COMP34112
Natural Language Systems

Coursework 1

In this coursework, you will explore corpus-based approaches to POS tagging and distributional semantics. The coursework is worth a total of 30% of the final COMP34412 final mark.

All the datasets needed for the coursework are in Blackboard. Please use Python and provide detailed comments – submitting a Jupyter notebook is fine. You may want to use functions that are available in the NLTK framework to implement the features required.

The coursework submission will have both code and a report.

Task 1: Part-of-speech tagging [7 marks]

Use the POS-tagged Brown corpus (corpus A in Blackboard) to estimate word likelihood and tag transition probabilities you would need to be able to disambiguate which of the following two POS tagging results is more likely:

> (1) People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN
>
> (2) People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/VB for/IN outer/JJ space/NN

Explain what you have done and comment/explain the results in the report (½ page). Please do not include book-work theory and explanations.

Task 2: Distributional semantics [23 marks]

The task here is to implement, evaluate and analyse a program to cluster a given list of *m* target words into *n* groups based on their distributional co-occurrence patterns, so that words that appear in *similar contexts* (represented by these "co-occurrence patters") are in the same cluster. Your program should take as input a list of words to cluster, a number of clusters and a corpus that is used to learn the patterns from. Note that nltk provides some clustering modules (https://www.nltk.org/api/nltk.cluster.html) but you can use any available machine-learning framework for clustering (e.g. Weka (http://www.cs.waikato.ac.nz/ml/weka/), sckikit-learn (http://scikit-learn.org/), etc.).

a) *Implementation*: your programme will need to cluster target words, but a key challenge is to design and implement how you will represent the context of a given (target) word. As an idea, you may want to construct a word-by-word matrix with different features that capture co-occurrence patterns of given target words using a given corpus. Design and use features flexibly so that you can analyse the results in part c) below. Explain your rationale in the report (½ page).
b) *Evaluation*: run your programme using the Brown corpus (available in Blackboard) and a target list of words (with 50 words, also available in Blackboard) to evaluate the results of your clustering. To make the evaluation feasible, use the following pseudoword disambiguation approach: for each target word, randomly substitute half of its occurrences in the corpus with its reverse (e.g., half of the occurrences of "*procedure"* will be transformed into "*erudecorp*") and treat these as different words. Now, apply your clustering algorithm to the list of 100 target words (containing 50 original words and 50 reverses). If you now generate 50 clusters, how many of them will contain "correct" pairs

(i.e., a word and its reverse)? Repeat this process 5 times (randomisation of reverse words) and give the average accuracy. Explain the results in the report (½ page).

c) *Analysis:* analyse and discuss your implementation by focusing on the impact on the quality of generated clusters of the following decisions you have made:
    1. the size of context used (i.e. dimensions/definition of the context window)
    2. type of features used to represent the context (e.g. stems vs. words, POS tags, stop-words, etc.), and
    3. corpora used (e.g. in addition to the Brown corpus, use also a product review corpus (available in Blackboard) and run your system on each corpora separately, and also on their combination).

    Discuss the pros and cons of your decisions, comment on the results obtained for the list of target words used above (with some examples if appropriate), and report/explain any differences (1 page).

    What other type(s) of feature or other resources you may consider using (you don't need to implement or analyse the impact of the additionally proposed feature(s)) (½ page)?

Data
All the datasets are available in Blackboard.

Submissions
The deadline for submissions is 6pm on **Friday March 12th 2021**. Your submission should be a single zip file uploaded via Blackboard. You should submit a single pdf report (up to 3 pages) that clearly explains what you have done and presents the results in a form you consider appropriate. You should also submit your source code and the output of your code (where applicable). The README file should clearly specify how to run your program.

Please add your name to the report and make it look professional. Please provide useful comments in the code. All code and reports will be checked for academic malpractice.

Marking schema and indicative rubrics:

Task 1 [total of 7 marks]:
- Probability estimations [total of 4 marks]
    - Word likelihood probabilities calculated (2 marks)
    - Tag transition probabilities calculated (2 marks)
- Implementation and discussion [total of 3 marks]
    - Discussion of the result (2 marks)
    - Quality of implementation (1 mark)

Task 2 [total of 23 marks]:
- Implementation [total of 8 marks]
    - Flexible context representation (e.g. word-by-word matrix) (4 marks)
    - Target words represented and clustered (2 marks)
    - Quality of implementation (2 marks)
- Evaluation [total of 5 marks]
    - Corpus properly 'randomised' by target words (2 marks)
    - Average accuracy calculated (1 mark)
    - Discussion of results (2 marks)
- Analysis [total of 10 marks]
    - Window size effect discussions (2 marks)
    - Context feature discussions (3 marks)
    - Corpora discussions (3 marks)
    - Other ideas for features/resources (2 marks)