Mandatory Assignment 01
Natural Language Processing and Text Analytics (KAN-CDSCO1002U)

Ask question to: Rajani Singh (email: rs.digi@cbs.dk)

Deadline: **{check digital exam for exact date and time}**

Instructions

1. This mandatory assignment contains two main parts. Part 1 is mandatory and in part 2 you have to choose any one question to answer.
2. You will have to upload your solutions before the deadline on **Canvas**. Include you student numbers on the mandatory assignment on Canvas.
3. Please use Python 3 for answering the following questions whenever it is necessary.
4. It is important that you use comments extensively in your code, so that it will be easy for the reader.
5. The Python code for answering the questions can be either submitted as one single *jupyter* notebook or multiple *jupyter* notebooks. In addition to submitting your *jupyter* notebook/notebooks, please also export the Python code as a file from the *jupyter* notebook ('*File*' → '*Download as*' → '*Python (.py)*') and upload the exported Python code file in addition to *jupyter* notebook. Alternatively, if you are using any other IDE (Integrated development environment), then you can also submit your code as one single python file (with .py extension) containing all the source code from different classes/modules/functions etc.
6. In case, if your answers involve any math, then please feel free to use whatever format you like [Latex, Word, plain paper and pen], but make sure that it is clearly readable.

Part 1: Mandatory Question

The following two sentences are ambiguous – that is, they each have two (or more) different readings. For each sentence, explain the different readings:

1. The girl attacked the boy with the book.
2. We decided to leave on Saturday.

Are the following sentences ambiguous? If the sentence is ambiguous, explain the different readings. If not, explain why not.

3. I saw a man with a briefcase.
4. I saw the planet with a telescope.

Try the above sentences (3 and 4) in this parse tree generator:
`https://huggingface.co/spaces/nanom/syntactic_tree`
Compare the dependency structures for both of them. Explain the differences in the link to the preposition "with" in 3 vs. 4.

Part 2: Select any one question to answer in this part

[Linguistic Analysis of a Text Corpus Using spaCy]

There is an uploaded text corpus named "**sample.xlsx**". Conduct a linguistic analysis using Part-of-Speech (POS) tagging and Named Entity Recognition (NER) in spaCy library only on the "**SOS Tweet/SOS Message**" column. To complete this question you have to do the following tasks:

1. **Prepare the Corpus:** Pre-process the corpus data by removing stop-words, removing extra spaces, converting all text to lowercase, or apply any other text normalization techniques that you think is relevant.
2. **POS Tagging and NER:** Apply POS tagging to the entire corpus to analyze the distribution of different parts of speech. Use NER to identify and categorize named entities within the text such as name, date, locations etc.
3. **Analysis:** Determine the most common POS tags and discuss what this reveals about the structure of the text.
4. **Prepare a short report:** Summarize your findings in a brief report of 1-2 page, including any interesting patterns or insights gained from the POS and NER analysis.

[Next word prediction system based on n-gram model]

In this part you are going to work with language model using Natural Language Toolkit (NLTK) [1]. Create a n-gram (either for n=3 or n=4) language model and use it to predict next word. To complete this question you have to do the following tasks:

1. **Prepare the Corpus:** Choose a text corpus of your choice from Gutenberg as your training corpus. As long as the corpus is of significant size (few mega bytes of text), it should be fine. For example you can choose 'austen-emma.txt'.

   Pre-process the corpus by removing stopwords, removing extra spaces, converting all text to lowercase, or apply any other text normalization techniques that you think is relevant.

2. **Build the Language Model:** From the preprocessed text, generate n-grams. You might use NLTK's ngrams function to do this. Calculate the counts of each unique n-gram in the corpus. Calculate the conditional probability of each word following a given (n-1) word sequence based on the n-grams counts.

3. **Word Prediction Mechanism:** Create a function that takes a sequence of (n-1) words as input and returns the most probable next word based on your n-gram model. The function should handle cases where the input sequence has not been seen in the training data (you may choose a simple strategy such as returning a random word from the corpus or use any smoothing technique).

4. **Evaluation:** Test your model with 10 input sequences of (n-1) from within and outside the corpus. Assess the model predictions ability by checking if the predicted next words are sensible.

5. **Prepare a short report:** Summarize your findings in a brief report of 1-2 page.

[Hint] Note that the probabilities of n-grams will have very low values and in order deal with multiplication of low probabilities and thereby arithmetic underflow, you can use logarithmic (log) values for probabilities.

References

[1] S. Bird. NLTK: The Natural Language Toolkit. In J. Curran, editor, *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia, July 2006. Association for Computational Linguistics.