Part 1: Text Classification

This part focuses on building a text classifier to predict ComputationalLinguistics. Different models, are adapted with different configuration, such as using statistical model vs Recurrent Neural Network (RNN), training on different datasize, using abstracts vs title only. After building the model, the performance is evaluated using metrics like F1 score, precision, recall, and accuracy on test dataset. The results are visualized in Part 1D to provide insights into the effectiveness and limitations of each model.

**Data Loading & Import libraries**

Data is loaded from existing files, including training, validation, and testing sets, each containing 5 columns: title, abstract, ComputationalLinguistics and other columns. The training set has 152,586 records, while the validation and testing sets have 19,073 and 19,074 records, respectively.

**Preprocessing:**

The goal in preprocessing is to ensure the data is prepared and ready for modelling. Data was tokenized using regular expression, lemmatized and clean from stopwods. All numeric and tokens with length shorter than 1 are removed as they carry no sematic information.

**Part 1A**

Three models were constructed using Logistic regression as ComputationalLinguistics has binary outcome. Three model differ in data input: the baseline model use abstract of entire training data and validation data, the second model use the title of all training data, and the last one use the abstract of only first 1000 articles. The data were vectorized using TFIDF vectorizer and those with less than 20 occurrences were removed for better model training process. The training data were combined with the validation data. 5-fold cross validation is implemented to avoid overfitting using the combined data to train the model.

Then the model takes the data input, depends on the variation, to make a prediction and compared against the actual label of ComputationalLinguistics in the testing set. The model is being assessed on its performance using the metrics listed above. In the baseline model, the cross-validation accuracy scores were shown, the accuracy increase with more fold. This shows the model performance improved over the training process. The same procedure was repeated with the other two input data configurations.

**Part 1B**

RNN is used in this session with the same 3 data input defined in previous part. A vocabulary was built with the pre-processed list of tuples, containing the data and its label. The data was parsed into batches for model training and optimized using SGD to minimize loss. Similar to cross validation, the model will be trained with 5 epoches and the best model with the smallest loss will be saved for later comparison.

The original training data has a dimension of 152586 x 5. The vocab list is set to exclude token with occurrence less than 20. There are 17070, 4196, 1105 unique token in the vocab of baseline model, model using title only and model using first 1000 records respectively.

To maintain consistency in evaluating the impact of input data on model performance, the embedding dimension and hidden dimension are kept the same across all three models.Embedding dimension is calculated as 350 using the rule of thumb (Stack Exchange). Hidden dimension was chosen to be 250, somewhere between the output dimension and embedding dimension.

Given there are 6125251 trainable parameter in baseline model, it is trained using GPU acceleration in Google Colab. During training, the model is fed with batches of training data and evaluated on a separate validation set to monitor its performance. The best performing model with the smallest loss is saved as the best model and is used to generated prediction on testing set, to be compared with the actual label. This approach allows investigation of how the data input affect the predictivity of RNN model.

**Part 1C**

All the results from previous part are combined in this session and stored in a pandas dataframe. The row represents the model, the column represent the each of the metric and hence the dimension is 6x5.

**Part 1D**

The visualised result using barchart and line plot revealed some patterns

* Between statistical and RNN model: In general, statistical models have a better result than rnn model. Only RNN model using title has a similar performance of the stat model using first 1000 abstract
* Among statistics models: the baseline model using all abstract perform the best, followed by using all title, then using first 1000 abstract only;
* Among rnn model: using title only has the best result and using abstract, regarding of the size, has a unsatisfying result comparatively.

This suggests using title is better in rnn model and this could due to the architecture and the shortcoming of rnn model. RNN model is prone to propagating error, meaning if a mistake was made at one stage, it will be carried onto the subsequent predictions. As abstract contains a large number of tokens, they may introduce too much noise to the model and hence lower the performance of the model.

MCC is a more comprehensive measures of the performance in identifying two classes. RNN baseline model and the last model both have a negative value, indicating wrong prediction compared to observation. Again showing RNN is weak in making prediction compare to statistical model. This could be confirmed by looking at the confusion matrix of RNN baseline model which has much more false prediction compared to logistic model.

In the Line chart illustrating the relationship between precision and recall, statistical model exhibit a concave relationship. Precision measures the number of correct positive prediction made among predictions while recall measures the correct positive case among the actual case. For statistical model, there is trade-off between precision and recall given its concave shape, often found in imbalanced dataset (Nath 2023). This is indeed the case with almost 2/3 of the data were labelled as 0.

