Disclaimer

- The material provided in this document is not my original work and is a summary of some one else’s work(s).
- A simple Google search of the title of the document will direct you to the original source of the material.
- I do not guarantee the accuracy, completeness, timeliness, validity, non-omission, merchantability or fitness of the contents of this document for any particular purpose.
-

HMM Based POS Tagger

Semantic Processing System

Term Project

Presented by

Najeeb Khan

2013-6-17

Outline

- Introduction
- Hidden Markov Models
- Simulation Results
- Demonstration
- Conclusion

Introduction

- The process of classifying words into their parts-of-speech and labeling them accordingly is known as part-of-speech tagging
- Some words can represent more than one part of speech at different times e.g. ‘closed’
- POS is important step in many Language and Speech processing tasks

Introduction

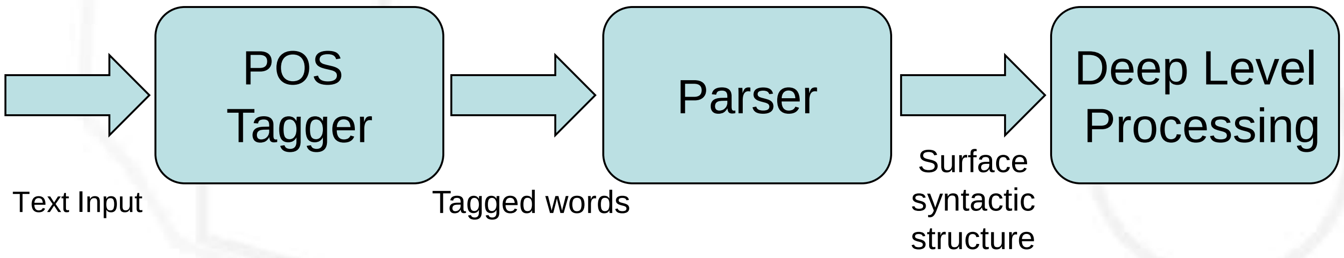
- High accuracy is required in POS tagging tasks, why?

Introduction

- High accuracy is required in POS tagging tasks, why?
- Language Processing

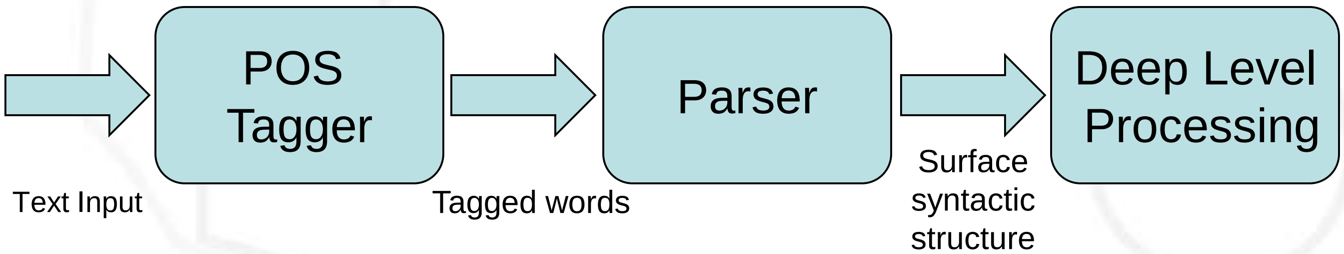
Introduction

- High accuracy is required in POS tagging tasks, why?
- Language Processing



Introduction

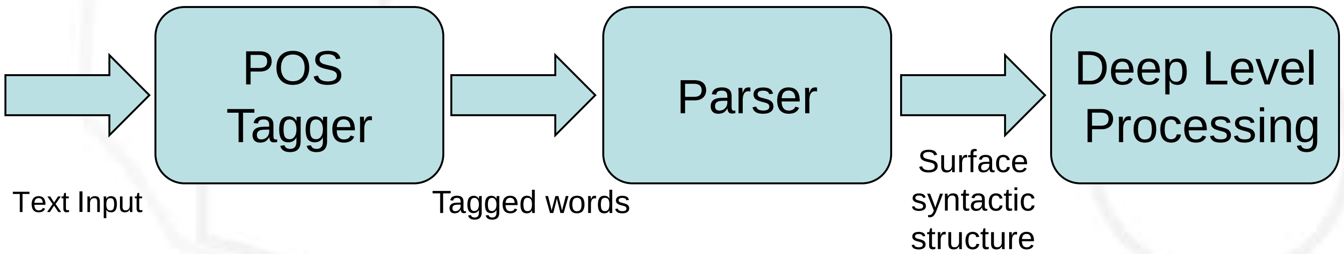
- High accuracy is required in POS tagging tasks, why?
- Language Processing



