Assignment 2:
Word Embedding & LSTM

- This assignment is due on **Jun 16th, 2024 (23:59, CET)**
- You can discuss the problems with your classmates or browse the Internet to get help. However, copy and paste is cheating.
- You need at least 50% of the points of every assignment to take part in the exam.
- There are 3 assignments in total.
- Submit at `https://elearn.informatik.uni-kiel.de/course/view.php?id=274`
  - only pdf and jupyter files and only one zip file per assignment.
  - put your names and matriculation numbers on *each* page in the pdf file

Task 1: Word2vec

Train Word2Vec Model on news category dataset. You can download this data from this link: News Category Dataset .

Feel free to utilize any libraries such as Scikit-learn, Keras, PyTorch, gensim, etc.

a) Dataset Preparation **2 P**

- Load the dataset and explore its structure and contents.
- Tokenization; Split text into individual words
- Clean the data (Remove non-alphabetic characters,. etc)
- Convert words to lowercase.
- Apply stemming or lemmatization (if you think it's needed)

b) Data Splitting **1 P**

- Split the dataset into training and testing sets to evaluate the model later. Data split ratio can be 80 to 20 percent.

c) Word2Vec Model Training **10 P**

- Prepare the tokenized text data for Word2Vec training.
- Initialize the Word2Vec model with appropriate parameters:
  - vector size, window size, minimum cunts, sg and workers. (choose your parameters that works best for this dataset)

d) Model Evaluation **5 P**

- Save the trained Word2Vec model to disk for future use.
- Load the model and test it by finding similar words and their vector representations
- **Word similarity**: Find similar words for each of these given words. (You can used cosine similarity between word vectors)

  Words: americans, COVID, sleeves and british
- **Analogy tasks**

  king - man + woman = ?

  man - woman + son = ?

  berlin - germany + france = ?

  beijing - china + tokyo = ?

e) Visualization **3 P**

Visualize word vectors using t-SNE (t-Distributed Stochastic Neighbor Embedding) to reduce dimensionality.

- Plot the most frequent top 200 to 400 words in a 2D space for test dataset.

Task 2: LSTM: Multi Class Text classification

Implement a Long short-term memory (LSTM) for the same dataset used in task 1.

a) Exploratory Data Analysis **2 P**

- Calculate and visualize the class distribution
- Calculate and visualize the distribution of unigrams and bigrams (top 20 most frequently occurring terms after removing stop words).
- Calculate and visualize the number of words in each sample.
- Plot wordclouds for two classes: one with the highest number of samples in the dataset and another with the lowest.

b) Data Pre-processing **3 P**

- Remove stop words.
- Apply either lemmatization or stemming based on your analysis (provide reasoning for your choice).
- Remove punctuation, non-English characters, special characters, and URLs (if present).

c) Data Split **1 P**

- Divide the dataset into training and testing sets, with 80% allocated for training and 20% for testing

d) Word Embedding **5 P**

- Use the Word2Vec embeddings that you trained in Task 1 to convert the text into numeric form.

e) Model Designing and Tainning **6 P**

- Implement LS TM model network. You have the freedom to design the architecture, including the number of neurons per layer, the number of hidden layers, choice of activation function, loss function, optimizer, etc.
- Experiment with different architectures, activation functions, and optimization algorithms to improve the model's performance.

f) Data imbalanced (optional) **3 P**

- Check if the data is imbalanced or not.
- Explore and design a solution to tackle this problem.

g) Model Evaluation **2 P**

- Evaluate the model by computing F1 scores (Micro and Macro) and accuracy scores.

h) TensorBoard **3 P**

- Implement TensorBoard to visualize the plots of losses, F1 scores, and accuracy scores on both training and test data.