From the graph, it can be concluded that the statistical baseline perform the best among other statistical model as the degree of concavity is the highest.

In contrast, for RNN, only three pairs of values are plotted in the graph. This indicates that only three unique thresholds are considered for a binary outcome. As a result, lines representing the RNN models are rougher compared to the logistic model, which outputs probabilities of class and hence a smooth curve.

The curves for RNN baseline and the third model shows a high precision, low recall, then low precision and recall , and low precision and high recall. This implies the mode is making little positive prediction and is being too conservative at first, but gradually became too sensitive by making too many false positive prediction (scikitlearn) . Therefore among 3 RNN models, model using title is the optimum model.

**Conclusion**

In general, it is expected Neural Network model to have a better performance compare to statistical model as NN models could capture more complex patterns and sematic information. However in this case, even with a large embedding and hidden layer dimension, logistic model still outperformed RNN. This is due to the nature of RNN is prone to propagation error especially for a long sequence of process. The huge size in vocabulary list might introduce too much noise and increase the chance of making wrong prediction. This explains the reason RNN model using title only perform the best. Meanwhile, logistics model using all abstracts has the most information to make prediction and hence has the best performance.

With the same preprocessing steps, dataset and input data, one underlying factor that might contribute to the bad performance of RNN model, is the imbalance class problem in the testing set, indicated by the precision-recall curve. RNN might be bias to one of the label and therefore fail to converge in gradients comprehensively. On the other hand, logistic regression produces probability predictions rather than strict class prediction and therefore the effect of imbalanced dataset has less effect on the results.

Part 2: Topic Modelling

The objective of this task is uncover grouping of articles by their topics using LDA model. Two variations of LDA model were employed with a difference in the use of bigram, the number of topics and the number of data input. There would be 2x2 configuration:

1. 1000 articles, without bigram, 10 topics
2. 1000 articles, with bigram, 40 topics
3. 20000 articles, without bigram, 10 topics
4. 20000 articles, with bigram, 40 topics

Output of above models were visualized to help interpretating the topics. The dataset used is the same as part 1 with same preprocessing process: tokenization, lemmatization, stemming and removing stop words.

Among the 4 models, Model 1 has the highest average coherence score of -1.9431, indicating a relatively stronger relevance between topics (Pickett 2020) although the score is still not satisfying. Focusing on the top 30 most salient terms, which represents the most important words of the topics (Chuang et.al 2012), the dataset seems to be data science related with terms like 'feature,' 'problem,' and 'training' suggest a focus on issues within the training process.

Moreover, some domain-specific keywords like 'image,' 'llm,' 'video,' 'word,' and 'network' indicates topics related to areas like natural language processing (NLP), complex data analysis, and neural networks. However, there are some overlapping between topics as some term appear across various topics such as ‘image’, ‘code’ etc.

In contrast to Model 1, the top 30 most salient terms in Model 2 present a different view. While some terms like 'image,' 'language,' and 'graph' overlap with those in Model 1, Model 2 includes new terms such as 'adversarial,' 'attack,' and 'face.' These suggest additional topics related to cybersecurity and potentially face recognition technologies. The overlap problem is more obvious as the number of topics increase, causing difficulty in interpretating the topics and groupings. This is reflected in an even lower average topics coherence score of -2.1224.

Model 3 and 4 score -2.1145 and -3.5622 in average topic coherence. Despite having a larger data input, both models are still failing to group the topics coherently. Model 3 has a more frequent occurrence of bigram compared to Model 1, suggesting a higher ability to capture more complex contextual relationship and patterns given a boarder range of vocabulary and larger input. However there are some signs suggesting the use of trigram. For example Topic 6 of Model 3 includes ‘ natural\_language’ and ‘language\_model’ and they could have considered the same thing as natural language model.

Model 4 shows fewer overlapping compare to Model 2, implying a clear topic grouping. However, noise in the dataset still pose a great challenge in interpretating the topics even with a larger data input. For instance in Topic 36, a lot of strange keywords were included such as ‘ri’, ‘bd’, ‘thz’. These anomalies seem to be related with the data quality rather than lemmatization error. Most of the topics on left of PC2 contain these strange words and abbreviations like ‘ml’, ‘nlp’ etc while topics right to PC2 are more complete.