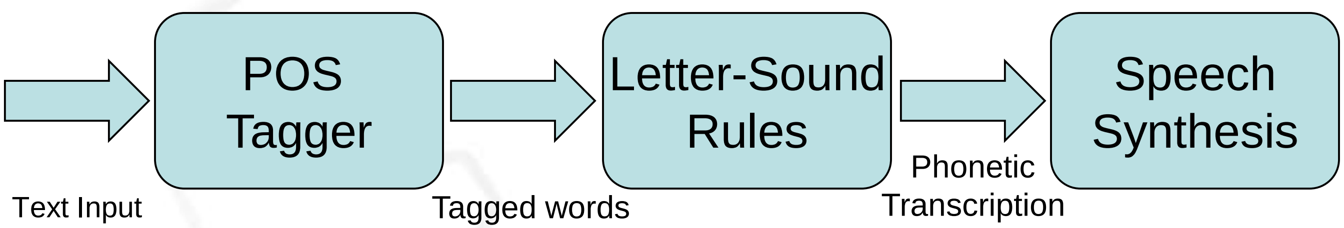
- Speech Processing

Introduction

- High accuracy is required in POS tagging tasks, why?
- Language Processing



- Speech Processing



Introduction

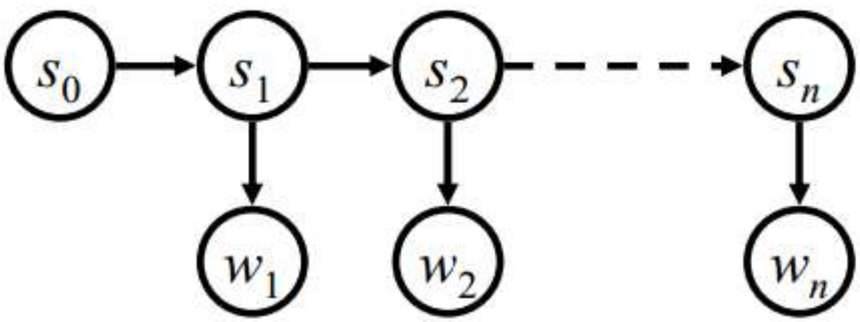
- The HMMs we have seen in the NLK so far i.e. REGULAR EXPRESSION, UNIGRAM, and BIGRAM have accuracies in the range of 80 to 90%
- In this project, I have used the Hidden Markov Model module available in NLTK for POS tagging
- The results of the HMM taggers and other taggers have been compared

Hidden Markov Models

- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words

Hidden Markov Models

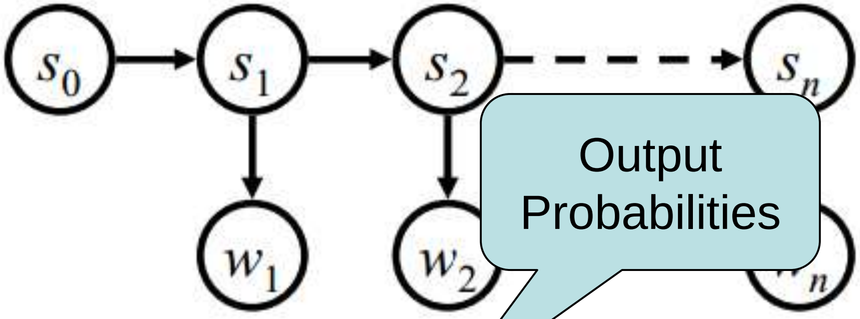
- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words



$$P(S \,|\, \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \,|\, \mathrm{s}_i)\,\mathrm{P}(\mathrm{s}_i \,|\, \mathrm{s}_{i-1})$$

