ECSE 526
Lab Report 2
Markov and Hidden Markov Models for Text

Yajiv Luckheenarain (260986423)

This project implements a Markov Model and a Hidden Markov Model for English Text. The first part of this experiment uses a Markov Model to generates sentences given a vocabulary, a unigram, a bigram, and a trigram joint probability distribution between words in the vocabulary. The second part of this experiment uses a Hidden Markov Model to correct Noisy sentences as inputs. This is done using the bigram probability distributions from the first part of this project as well as using the Viterbi algorithm and the Levenshtein distance between observed words and expected words to determine the de-noised input.

I. Introduction

Markov models, including Hidden Markov Models (HMMs), are crucial in linguistics for generating sentences and correcting noisy inputs. They rely on Markov processes, where future states depend on the current state. Markov models predict word occurrence based on preceding words, aiding in text prediction and autocomplete. HMMs, with hidden states, model linguistic phenomena like part-of-speech tags, error correction, word meaning disambiguation, and improving speech recognition and translation. These models are essential in linguistics, enabling sentence generation, noise correction, and unraveling language structures.

II. Sentence Generation Using a Markov Model

In sentence generation, a tailored Markov Model for English text uses data from 'vocab.txt' and probability distribution files 'unigram.txt,' 'bigram.txt,' and 'trigram.txt.' These files contain joint probabilities of word occurrence in a sentence, based on preceding words. Unigrams represent standalone word probabilities, while bigrams and trigrams capture conditional probabilities with one or two prior words.

The 'sample' function plays a vital role, employing stochastic selection based on probability distributions. It calculates cumulative exponential probabilities for words, selects a random threshold from a uniform distribution, and iteratively aggregates probabilities until surpassing the threshold, selecting the corresponding word for the sentence. This probabilistic sampling ensures coherent sentences that mimic authentic English text structures.

The following figure displays some generated sentences using this Markov Model.

```
<s> so long a time , Sir John did not know when I am sure it is not quite unconnected in
this manner , and his wife . </s>
<s> She is a very good style . </s>
<s> " </s>
<s> why cannot I speak , and you may be a much better for her , had been the effect of
time and strength of his eye , and water , and the going abroad in such affectionate
remembrance . </s>
<s> I can express . </s>
```

Fig. 1: Five Generated Sentences of the Markov Model

III. HMM and Sentence De-Noising

A Hidden Markov Model (HMM) with the Viterbi algorithm was used to determine the most likely sequence of states from an observed sequence of words in noisy text data. It utilized a tabulation method and Levenshtein distance as an emission probability metric for transitioning between states, followed by backtracking for denoising the observed word sequence. The model also incorporated known bigram relationships from the vocabulary for initialization. The Levenshtein distances were computed using a Python library [1].

The following figure shows some noisy sentences and their denoised equivalent

```
Noise input: I think hat twelve thousand pounds


Resolved Input: I think at twelve thousand pounds
Noise input: she haf heard them


Resolved Input: she had heard them
Noise input: She was ulreedy quit live


Resolved Input: She was already quite like
Noise input: John Knightly wasn't hard at work


Resolved Input: John Knightley was hard at work
Noise input: he said nit word by


Resolved Input: he said it would be
```

Fig. 2: Noisy and Denoised Sentences

References

[1] *Levenshtein*. Version 0.22.0. Python library. URL: https://pypi.org/project/python-Levenshtein/.