Model 2 and Model 4 share a same problem of lacking clear sematic meaning from unigram. Keywords are extracted in unigram without context, making it hard to understand its underlying meaning. This ambiguity can cause problem interpretating the topics. For instance, keywords from Model 4 like ‘high, ‘low, ‘neural’. High and low could refer the resolution of an image or a dimension of dataset while neural might be referring to psychology neural if the article to relate medicine so it is challenging in interpretating the topics without more context.

To conclude, the training data set is related to Data science with topics like NLP, LLM, neural network, image processing etc. Below are some example topics found in the training dataset:

Record 6 is about road detection using 3D image data, in one sentence of the abstract ‘… the fine-grained segments of visual images are demonstrated.’ include keywords like ‘image’ and ‘visual’. In fact, if filter the abstract to those include ‘image’, 43275 records show out of 152586 (28%). Another example ‘Convolutional Neural Network (CNN) has been successfully applied on classification of both natural images and medical images…’, from Record 8 related to brain MRI using image NN model. The sentence contains topics keywords like ‘image’, ‘network’, ‘classification’.

Topic modelling provide valuable insights in understanding large amount of data such as the underlying topics and the patterns in the textual data. This could be beneficial in identifying trending or hot-topics. It provides a brief summarization among the documents and discover their relationship. The relationship can be visualized with distance map, provide insights of how similar are the topics and potentially identify improvement or relevant area for further exploring.

Although the advantages of LDA topic modelling mentioned, more shortcomings are found in this case.

Firstly, as mentioned previously the unclear semantic meaning of unigrams presents challenge in understanding the actual topics. This ambiguity makes it difficult to interpret the meaning behind individual words within topics, hindering the overall comprehension of the topics.

Additionally, all models show some degree of overlapping topics, particularly as the number of topics increases. This overlapping further complicates the interpretation of topics, as it becomes challenging to distinguish between distinct thematic elements.

Moreover, the frequent occurrence of certain words across topics, such as 'code,' 'network,' 'language,' and 'task,' should be considered as context-dependent stop words. These domain-specific terms appear frequently across topics, causing overlap in the distance map and hinder the ability to effectively group topics. For instance, Topics 10 and 8 in Model 3 are both related to neural networks and deep learning. Despite of the similarity, they are far from each other in the distance map. To solve to this problem, in addition to remove context independent stopwords, context dependent stop word or words appear in most of the articles should also be identified and excluded. By reducing the effect of these high frequency words, the model can focus more on the sematic difference between topics.

Furthermore, data quality affects the model's ability to group topics effectively. The existence of strange keywords in some topics indicates quality issues within the input data. These anomalies suggest that certain topics may be influenced by erroneous or irrelevant data, impacting the accuracy and reliability of the model's output. For instance, in Topic 36 of Model 4, numerous abbreviation and strange keywords such as ‘ri,’ ‘bd,’ and ‘thz’ are included. The inconsistent naming also contribute to overlapping topics. For example nlp is the same as natural language processing but since abbreviation is used, they are treated as 2 different terms.

With data quality problem identified, this explains why Model 3 and 4 perform worse than Model 1 and 2 although having more information input. Models with fewer topics are less affecting by the context dependent stop words and overlapping topics, and hence perform better than model with more topics given the same configuration. This data quality also resonate with the hypothesis of weak performance in RNN in Part 1.

Reference

Chuang,J. et.al (2012). Termite: Visualization Techniques for Assessing Textual Topic Models. *Stanford University Computer Science Department*. <http://vis.stanford.edu/files/2012-Termite-AVI.pdf>

Nath,S (2023). Precision-Recall Trade-Off: Navigating the Delicate Balance in Imbalanced Datasets. *Medium*. <https://medium.com/@sruthy.sn91/precision-recall-trade-off-navigating-the-delicate-balance-in-imbalanced-datasets-66ad175ccf1e>

Scikitlearn (n.d.). Precision-Recall. *Scikitlearn*. <https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html>

Stack Exchange. (n.d.). How to determine the embedded size? *Stack Exchange*. <https://ai.stackexchange.com/questions/28564/how-to-determine-the-embedding-size>

Pickett,M (2020). Exploring Coherence Metrics for Optimizing Topic Models of Humpback Song. *Monterey Bay Aquarium Research Institute*. https://www.mbari.org/wp-content/uploads/Pickett.pdf