Báo cáo kết quả thực hiện

Thành viên nhóm:

Nguyễn Lê Anh Quân:19522081

Lý Hoàng Thuận:19522315

Phan Thành Nhân:19521944

1) Mô tả bài toán:

- Phân loại bài báo từ các trang web và tiêu đề của bài báo để xác định bài báo có tính châm biếm (sarcasm) hay không.

-Bài toán thuộc lớp bài toán Text classification.

2) Dataset:

- Data set được lấy từ việc craw từ 3 trang web châm biếm lần lượt là:

+ rochdaleherald.co.uk

+dailysquib.co.uk

+newyorker.com/humor/borowitz-report/

-3 trang báo uy tín:

+Foxbusiness

+NBCNews

+BBCNews

3) Quá trình thực hiện:

a) Clean data:

-Phần data sau khi craw đã được lọc bỏ bớt các tin trùng lặp, loại bỏ các tin là video phỏng vấn (chủ yếu từ trang BBC).

-Một số nhãn được thay đổi do 1 số bài báo của BBC mang tính châm biếm về cả câu chữ lẫn ngữ nghĩa.

- Sử dụng thư viện nltk.stopwords `from nltk.corpus import stopword`, để loại bỏ những từ như A, An , the,… .Những từ này rất nhiều trong Tiếng Anh nhưng không có tác dụng trong việc hỗ trợ việc classification.

-Sử dụng *punkt* một thư viện Tokenizer này chia văn bản thành một danh sách các câu.

- Dataset bao gồm 13938 điểm dữ liệu với 4 feature mỗi điểm và không có điểm dữ liệu nào là NULL.

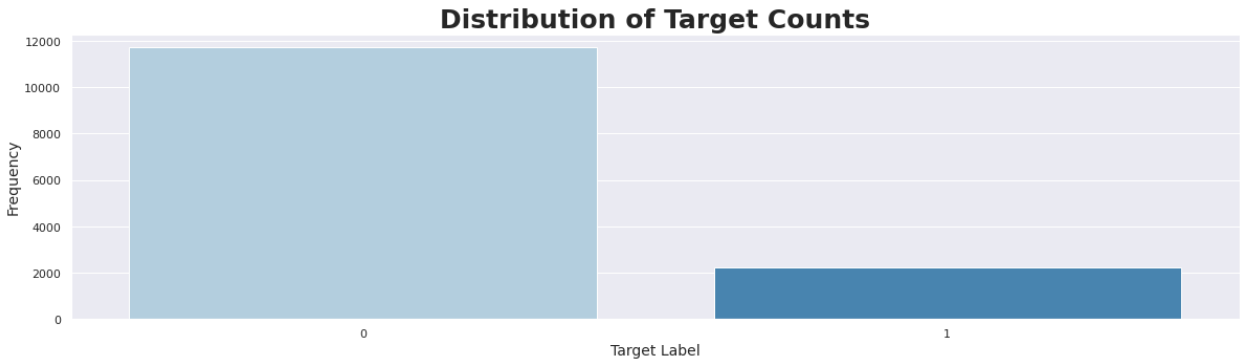
```
headlines.shape

(13938, 4)

[ ]  # count null
     headlines.isnull().sum()

     article_link     0
     headline         0
     news_category    0
     is_sarcastic     0
     dtype: int64
```

b) Preprocessing:

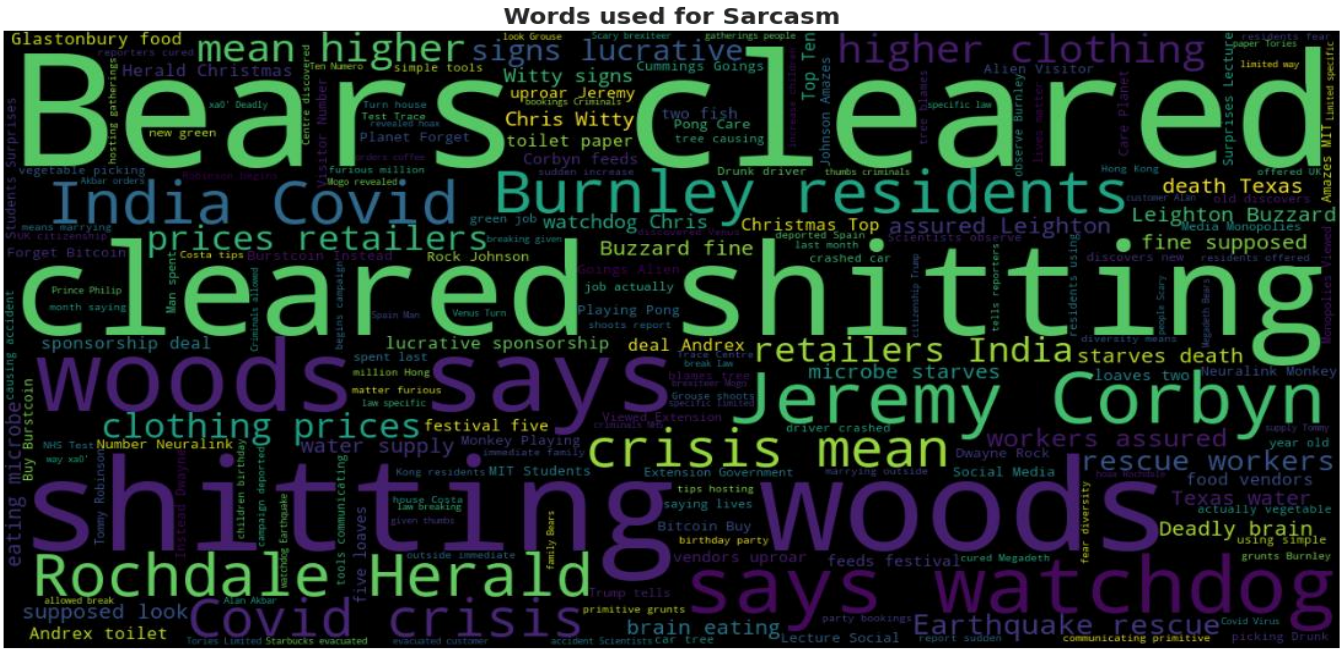
-Sử dụng **GLoVe Embeddings** vì chúng đại diện cho các từ sử dụng ngữ cảnh và học các word vector sao cho tích của chúng bằng logarit của xác suất đồng xuất hiện của các từ.

