Word Sense Disambiguation Mini Project

I223E - Natural Language Processing(E)

Description

This programming project is about developing a method for Word Sense Disambiguation (WSD).

In this project. You will implement **one of the three Machine Learning methods** for WSD

- **Support Vector Machine (SVM)**
- **Naive Bayes**
- **Decision List**

You can choose to implement any of the three methods above. You can also use **Word2Vec** in your implementation. You can use **any Python library** to support your implementation, but the fundamental idea of the method must be implemented by you. The quality of your implementation is also part of the grade.

You are required to submit the source code and a report describing your method and implementation. The preferred programming language is **Python**.

It is important that you can explain your method and show some small demos using your code. Your method can contain many steps. Some scores might be given even if you cannot run all the steps in your method, but you can explain the method and provide code for some important steps.

Task

Given a sentence and a word, your method must determine the sense of this word (based on the sense inventory **WordNet**). Example input and output are as follows.

Input

Sentence: **['I', 'went', 'to', 'the', 'bank', 'to', 'deposit', 'money', '.']**
Word: **'bank'**

Output

The sense name (in WordNet) of the word 'bank': **'savings_bank.n.02'**

Submission

Your submission must include:

- The source code of your project. This must be **Python Notebook** (***.ipynb**) files
- The report describing your method and implementation in PDF (*.pdf)

Important note: Your code needs to provide a **predict** function as described below.

```
def predict(sentence: List[str], word: str)
```

Given a *sentence* (this sentence is tokenized and is a list of words) and a *word*, your **predict** function will return the name of the *word sense (in WordNet)* of this word. This function will be used to evaluate the performance of your method.

```
from typing import List


def predict(sentence: List[str], word: str):
   # Your implementation
   # Return the name of the word sense in WordNet
   return ...

# Example
sentence =  ['I', 'went', 'to', 'the', 'bank', 'to', 'deposit', 'money', '.']
word = 'bank'

word_sense = predict(sentence, word)
print(word_sense)

# Output will be a string
assert word_sense == 'savings_bank.n.02'
```

You must submit the Python Notebook (*.ipynb) of your source. You need to make sure that anyone can run the code from scratch (your notebook must include the commands to install any required libraries or download any required resources).

Your code cannot be evaluated if we cannot run your **predict** function.

Datasets & Corpora

The SemCor corpus is provided to train the model for this project.
WordNet is the sense inventory for English used in this project.

SemCor: https://www.kaggle.com/datasets/nltkdata/semcor-corpus
WordNet: https://wordnet.princeton.edu/
Example usages of corpus in NLTK library: https://www.nltk.org/howto/corpus.html

You can access these resources via the Python library **NLTK**. The instructions will be given during Tutorial hours. Alternatively, you can download these resources from the above website and access them manually.

Evaluation

Your work will be evaluated based on:

- The quality of your report.
- The performance of your method (it will be evaluated using a **private Test set**)
- Your implementation of the method