KENTUCKY STATE UNIVERSITY, COMPUTER SCIENCE DEPARTMENT

Building and comparing Parts Of Speech tagger models using Natural Language Processing Project Status Report

Anil Varma Bethalam
March 11, 2020

PROJECT DESCRIPTION

The Aim of the project is to research and build different Parts of Speech tagger models using the natural Language Tool Kit library in Python and compare the different pros of cons of each model. The different model are built using different algorithms and methods and are described as follows.

- Model 1: Lexicon and Rule-based POS Tagger A Lexicon tagger uses a simple statistical tagging algorithm. For each token, it assigns the most frequently assigned POS tag. Rule-based taggers first assign the tag using the lexicon method and then apply predefined rules. We use the Treebank corpus of NLTK to implement the Lexicon and rule based tagger.
  - Reading and understanding the tagged dataset
  - Exploratory analysis
  - Lexicon and rule-based models:
  - Creating and evaluating a lexicon POS tagger
  - Creating and evaluating a rule-based POS tagger
- Model 2: HMMs and Viterbi algorithm for POS tagging

  Markov processes are commonly used to model sequential data, such as text and speech. The Hidden Markov Model (HMM) is an extension to the Markov process which is used

to model phenomena where the states are hidden and they emit observations. In POS tagging, what you observe are the words in a sentence, while the POS tags themselves are hidden. Below are the steps that will be used for building the model.

  - Conduct exploratory analysis on the tagged corpus
  - Sample the data into 70:30 train-test sets
  - Train an HMM model using the tagged corpus:
    * Calculating the emission probabilities
    * Calculating the transition probabilities
  - Write the Viterbi algorithm to POS tag sequence of words (sentence)
  - Evaluate the model predictions against the ground truth

- Model 3: Building POS tagger using Long short-term memory (LSTM) and Bidirectional LSTM, these are recurrent neural networks architecture. Below are the steps for building the model. We will use a combination of the Tree-bank, brown and CONLL 2000 Chunking Corpus corpus available in NLTK library to implement the RNN models. Below are the steps that will be used for building the model.
  - Preprocess data
  - Vanilla RNN
  - Word Embeddings
  - LSTM
  - Bidirectional LSTM
  - Model Evaluation

PROJECT STATUS

This section states the current status of the project. The python coding has been done using Jupyter notebooks. The first model is built and the coding has been completed and the documentation for the first model is in process. The second model is still being researched and the coding has been started and is in process.