Hidden Markov Models

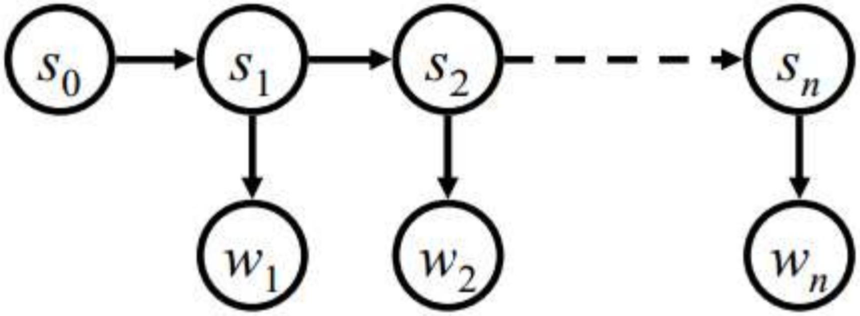
- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words



$$P(S \mid \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \mid \mathrm{s}_i)\,\mathrm{P}(\mathrm{s}_i \mid \mathrm{s}_{i-1})$$

Hidden Markov Models

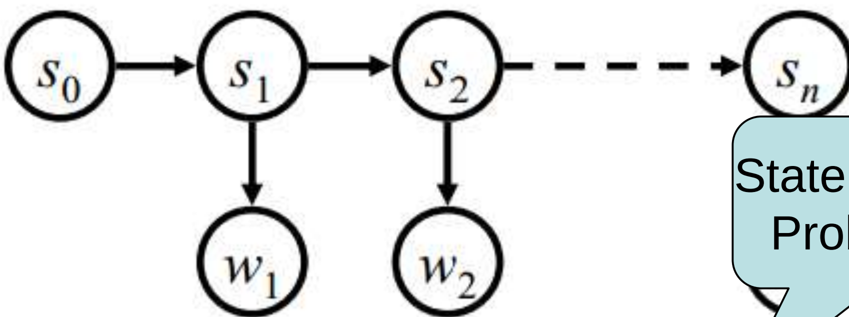
- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words



$$P(S \mid \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \mid \mathrm{s}_i)\,\mathrm{P}(\mathrm{s}_i \mid \mathrm{s}_{i-1})$$

Hidden Markov Models

- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words

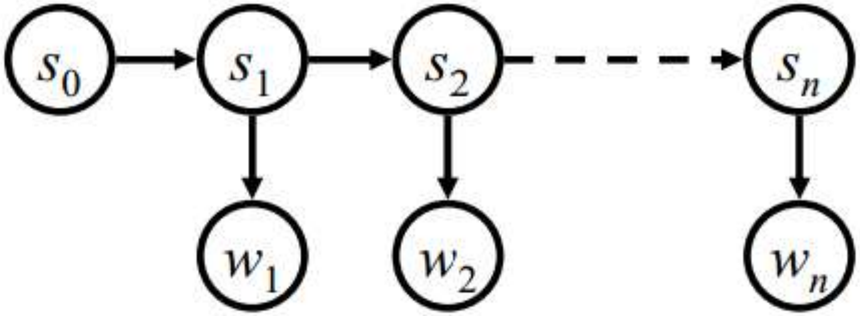


State Transition Probabilities

$$P(S \mid \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \mid \mathrm{s}_i)\, \mathrm{P}(\mathrm{s}_i \mid \mathrm{s}_{i-1})$$

Hidden Markov Models

- The HMM is a directed graph, with probability weighted edges where each vertex emits an output symbol when entered
- In POS tagging problem the states are the tags and the symbols are the words



$$P(S \mid \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \mid \mathrm{s}_i)\,\mathrm{P}(\mathrm{s}_i \mid \mathrm{s}_{i-1})$$

Hidden Markov Models

- There are two steps in using HMMs for POS tagging
  - Estimating the HMM parameters (transition and output probability distributions)
  - Finding the tag sequence $S=\{s_1 \ldots s_n\}$ which maximize the probability $P(S|W)$ given a word sequence

Estimating HMM Parameters

- There are many methods for estimating HMM parameters
- Maximum Likelihood Estimate

$$P(s_i \mid s_{i-1}) = \frac{count(s_i \text{ seen after } s_{i-1})}{count(s_{i-1})}$$

$$P(w_i \mid s_i) = \frac{count(w_i \text{ seen as } s_i)}{count(s_i)}$$

- MLE suffers from sparse data problem
- Some smoothing technique is required to deal with unseen words

Maximizing the Probability

- To find the POS for a sequence of words $W=\{w_1 \ldots w_n\}$, we now need to find the tag sequence $S=\{s_1 \ldots s_n\}$ which maximizes $P(S|W)$

