Fake News Identification

With the spread of fake news on social media and its social and political impact, it has become vital to be able to identify fake news articles. Natural language processing and machine learning are seen at the forefront of fighting and identifying fake news.

Given your knowledge in Natural Language Processing and deep learning, develop a solution that is able to identify fake news. To help you build it, a dataset of fake and real news articles is provided. The data contains 6335 articles with each marked as FAKE (0) or REAL (1). The file contains 3 columns: ID, TEXT, LABEL. 3164 articles are fake news, and the rest are real news.

The goal of the coursework is to take you through the complete process of approaching a data science project from environment setup, to problem solving, solution implementation, reporting, and, final presentation.

The course work consists of viewing the problem from two perspectives: the shallow learning and deep learning. **You are expected to implement both approaches**.

Shallow learning approach:

A. Pre-processing and feature extraction:
Given that the textual information are unstructured data.

I) Apply what you see necessary to clean the data. **(5 marks)**
II) Use NLP techniques to extract numerical features that you could use to build a classifier. This may include: tf, *tf-idf, N-grams*. **(5 marks)**

B. Classification:

I) Use a Multinomial Naïve Bayes classifier to discriminate fake from real news. **(5 marks)**
II) Compare the classification results using the different features you extracted in the previous step. Use the classification accuracy, Precision, Recall, and F1-measure as comparison metrics. **(5 Marks)**

Deep Learning approach:

C. Feature Extraction:

I) Write a function that takes a text as input and returns a list of word2vec features using a pre-trained word2vec model. **(5 Marks)**

D. Classification:

I) Build a Long Short-Term Memory (LSTM) to model the sentences in the news articles that can discriminate the articles into fake and real. **(15 Marks)**

II) Compare the results of your LSTM model with those of a Recursive neural network (RNN) in terms of quality metrics as in *BII* and the execution time. **(10 Marks)**

E. Results Presentation:

I) Write a main function within a file ***main.py***. The functions should present in a clear and understandable format all the output required to address all the problems above. **(5 Marks)**

Tools and Hints:

You have the freedom to choose the tools to use within the Python framework. Suggested libraries:

- Pandas
- numpy
- Spacy
- NLTK
- Keras
- Tensorflow
- Scikit-learn

Pandas is useful for data pre-processing/cleaning and grouping.

Spacy (https://spacy.io/) is an example of a package that provides pre-trained Word2Vec models for English.

The LSTM will use sequences of word2vec features extracted from each article. Set the maximum sequence length to 1000. Use zero-padding for shorter sentences (in Keras you can use the pad_sequences utility function)

In Keras you may use an Embedding input layer (https://keras.io/layers/embeddings/) to map the word2vec features into a structure that Keras can process.

Remember to split your data into training and testing sets.

When working with LSTMs start experimenting with a subset of the data until you are satisfied with your architecture and then run the model on all the training data. This will save you time when debugging your code or deciding on model parameters.

Report:

Write a report to describe and justify your solutions and the design choices you have made at every step. In particular

I) justify the features used in A II

II) What is your conclusion of the comparison among the models in B II

III) Provide rationale for the architecture of your deep learning models and the choice of activation functions

Describe your conclusion of the use of the two approaches and discuss which approach you think is fit for purpose.

Be creative in presenting the results in a clear and understandable format.

Write a maximum of 2000 words. Figures and tables are excluded from the word count.
(20 Marks)

Additional Assessment Criteria:

A. General Performance of the solution on the test data set

- Are the results comparable or above the expected baseline (i.e. > 75 %accuracy)? **(10 Marks)**
- How all the components work together to achieve the reported results **(10 Marks)**

B. Code style:

- Clear, well documented program source code **(5 Marks)**

Submission:

1- Source code for both your shallow and deep solutions
2- Report, maximum of 2000 words
3- **Clarify** what libraries you have used and any specific installation instructions if applicable.
4- **CODE THAT DOES NOT RUN WILL LOSE ITS FULL ALLOCATED MARK.**