COMP 4901K Course Project Report

Team 12

Name:	**Lam Kwun Yuk**	**ID:**	**20512073**	**Email:**	**kylambg@connect.ust.hk**
Name:	Sha Yu Hin	ID:	20516835	Email:	yhsha@connect.ust.hk

1. Methodology

Throughout this project, Python 3 is used as the programming environment. Neural network models are implemented in Keras and notebooks are hosted in Kaggle and Google Colab.

2. Data analysis and Preprocessing

The given dataset contains excerpts of academic articles with topics related to Covid-19. Within the paragraphs, each word is labelled by its entity type. Among all types, the null type ('O') dominates the training set as 69.9% of the word occurrence is of this type. The following figure shows the 20 most frequent types aside 'O' and '_w_pad_', which we observe an exponential decrease in the frequencies.

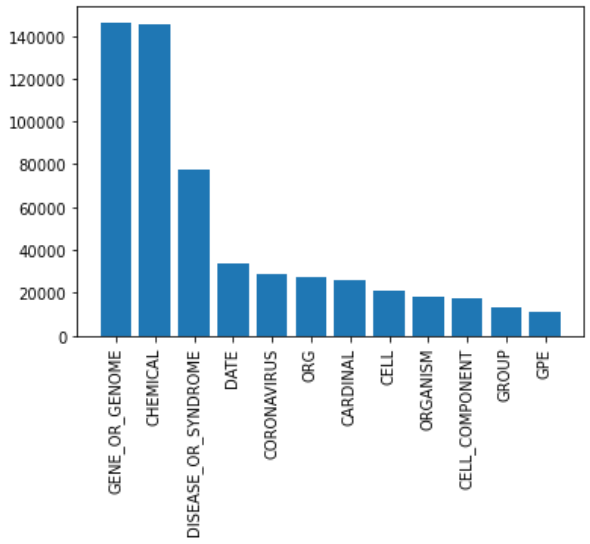

Fig 1: Frequencies of the 12 most frequent tags besides 'O' and '_w_pad_'

The following are some observations we discovered within the dataset:

- Most texts in the corpus are in English, but a few in different languages like French are observed.
- The vocabulary contains 82275, or 67255 after enforcing lower letters only. Among these words, 38275 words are labelled for more than one distinct tag. The word 'and' has the greatest number of tags of 29.

- By comparing the corpus with the stopwords, 80 of the stopwords have multiple labels, and a quarter of them have more than 7 types.
- The labelling of words may be "counter-intuitive", for instance the word "Korea" has been labelled as "CELL", "CHEMICAL", "GENE_OR_GENOME" alongside "ORG".
- Word tags may change within the same excerpt. For example, the phrase "septic shock" is tagged as "GENE_OR_GENOME", "Septic", "DISEASE_OR_SYNDROME" as the term appears in two consecutive lines.

Our team have chosen to stem the words using the Porter stemmer to further reduce the vocabulary size. But stopwords were not removed as this might affect the model accuracy considering it could be classified into many entities.

3. Model Implementation and Evaluation

In this project, we mainly applied neural network including Convolution Neural Network (CNN) and Recurrent Neural Network (RNN). The default number of RNN layer, hidden layer, hidden size and embedding size for RNN are 1, 1, 128 and 64 respectively without mentioning specifically.

3.1 Training Loss and Testing Accuracy of Different Models

Model	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy (basing on accuracy function)
CNN (single convolution layer)	0.4548	0.86245	0.5033	0.8401
LSTM	0.3008	0.9095	0.4516	0.8634
LSTM (Bidirectional)	0.2046	0.9404	0.4170	0.8828
LSTM (Bidirectional) 200 emb size, 256 hidden size	0.1704	0.9506	0.3772	0.8945
GRU (Bidirectional)	0.2046	0.9383	0.3844	0.8908
2 layers GRU (Bidirectional)	0.1658	0.9520	0.3821	0.8962
3 layers GRU (Bidirectional)	0.1567	0.9551	0.4412	0.8956
2 layers GRU (Bidirectional) 2 MLP layers	0.1614	0.9531	0.4037	0.8937
2 layers GRU(Bidirectional) 200 emb size	0.1536	0.9559	0.3875	0.8968
2 layers GRU (Bidirectional) 200 emb size, 128 hidden size	0.1600	0.9075	0.3760	0.8990
2 layers GRU (Bidirectional) 200 emb size, 128 hidden size, 0.2 dropout rate	0.1650	0.9501	0.3768	0.8966
4 layers GRU (Bidirectional), 2 MLP layers 200 emb size, 256 hidden size, 0.2 dropout rate	0.1542	0.9533	0.3581	**0.9902**

Table 1: Training, testing losses and accuracies

(emb stands for the number of nodes in the embedding layer)

From our observations, the performance of RNN is generally better than CNN. The reason can be attributed to that CNN is mainly focused on some specific tokens for classification while RNN is mainly focused on the relationship between tokens within sentences. In the text classification project, the neighbour tokens may have some effects on current token, for instance, “Virus” and “Organism” are usually appeared together. In addition, there may not have some “key” tokens can dominated the semantic meaning of whole sentence hence the performance of CNN is reduced.

Also, it is observed the performance of a more complex model is better than simple model. For instance, 2 GRU layers with 200 embedding size and 128 hidden size contributes to the best model performance. This can be attributed that the number of output classes (65) is relatively large. Therefore, a more complex model can capture more precise features of the sentence for fitting the data. It is remarked that overfitting is not occurred owing to the utilization of early stopping.

4. Conclusion

Based on the validation accuracy, we have chosen the 4-layer GRU model for making the predictions. The specifications of the model are as follows:

- Embedding layers of size 200.
- Layers: 4 bidirectional GRU then 2 MLP layers, all layers have 256 nodes.
- Relu activation is used in all layers.
- Dropout rate in each layer is 0.2.