Natural Language Processing – Assignment I

Reykjavik University – School of Computer Science

Fall 2023

The purpose of this assignment is to experiment with several text processing tools, for example in the context of language modeling and writing a Python program. In addition, you will experiment with word embeddings and OpenAI's GPT models. The assignment consists of 4 parts, in total worth 64 points.

1 Python: Gutenberg corpus (16p)

Write a Python program, *corpusAnalysis.py*, which prints out various statistics/information about a given text file, which is a part of the Gutenberg corpus (accessible using NLTK). The name of the text file is given as a parameter to the program, i.e.

```
python corpusAnalysis.py carroll-alice.txt
```

The program should print out the following information for the given text file:

```
Text:  carroll-alice.txt
Tokens:  34110
Types:  3016
Types excluding stop words:  2872
10 most common tokens:  [(',', 1993), ("'", 1731), ('the', 1527), ('and', 802),
('.', 764), ('to', 725), ('a', 615), ('I', 543), ('it', 527), ('she', 509)]
Long types:  ['affectionately', 'contemptuously', 'disappointment', 'Multiplication']
Nouns ending in 'ation'  ['usurpation', 'station', 'accusation', 'invitation',
'consultation', 'sensation', 'explanation', 'Multiplication', 'conversation', 'Uglification',
'exclamation']
```

Note: Long types are those with more than 13 characters.

Return your program code (.py file) along with the output of your program when running against the file *austen-emma.txt*.

2 Language Modeling (16p)

In this part, you develop an English trigram language model based on *eng.sent*[1] by only using the following UNIX tools: `awk, head, tail, paste, sort, uniq, wc`. Note that here we are only interested in word (token) trigrams (including punctuations), not part-of-speech trigrams.

1. (3p) *eng.sent* is pre-tokenised even though the <token,tag> pairs do not appear on a separate line. However, in order to construct the language model you need a file with one token (word) per line without any empty lines. Use *awk* for this purpose. Show the *awk* command that you use for constructing this file, *eng.tok*.

2. (6p) Show the sequence of commands you use to construct a trigram frequency file *engTri.freq*[2] (from *eng.tok*), sorted in descended order of frequency. Note that you can specify the flag $-k1,1$ to *sort*, for specifying sorting only on the first column.

[1] Due to copyright reasons, please make sure you do not distribute this corpus.
[2] Containing four columns: frequency, $word_1$, $word_2$, $word_3$.

3. (3p) How many trigrams and distinct trigrams exists in *eng.sent*? Use *awk* and *wc* and *engTri.freq* to figure this out (show your commands and the output).

4. (4p) Use the data from *engTri.freq* to estimate (using Maximum Likelihood Estimation):

   P(Monday | said on)

   Show which lines from *engTri.freq* you use to estimate this probability and your calculations.

3 Bias in Word Embeddings – 16 pts

Word embeddings are dense vector representations of words that have successfully improved the state-of-the-art on many NLP tasks. These vectors are produced by training classifiers on text corpora, such as Wikipedia, Twitter, and news sites. This data may contain biases of various kinds, e.g., denigration, stereotyping, recognition, and under-representation. Specifically, word embeddings may contain gender bias [1, 2], sexual orientation bias [3], ethnic bias [4], and ageism 5. These biases may be present in the word embeddings themselves and therefore have an adverse effect on the NLP task they're being used for, e.g., text classification, text summarization, language generation, and machine translation [6].

Investigate whether there are any potential biases to be found in word embeddings created using (1) data from Wikipedia and (2) data from Twitter (e.g. `glove-wiki-gigaword-100` and `glove-twitter-100`). There are several methods you can use to accomplish this task. For example, Lab 3 has methods for finding similarities and relationships between words that you can use and adapt, as you see fit (e.g., *find_word*, *most_similar*, *similarity*, *similar_by_vector*, *doesnt_match* – See more in the Gensim docs, like KeyedVectors). You may for instance explore combining vectors in various ways (e.g., using the NumPy package) or use any other method you discover.

Answer the following questions in essay-form:

1. What kinds of biases are present in either dataset, if any?

2. Are there any differences with respect to bias between the two kinds of data?

3. Can biases between concepts be revealed through visualizations?

4. Can bias in word embeddings be removed? If so, how?

Write about your findings and show example code you used to investigate this, including examples of relevant output. Additionally, there are several web-apps on the internet for visualizing word embeddings (e.g., WebVectors). Use one of these tools is to plot relations between concepts and include in your handin.

4 Case Study - GPT – 16pts

GPT-3, 3.5, and 4 are large language models developed by OpenAI. In this task you are to study the capabilities of a model using OpenAI's "Playground" available through their website `https://openai.com`. You can use 3, 3.5, or 4, depending on the the kind of access you have. It does not matter with respect to this assignment. Just make sure you indicate what model you are using. Perform the following steps to get access to their services:

1. Go to `https://openai.com`

2. Click on "API" and then "Sign Up"

3. Create the account with a username and password

4. Log-in using your credentials

5. Click the "Quickstart Tutorial" to get started
6. Explore the capabilities of the model on the "Playground" and "Examples" pages

Concretely, investigate the capabilities of the GPT model:

1. Have at least five sessions with the model over the course of a few days.
2. Explore at least one aspect of this model per session.
3. In an essay-format, answer the following questions based on your investigation:
   (a) What are the capabilities of the GPT model?
   (b) What are the parameter settings and how do they affect the model?
   (c) What data was used for training?
   (d) What are the model's limitations?
4. Show examples for each of the capabilities and limitations you explored. Include model prompts, settings, and output you used during your sessions.

5 What to return

A single .zip file containing the following:

1. The Python program **corpusAnalysis.py** and a file containing the output of your program when running against the file *austen-emma.txt*.
2. A PDF file that contains: (*i*) the responses to parts 1-4 in Section 2, (*ii*) the essay in Section 3, and (*iii*) the essay in Section 4.