COMP34711 Natural Language Processing

Coursework 2, Nov 2022

You are provided with the product review corpus. Check the README file and observe the content, format and structure of the corpus. You are asked to design and evaluate solutions for two NLP tasks using this corpus. Overall, this coursework is marked on the basis of

- rigorous experimentation,
- knowledge displayed in report,
- independent problem-solving skill,
- self-learning ability,
- how informative your analysis is,
- language and ease of reading of the report,
- code quality based on correctness and readability (which includes comments).

To implement your design, you are strongly recommended to use functions that are available in the NLTK framework and machine learning libraries specified in the instruction. This includes Weka, scikit-learn, PyTorch (above version 1), TensorFlow (below version 2) and tensorflow.keras. If you choose to use libraries beyond these, you are responsible for enabling the TAs to run your code directly without needing to install extra things themselves. Otherwise, there could be marks reduced because of this. If you are not sure, please speak to a TA for advice.

You should **solve all the tasks on your own**. You are not permitted to collaborate with other students on this coursework. In lab support sessions, you can ask the TAs to explain knowledge taught in the lecture or seek advice on how to use a natural language processing or machine learning library. But you are not permitted to ask TAs to help with the solution design, or to check the correctness of your solution.

Your submission should include both code and report. About your code, provide comments when you see fit and your code will be marked based on both correctness and readability (which includes comments). About your report, use **Arial Font 11**. Your main report should be no more than **2 pages**. If needed, you can include additionally up to 1 page of screenshots (e.g., of your results) as an Appendix of your report.

Task 1: Distributional Semantics (13 marks)

The following experiment is for evaluating the performance of a distributional semantic approach.

- **Step 1**: Clean and pre-process all reviews in your text corpus as you see fit. Choose the top 50 most frequently occurred words (after removing the stop words) as the target words. You are free to use functions that are available in the NLTK framework to help your text pre-processing.

- **Step 2**: For each of the 50 target words, uniformly sample half of its occurrences in the corpus and substitute these with a made-up reverse words, e.g., half of the occurrences of "canon" will be transformed into "nonac". Refer to these 50 new words as pseudowords.

- **Step 3**: Construct a d-dimensional feature vector to characterise each of the 50 target words and 50 pseudowords using a distributional semantic approach (more detailed requirements are provided later). Store your obtained feature vectors in an $N \times d$ matrix $\boldsymbol{X}$ *(N=50+50=100)*.

- **Step 4**: Take the feature matrix ***X*** as the input, and apply a clustering algorithm to cluster the 100 words into 50 clusters, where **no empty cluster** is allowed. You are free to use existing clustering algorithm implementation as you see fit. For instance, clustering modules (https://www.nltk.org/api/nltk.cluster.html) from NLTK, machine-learning framework for clustering from Weka (http://www.cs.waikato.ac.nz/ml/weka/) and scikit-learn (https://scikit-learn.org/stable/modules/clustering.html#clustering).

- **Step 5**: For each pair of the target word and its corresponding pseudoword, if these two are grouped into the same cluster, it is defined as a correct pair. Among the 50 pairs, check the percentage of the correct pairs, denoted by *p*.

- **Step 6**: Repeat this whole process multiple times, e.g., 5-10, and calculate the mean and standard deviation of the obtained percentages *p*.

Applying what you have learned on lexical processing and distributional semantics, you should obtain a **dense distributional semantic representation** for each word and pseudoword. You should put thoughts in ways of constructing the dictionary (e.g., stems vs. words), extracting the context features, and the use of dense approaches. You should aim at achieving a good clustering performance and understanding the reason behind. You should conduct a performance analysis of your approach and discuss hyperparameter setting.

1. Submission Instruction

Your implementation should be well-structured, defining a function for each step and executing the functions in a main file.

You should submit the implementation and evaluation of your approach as one Jupyter notebook file, named as "Task1". The TA will run this file during marking.

You should prepare a report (up to 1 page) containing two sections:

- Method Description: Explain your text cleaning and pre-processing steps, as well as your approach for constructing the distributional semantic representations.
- Result Analysis: Analyse and discuss the obtained clustering results. You should discuss hyperparameter relevant issues if your approach requires any hyperparameter setting, e.g., setting context window size, determining feature dimensionality *d*, etc.

2. Mark Allocation

Marks are allocated as below (13 marks in total):

- 2 mark for text cleaning, pre-processing, target words selection, pseudo words construction.
- 6 marks for implementation and description of your dense representation learning approach.
- 3 marks for clustering result analysis, based on it discuss the choice of your approach.
- 2 marks for design novelty of your approach. This can be either an improvement of what has been taught or a new reasonable approach not taught in the "Distributional Semantics" Chapter. You need to highlight in the report what the novelty is, if to gain these marks.

Task 2: Neural Network for Classifying Product Reviews (12 marks)

The product review corpus contains reviews scored as positive and negative opinions. Pre-process your text, prepare the review examples for training and evaluation. Implement, train and evaluate a neural network that can classify an input review to either a positive or a negative class. You are free to choose any neural network/deep learning technique taught in the Chapter "Deep Learning for NLP", e.g., multi-layer perceptron, LSTM, bi-directional LSTM, etc. You should design appropriate experiments to evaluate your classifier's classification accuracy based on 5-fold cross validation (CV).

1. Submission Instruction

Your implementation should be well-structured with comments. You should submit the implementation and its evaluation as a single file, named as "Task2". The TA will run this file during marking.

Prepare a report (up to 1 page) containing 2 short sections:

- Method Description: Explain your classification model design and training.
- Experiment and Result Analysis: Describe your experiment and evaluation approach. You should discuss hyperparameter relevant issues if your approach requires any hyperparameter setting. Report and analyse classification accuracy.

2. Mark Allocation

Marks are allocated as below (12 marks in total):

- 2 marks for text cleaning, pre-processing, and preparing the input data for the classifier.
- 5 marks for implementation and description of the classifier.
- 2 marks for designing and implementing experiments to evaluate classification accuracy based on 5-fold cross validation.
- 3 marks for result analysis of the classifier and discussion of hyper-parameter setting.

Submission Checklist

A .zip file named as "34711-Cwk-S-DeepLearning" containing

- Two code files: Task1 and Task2.
- One .pdf file, combining reports for both Task 1 and Task 2.