FIT 5212 - Data Analysis for Semi structured Data
Assignment - 1
Part 1 - Text Classification

Name: Anoop John Alocious
ID: 31084354

In this section, we have been assigned with a text classification task where we build different classifiers in order to predict the class of each article in the dataset.
We need to use 2 different preprocessing techniques based on which we will be building different classifiers. The 2 types of preprocessing techniques used are as follows:

1. Convert text to lower case, perform stemming.
2. Convert text to lower case, remove digits, remove special characters, remove stopwords, lemmatize the text.

Once we perform the necessary preprocessing techniques, we are going to train 2 different classifiers using the inputs received after the preprocessing.
The 2 classifiers that we will be using will be LinearSVC and RNN

Before training on the first 1000 articles in the dataset, we had inspected their labels and we could see there was a heavy imbalance in the number of 0's and 1's for each label. Training on these articles will most likely favour outcomes based on the majority class which are 0's. So, in order to help us get better results for our model, we have used gridsearch to tune the hyperparameters so that it can improve the model performance on the testing set.
Observations:
The model performance on the testing was rather poor even after tuning the hyperparameters. This is mainly due to class imbalance in the first 1000 instances of the training data. However, we can see that the second preprocessing method did give us a fairly better result as compared to the first one. We can see that the model built using the second preprocessing technique resulted in a better F1-score of 0.53, for the class label 'InfoTheory'as compared to the latter with F1-score 0.47. The results for the remaining labels are almost identical, and certainly the model does not do well in classifying the labels. To validate the model performance, we have checked the precision-recall curve. The plot shows that the area under the precision-recall curve seems to be low, and this suggests that the model is biased towards predicting the majority class better, and hence we need to use more training data to improve model performance.

Moving on, we train our classifier on the entire articles present in the training dataset. By doing this, we are expecting better results as the issue of class imbalance is now less significant as we have more number of 1's as compared to earlier.
Observations:
We can see that LinearSVC does a good job in classifying the labels for both the preprocessing techniques. For each label, it has a really good F1-score, and also the precision and recall values are also high. The number of false positives and false negatives have

reduced drastically, this proves that as we have more data and have preprocessed our texts well, we get better results. The average precision-recall scores for especially 2 labels ‘InfoTheory’ & ‘CompVis’ were 0.95, which is a good indication that the model performs well. By observing the precision recall curve, we can see that the area under each curve is really higher as compared to the results we got when we trained on the first 1000 instances. This indicates that the model performance has significantly improved.

When we trained the RNN on the first 1000 training instances, it seems to perform better than the LinearSVC algorithm and gave us better performance metrics for precision, recall and F1-score. This could also mean the RNN has overfit the training data very well, as the nature of any neural network tends to overfit the data when there is not enough data.

Next, we have trained a recurrent neural network on the entire dataset with 2 types of preprocessing methods.
Observations:
The results were made into confusion matrices for better understanding, and we could observe that the RNN does not perform as well as the LinearSVC for both the preprocessing techniques. The F1-scores of each label are significantly lower to the ones we got for the statistical model. Although one important observation is that, the RNN built using the preprocessing technique 2 seemed to perform better than the latter, as it did a good job in properly classifying all the texts for the label ‘CompVis’. This can be used to justify that the preprocessing technique 2 worked better for modelling 2 different classifiers.

The best configuration to use would be preprocessing the text by converting to lowercase, removing stopwords, removing digits/special characters and performing lemmatization. The results obtained are good enough for a text classifier, as it can classify each labels accurately

Part 2 - Topic Modelling

The primary intention of this part is to build and train topic models using LDA (Latent Dirichlet Allocation) algorithm. This algorithm focuses on processing huge chunks of texts and generating meaningful topics out of them.
We have been assigned to perform topic modelling for 2 x 2 data configuration, wherein, we are asked to use 2 different text preprocessing techniques on 2 different training of size 1000 and 20,000 respectively.
The 2 different preprocessing techniques that we have used are as follows:

1. Convert to lowercase, remove digits, remove words of length < 2 and perform stemming.
2. Convert to lowercase, remove digits, remove characters of length < 2, remove special characters, remove stopwords, lemmatize the texts.

The first version was trained using the first preprocessing technique and on the first 1000 articles of the training data. We also decided not to include the bigrams.
The second version was also trained on the same 1000 articles, whereas the second preprocessing technique was used and we have included the bigrams that have occurred in more than 5 documents.
The number of topics to be visualized was given as **5.**
Additionally, we have decided to remove the words that have appeared in less than 3 documents, and also that has appeared in more than 50% of the documents.

Observations:
Using version 1, we could identify topics related to language parsing & translation, grammar, semantics, language discourse processes such as centering
Titles of related articles:

1. `Parsing of Spoken Language under Time Constraints`
2. `An Abstract Machine for Unification Grammars`

The results from Version 2 was also similar to that of version 1, but it also discussed other topics like morphological analysis/processing of different languages like Japanese, Korean etc.
Titles of related articles:

1. `Chart-driven Connectionist Categorial Parsing of Spoken Korean`
2. `Integrated speech and morphological processing in a connectionist continuous speech understanding for Korean`

As we have used just 1000 articles, the output of the topic modelling was not very distinct.
Lastly, when we tried increasing the number of topics, we could observe an overlap between topics, this proves to show that there were no other topics that can be classified differently. It is also interesting to see that some bigrams like part_speech, natural_language were found in

version 2. This makes it easier for us to understand that there were articles related to natural language processing and so on.

Moving on, we have trained the third and forth versions on the first 20,000 articles, where 2 different preprocessing techniques were used in each of them.
The number of topics to be visualized was given as **10.**
We have included the bigrams that have occurred in more than 8 documents.
Similar to the previous 2 versions, we have decided to remove the words that have appeared in less than 7 documents, and also that has appeared in more than 50% of the documents.

Observations:
Using version 3, we can see that there are many more distinct topics as the number of articles have significantly increased. The set of topics include channel noise/interference, language logic program, security protocols for messages, algorithm optimization, image compression and recognition, application development and architecture, computational time for different algorithms, network diagram and algorithms.
Titles of related articles:

1. `Modeling Network Evolution Using Graph Motifs`
2. `A Systematic Approach to Web-Application Development`
3. `Resource Allocation in Public Cluster with Extended Optimization Algorithm`

Similar to version 3, version 4 also generates similar topics. But it is worth noting that as we used different preprocessing techniques, the overlap of topics that we got in version 4 is less than version 3. This proves to show that the preprocessing techniques were effective in generating sensible topics.
We can also see that it is easier to identify the topics from keywords obtained using version 4 as compared to version 3. There were 2-3 topics obtained using version 3 to which we could not really assign a subject. But using version 4, we were able to find some topics that were not well classified in version 3. For example, in version 3, the keyword 'game' corresponds to the topic of wireless network, but in version 4 we were able to find out that there are some articles about board games too.
Titles of related Articles:

1. `Gamed-based iSTART Practice: From MiBoard to Self-Explanation Showdown`

From our analysis, we can understand that the topics that we got when for the first 2 versions have not appeared for the remaining 2 versions. This goes on to prove that as we change the preprocessing techniques and the size of data, the number of topics change significantly.