+Link tham khảo: https://www.kaggle.com/shivam017arora/imdb-sentiment-analysis.

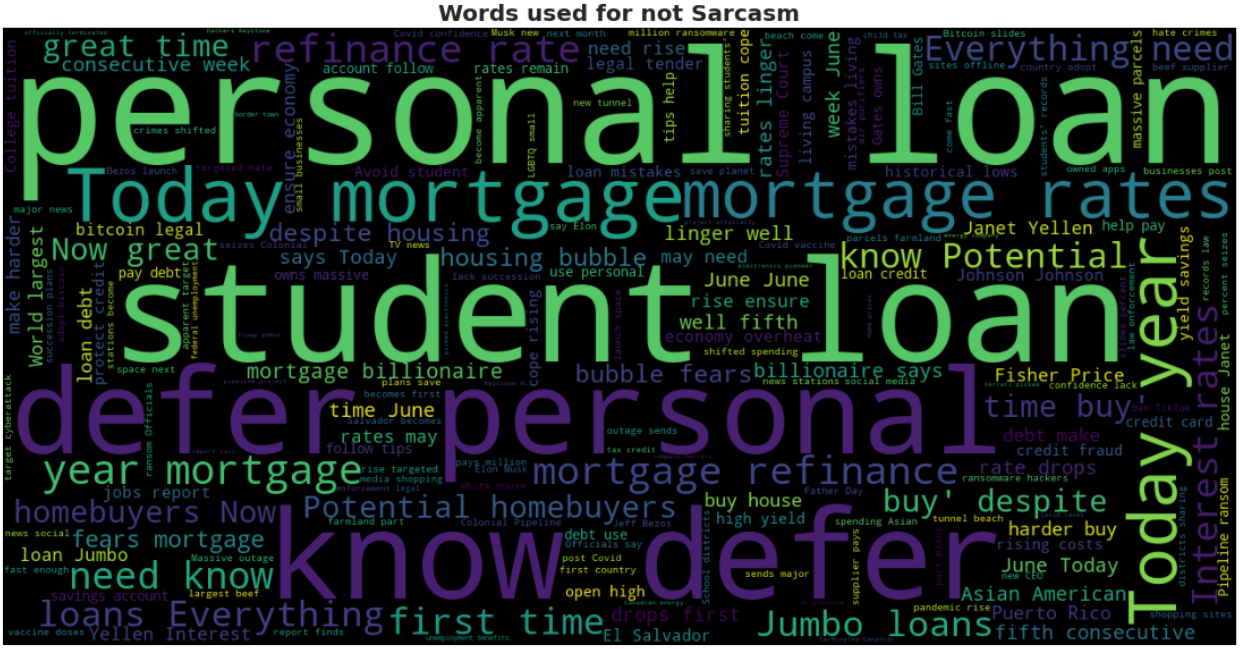
-Splitting data thành train(80%) and test(20%) do chênh lệch class khá lớn.



+Thống kê các từ trong class sarcasm bằng **WordCloud***:



+ Các từ trong class not sarcasm:



++Có thể thấy những từ như student loan, personal loan rate, need know,… sẽ không xuất hiện trong class sarcasm.

**Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.*

-Tạo từ điển của tất cả các từ trong kho ngữ liệu bằng glove embedding (1 kho ngữ liêu được đào tạo trước từ các dữ liệu thu thập từ twitter).

