Donghyun Park & Junbum Kim

Prof. Sable

Natural Language Processing

POS Tagging

Project #2 Proposal: POS Tagging Using N-grams

Introduction

The idea for this project is to utilize various n-grams in identifying and tagging parts of speech given a training set and test set. While trying to implement a text categorization system, the authors tried using POS tags, and realized that different implementations of POS tagger result in vastly different accuracy and time complexity. Given the results, the authors wanted to explore more on how different implementations of the POS tagger will perform.

Background Research

Part-of-speech tagging can be done through various methods, such as HMM, dynamic programming, SVM, KNN, percentron and such. This project will most likely be done using the Viterbi algorithm and HMM, together which have been tested to achieve accuracies of approximately >90% (1). As for n-grams, simply put, it is the sequence of n number of consecutive words from a text. In context of POS tagging, it would mean that the tagger would be taking in n number of words as the input, and then processing them sequentially dependent on the prior to improve the estimation of the tag (2).

Detailed Description

When tagging parts-of-speech, there are various methodologies which all have different levels of accuracy and complexities. Amongst the various methods, there are also multiple ways to improve the metrics provided by the POS tagger, such as the implementation of n-grams. For example, using bi-grams or tri-grams in the POS tagging process will result in progressively better performance than using uni-grams as it can better understand “context” of the sentence it is analyzing. So, this project seeks to see how well the initial POS tagging performs, and compare it to the performances of n-gram-implemented taggers. The goal would be to tell whether the difference in performance would be significant enough for the increasing complexity caused by the various n-grams used in the process. The overall program will be composed of 2 separate programs, training and testing. We will initially take in a corpus, and split it into a training and testing set. Then the tags in the testing set will be stripped so that it becomes an untagged set. The training program will, naturally, take in a training file, and using HMM and n-gram smoothing, output a file with the weighted tags vector associated to the input file. The testing program will take in the weighted file and the testing set, and then perform the tagging process onto the testing set. The output of the test program will be the tagged version of the stripped test set. This output will be compared to the original unstripped version of the test set, and then the performance of the system will be measured using the contingency table.

Evaluation Method

Contingency table will be used to check the accuracy and recall of the system using different N for n-gram. We expect the accuracy of the program to be above 90%, and increase as the number N increases.

Demonstration Material

The demonstration of the program will include three parts. The first part will be a demo of the running program. After the demo, we will run a script to create a contingency table to compare the result obtained from the program with the correctly tagged document. Since part of speech tagging produces clearly visible results, we also plan on showing the tagged result and the original document.

Citation

1. <https://en.wikipedia.org/wiki/Part-of-speech_tagging>
2. <https://en.wikipedia.org/wiki/N-gram#Syntactic_n-grams>
3. <http://www.phontron.com/slides/nlp-programming-en-04-hmm.pdf>