Assignment 3

CS 421: Natural Language Processing

Due: April 2, 2025 (11:59 p.m. CST)

1 Introduction

Welcome to Assignment 3 for CS 421! In this assignment, you will learn to build models for **Word Sense Disambiguation** task using `WordNet`.

Word Sense Disambiguation is the task of determining which sense of a word is being used in a particular context. **WordNet** will be used for this task, which is an annotated large lexical database of English where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Don't hesitate to reach out on Piazza or during office hours with any questions you have as you complete it! Happy Coding!

2 Instructions

Each part of this deliverable is labeled as Code or Written. The guidelines for these two types of components are provided below.

2.1 Code

The questions need to be completed using Python (version 3.10+). There are **no external packages** required to complete this assignment. If you want to use an external package for any reason, you are required to get approval from the course staff on Piazza prior to submission. Templates are provided for each Code question (.py files) as supplementary material. Do not rename/delete any functions or global variables provided in these templates and write your solution in the specified sections. Use the `main` function (provided in templates) to test your code when running it from a terminal. Avoid writing test code in the global scope; however, you should write additional functions/classes as needed in the global scope. These templates may also contain important information and/or examples in comments so please read them carefully. This part of the assignment will be graded automatically using Gradescope.

To submit your solution for Code questions, you need to compress the following files (after completion) in a single zip file. These files should be in the root of your zip archive for autograding to work correctly.

☐ `wsd.py`

Submit this `zip` file on Gradescope under `Assignment 3 - Code`. All specified files need to be submitted to receive full credit.

2.2 Written

You are required to submit all Written questions in a single PDF file. You may create this PDF using Microsoft Word, scans of your handwritten solution, LaTeX or any other method or design tool you prefer. To submit your solution for Written questions, you need to provide answers to the following questions in a single PDF.

☐ Q2

Before submission, ensure that all pages of your solution are present and in order. Submit this PDF on Gradescope under `Assignment 3 - Written`.

3 Questions

3.1 Code (80 points)

Q1 (80): Lesk Algorithm for Word Sense Disambiguation
In this question, you will implementing Simplified Lesk algorithm for Word Sense Disambiguation task.

1. Load SemCor corpus using NLTK[1] with `semcor.sents()`. Similarly, load WordNet model in NLTK[2] as `import wordnet as wn`. Select the first 50 sentences and store the sentences (`sents()`) and their corresponding tagged version (`tagged_sents()`) as data and labels for the 2 models.

2. Our first model for word sense disambiguation is Most Frequent Sense model, in which, as the name suggests, we choose most frequent sense for each word from the senses in a labelled corpus. For wordnet, this corresponds to the first sense in `synset()`. Using `synset()` and `definition()`, find the sense for each word. Evaluate and report the results using precision, recall and F-1 score.

3. Our second model is Simplified-Lesk algorithm as follows:

[1] `https://www.nltk.org/api/nltk.corpus.reader.semcor.html`
[2] `https://www.nltk.org/api/nltk.corpus.reader.wordnet.html`

Algorithm 1 Simplified Lesk Algorithm

```
function SimplifiedLesk(word, sentence)
    best-sense ← most frequent sense for word
    max-overlap ← 0
    context ← set of words in sentence
    for each sense in senses of word do
        signature ← set of words in the definition and examples of sense
        overlap ← ComputeOverlap(signature, context)
        if overlap > max-overlap then
            max-overlap ← overlap
            best-sense ← sense
        end if
    end for
    return best-sense
end function
```

Here, `ComputeOverlap` method calculates the number of words overlapping in the context (sentence) and the definition of the word from wordnet excluding the stopwords. The sense with largest overlap is chosen.

4. Evaluate and report the results with tags from the dataset using precision, recall and F-1 score.

Supplementary material: `wsd.py`

3.2 Written (20 points)

Q2 (20): Word Sense Disambiguation
With examples from test set, compare and analyze the results of the two models. Your analysis will include the performance of each model, if the model is able to disambiguate the senses, if it fails then why, etc.

4 Rubric

This assignment will be graded according to the rubric below. Partial points may be awarded for rubric items at the discretion of the course staff.

Q1 (80 points possible)	
Q1.1	+15
Q1.2	+20
Q1.3	+35
Q1.4	+10
Q2 (20 points possible)	
Q2	+20