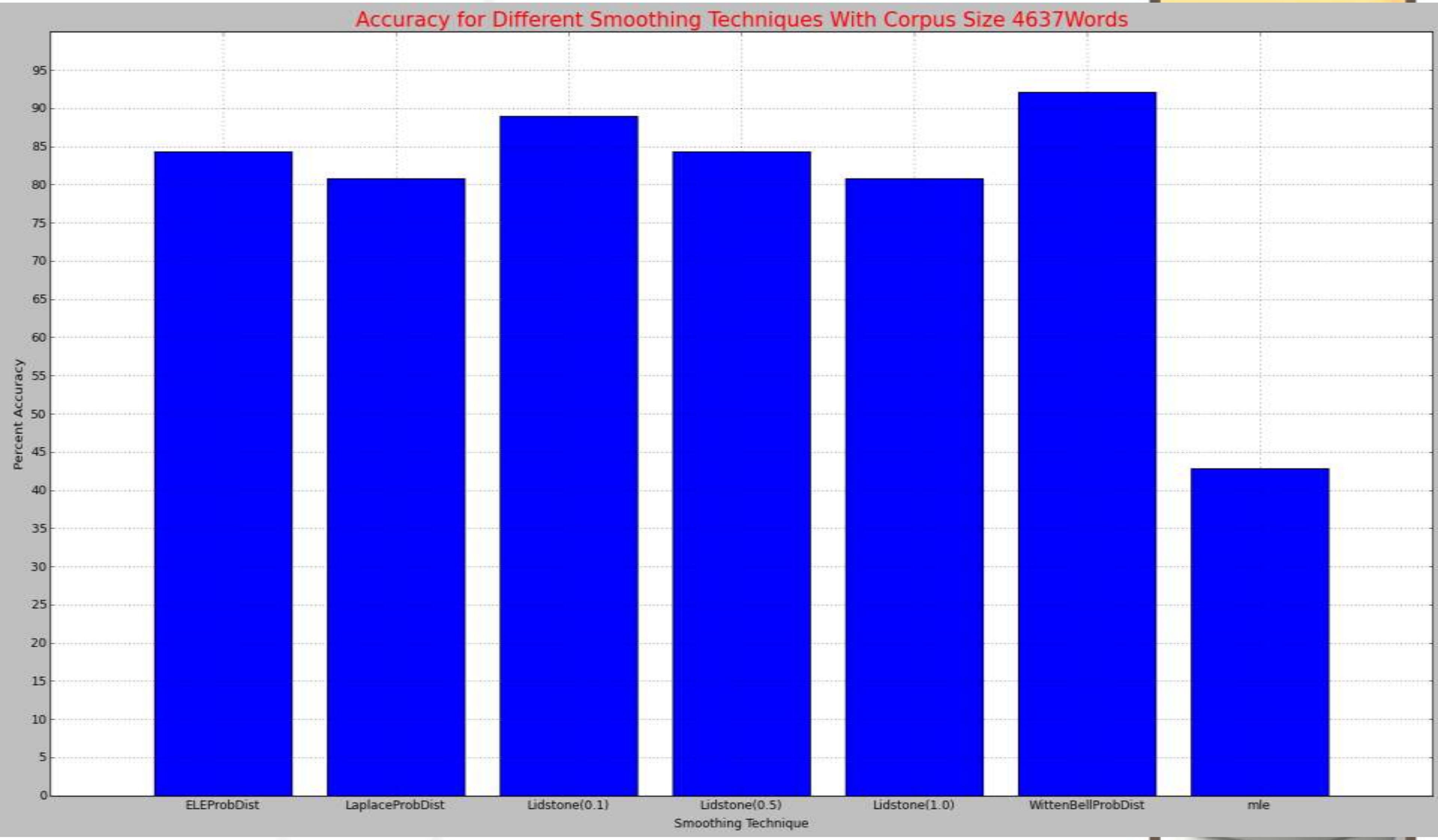
$$\mathrm{argMax}[P(S \mid \mathrm{W}) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{w}_i \mid \mathrm{s}_i)\,\mathrm{P}(\mathrm{s}_i \mid \mathrm{s}_{i-1})]$$

- Finding this maximum requires very high computational power
- Viterbi Algorithm is used to find out this maximum probability tag sequence efficiently

Simulation Results

- All the taggers presented were trained and tested on Brown corpus
- The testing data was separate and 10 10% of the training data
- The accuracy for different smoothing techniques was obtained for the HMM based tagger
- The Witten Bell smoothing technique yielded best accuracy

Simulation Results



Simulation Results