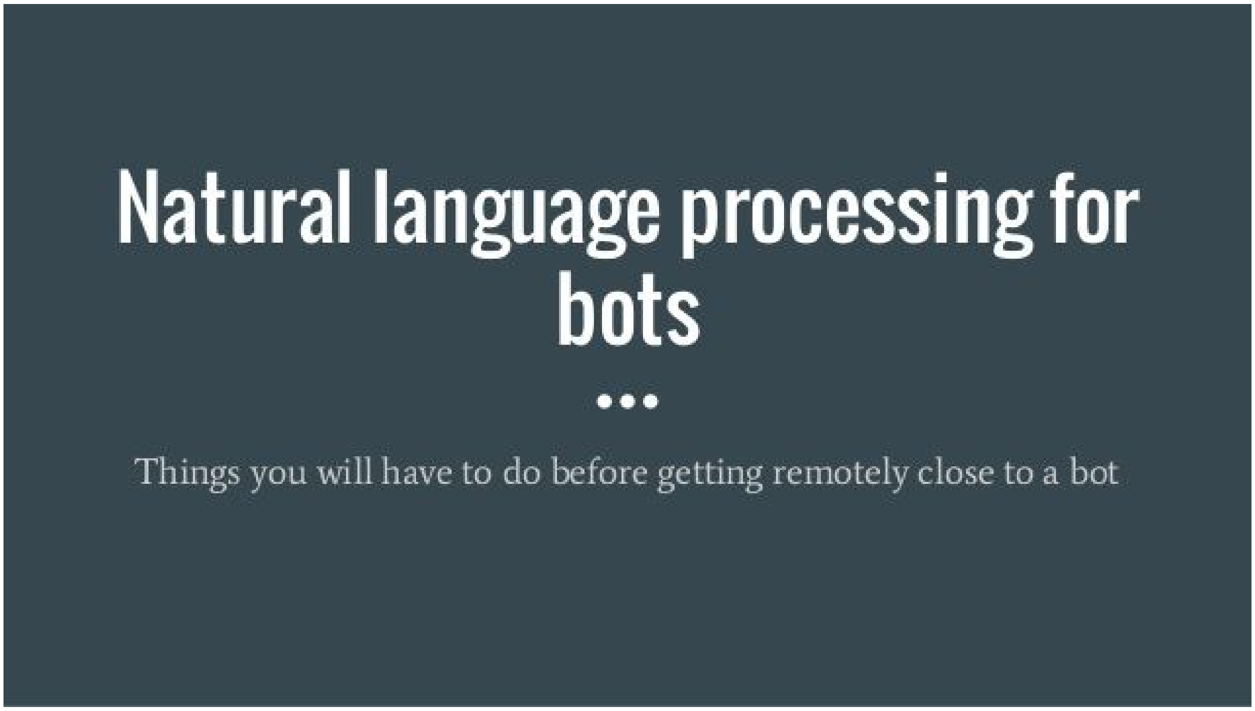

Natural language processing for bots

•••

Things you will have to do before getting remotely close to a bot

Twitter Bot

06.02.2018

Sudhanshu Kumar Singh
15EE10047
NLP Assignment-1

Overview

The objective of this assignment is to test out the language modeling techniques that we have discussed in the class. The primary task is to design a bot that can generate interesting tweets given an entity or topic. The bot will consult a language model for generating the tweets.

Goals

1. **Corpus Compilation:** The twitter data is in JSON format. So, a corpora has to be created out of the JSON files by concatenating the individual tweets. Remove the URLs if they are present in tweets.
2. **Build Language Models:** Use nltk language processing library to build n-gram language models.
3. **Experiment:** Use different smoothing techniques, Laplace, Good-Turing and KneserNey and report perplexity values using held-out test.

Dependencies

- Python 3 : programming language of the bot
- NLTK : used for language processing and model making
- Important python libraries : Pandas, json, math, regular expression, string
- Operating System : Windows 10

Abstract

- Created text file (Corpus) from all json files after processing text
- Applied n-gram language models on the corpus extracted
- Used 3 different smoothing techniques to the language models
- Calculated the entropy and perplexity values for each language model

Milestones

I. Corpus Making

I took all json files and concatenate them in one text file using for loop, which contains only the tweets and not other details of the user. I have also removed the non-english words from tweets using english language words in nltk library. I also removed the urls in the tweets using the regular expression. Using string library I filtered the punctuations from the text such as @, !, #, etc.

II. Train/ Test data

I splitted the corpus used in 95:5 for training and testing respectively and calculated the perplexity values on training and testing corpus.

III. N-gram Language model

Created Unigram, Bigram and Trigram models from the corpus created. For creating these models nltk library packages are used. Then conditional probabilities for each model is computed using nltk library packages. I have also padded the ngrams with <s> and </s> for the beginning and end of each sentence respectively.

IV. Smoothing Techniques

I have used three different smoothing techniques for each language model namely, Laplacian smoothing, Good Turing smoothing and Kneser-ney smoothing. I defined three functions for each smoothing technique and then used them while evaluating each model.

Perplexity values are calculated for each smoothing technique and found out to be different.

V. Evaluation methods

Entropy and perplexity values are calculated to evaluate the language model created. Log probability is also calculated by defining a function in python. Then i wrote functions for entropy and perplexity value calculation.