CE314/CE887 Assignment 2
Parsing and Word Similarity

Aline Villavicencio

Introduction

In this assignment, you will build an information extraction system and use it to find similar events in texts. You will also solve some parsing exercises using NLTK.

- The assignment is to be done in PAIRS. If you need to find a colleague to form a pair, you can use the forum in Moodle to do that.
- Both students need to submit a file, that has the same name following the format: registration number1 registration number2.zip file. The uncompressed folder should contain a report and a code directory.
- Read the **What to submit** section before you submit your work.
- Carefully read the instructions and details about the data.
- For the report, be concise. You **do not need to write more than 2 to 5 sentences** for each answer.
- This assignment is worth 20% of your module marks
- Please ensure that you submit versions of your work before the deadline, and if needed, update them, to avoid any problems.

The deadline for this assignment is as set by the School Office.

Important note on plagiarism

You are reminded that this work is for credit towards the composite mark in CE314, and that the work you submit must therefore be your own. Any material you make use of, whether it be from textbooks, the Web or any other source must be acknowledged as a comment in the program, and the extent of the reference clearly indicated.

Carefully read the university guidelines about plagiarism. All plagiarism cases will be referred to an academic offences procedure. No allowance regardless of whether the act was intended or unintended.

IMPORTANT: For Q1 to Q3, you only need to write the answers in your report. You do not need to submit the code.

You may find it useful to go over the Lab instructions before doing this assignment. Questions Q1-Q3 should be completed using NLTK.

Q1. Extending a Grammar (15 marks)

Use the following grammar as starting point for parsing sentences S1 to S3 with NLTK, extending the grammar as needed according to the grammatical rules of English, as discussed in the textbook. Use the Chart parser available in NLTK.

```
S  -> NP VP
NP -> Det Nom | PropN | NP PP
Nom -> Adj Nom | N
VP -> V NP | V S | VP PP
PP -> P NP
PropN -> 'Bill'
Det -> 'the' | 'a' | an
N -> 'bear' | 'squirrel' | 'park'
Adj  -> 'angry' | 'frightened'
V ->  'chased'  | 'saw' | 'put' | 'eats' | 'eat'
P -> 'on'
```

S1. Put the block on the table

S2. Bob chased a bear in the park along the river

S3. Bill saw Bob chase the angry furry dog

a) Which rules do you need to add to the grammar to parse S1 to S3? (10 marks)

b) How many derivations can you get for each sentence? (5 marks)

Copy and paste the output of the parsed sentences with the extended grammar.

Q2. Parsing Quality (25 marks)

Consider the sentences:

S4. *An bear eat an squirrel*

S5. *The dogs eats*

a) Are these sentences correct? What are the grammatically correct equivalents for these sentences? (5 marks)

b) Run 2 parsers from NLTK on the two sentences. What is the output of the parsers? (Copy and paste only these sentences and their derivations). Explain why the parsers are correct or incorrect. (10 marks)

c) Generate 2 other correct and 2 other incorrect sentences with this grammar. How would you have to change this grammar to prevent these sentences from being parsed? You can write your own rules to extend the grammar and ensure correct agreement. (10 marks)

Q3. Parsing Ambiguity (20 marks)

S6. *He eats pasta with some anchovies in the restaurant*

S7. *He eats pasta with a fork in the restaurant*

a) Do S6 and S7 have more than one interpretation? If so, draw all derivations and briefly describe each of the interpretations. (10 marks)

b) Run the Shift Reduce Parser and the Earley Chart Parser from NLTK on these sentences. Which of the parsers detects the ambiguity for S6 and S7? Copy each interpretation generated by each parser (10 marks)

Q4. Calculating similarity between words (40 marks)

Write a computer program that reads a file specified by the user, generates the vocabulary of the file and calculates the word similarity between each two words in the dictionary according to WordNet. The program will output a file where each line corresponds to a word pair and the value for the WordNet similarity between them.

Data: Download the file `sim_data.zip`. The folder has 2 files: `BioSim-100.txt and text1.txt`.

`BioSim-100.txt` contains in each row a pair of words and their similarity according to humans. Your aim is to produce a file like this, with columns separated by tabs.

Task1: Build a program to calculate word similarity in BioSim-100.txt using WordNet. For each word pair in BioSim-100.txt you will calculate the WordNet similarity between the pair, using the path_similarity function implemented in NLTK, and print this into a file, along with the gold standard similarity. (10 marks)

The program when run should print the following to the file BioSim-100-predicted.txt:

```
word1 word2 GoldSimilarity WordNetSimiliarity
w1 w2 0.75 0.34
w3 w4 0.12 0.45
w5 w6 0.01 0.26
...
```

Task2: Build a program to detect word similarity in other texts. You will need to pre-process the user specified input text, reading the file, performing sentence splitting, tokenization and lemmatization, and removing stopwords and punctuation. The resulting file should contain only content words, one word per line. For each word in the file you will calculate the WordNet path_similarity between the pair, and print this into a file. Now apply your program to the file text1.txt. (10 marks)

The program when run should print the following to the file original-pairs.txt:

```
word1 word2 Similarity
w1 w2 0.75
w1 w3 0.12
w1 w4 0.01
w2 w1 0.75
```

```
w2 w3 0.123
w2 w4 0.008
w2 w5 0.00065
```

Task3: Replace each word by its hypernym and calculate the similarities between each word pair printing this additional information to the file original-pairs-hypernyms.txt. (10 marks)

The program when run should print the following:

```
word1 word2 Similarity1 hyp1 hyp2 Similarity2
w1 w2 0.75 h1 h2 0.8
w1 w3 0.12 h1 h3 0.06
```

Task4: What are the 10 most similar pairs that you found for text1.txt? Print them to the file top.txt. (10 marks)

Copy and paste the output into your report.

Submission

The assignment, which counts for **20%** of the overall mark, should be submitted via the electronic submission system by the specified deadline.

The file you submit should be a zip file that includes

- **Report**. A file containing the answers for the questions in this assignment. If needed, paste the output of your code into the report.
- **Code**. Each exercise should be implemented in a separate file. Your code should run without any arguments. It should read files in the same directory. Absolute paths **must not** be used. When downloaded, you code should run with a simple command such as `python ParseText.py`.
- a **README.txt** file that explains for each exercise, how to run each program. The file should have a single line giving the command to run your code. Check your code by downloading your .zip file into a different machine and testing that it runs without modification. When your code is run, it should print values in the format specified in each question. Negative marks will be applied if code does not run out of the box.

The guidelines about late assignments are explained in the handbook.

IMPORTANT: Write your registration numbers in the report and also as comments in your code.

Assessment criteria

What we are looking for in your answers:

Clear understanding of the concepts demonstrated by taking the right approach, correct substitution and accurate answers

Ability to use concepts learned in class demonstrated by clear answers to questions which ask to analyze numbers and output

Delivering the requested software solutions demonstrated by code which satisfies the criteria, outputs the required solutions cleanly, runs without dependencies, contains proper comments. You will not be evaluated on the efficiency of your code or algorithms as long as they can be executed correctly (without errors or error messages) and run in reasonable time.