- Ma trận embedding_matrix (nhóm chọn shape 13938 x 100),giúp kiểm soát chiều dài tối đa input.

4) Thực hiện train model:

-Có rất nhiều model có sẵn để giải bài toán text classification, nhưng sau khi nhóm tìm hiểu qua nhiều paper và nhóm quyết định chọn model sử dụng RNN.

-Recurrent Neural Network (RNN) là một trong những kiến trúc nổi tiếng nhất được sử dụng cho các bài toán Natural Language Processsing (NLP) bởi vì cấu trúc lặp lại của nó rất thích hợp để xử lý văn bản có chiều dài thay đổi. RNN có thể sử dụng các biểu diễn phân tán của các từ bằng cách chuyển đổi từng *tokens comprising* của văn bản thành vectơ trước tiên sau đó tạo thành một ma trận.

-Theo bài báo: [The latest in Machine Learning | Papers With Code]

a) Simple RNN:

- Có 3 lớp:

+Embedding với *embedding_matrix* làm tham số, đầu vào 13938, đầu ra có shape (None, 32 ,100).

+Simple_RNN: output shape 64 và dropout=0.1 (giúp tránh việc overfitting).

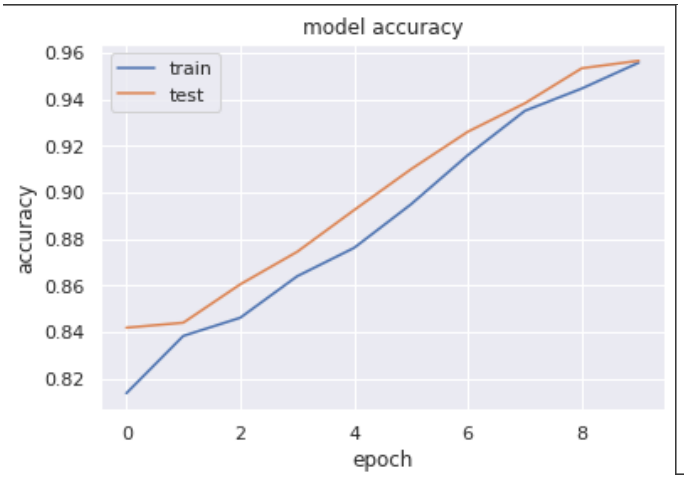
+Và lớp cuối cùng là 1 lớp simple_RNN(activation=’sigmoid’ có thể hiểu lớp này như 1 hàm logistic).

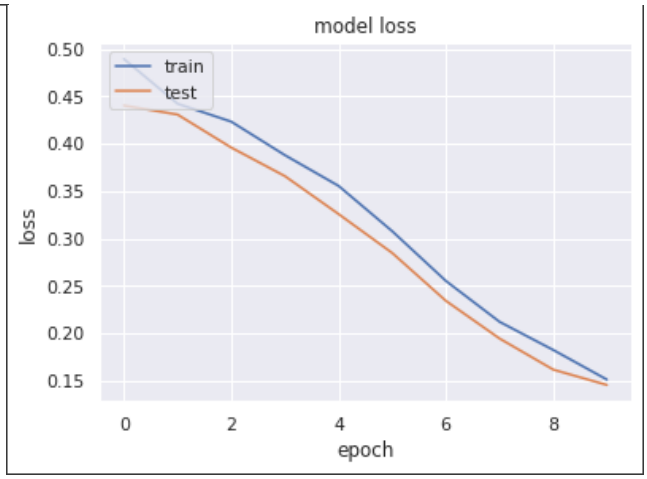
-1 hàm compile để định nghĩa LossFunction, Optimizer và số liệu được chọn là accuracy.

-Kết quả thu được:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	2349
1	0.95	0.76	0.85	439
accuracy			0.96	2788
macro avg	0.95	0.88	0.91	2788
weighted avg	0.96	0.96	0.95	2788

-Bên dưới là 2 biểu đồ thể hiện accuracy và loss của model sau khi tăng số lượng epoch lên đến 10 lần.





b) LSTM:

-Ở simple RNN ta có thể thấy tuy accuracy khá cao nhưng recall của class 1 lại thấp, chỉ rơi vào khoảng 0.76 trong khi class 0 thì 0.99.

-Một mô hình được nhóm lựa chọn là LSTM. Vậy LSTM có gì khác simpleRNN?

-LSTM mang 1 tính chất nổi trội hơn chính là **khắc phục việc vanishing gradient của RNN,** tức nghĩa đôi khi gradient hội tụ tại 1 lượng state nhất định dù chưa backprop toàn bộ mạng.

- Một nhược điểm khác của RNN trên là việc các weight và bias trong RNN được chia sẻ cho nhau => tham số các state giống nhau.

-Các lớp được thế kế khá giống với RNN trừ việc ở layer Dense hàm activate là hàm *ReLU.*

+Do:

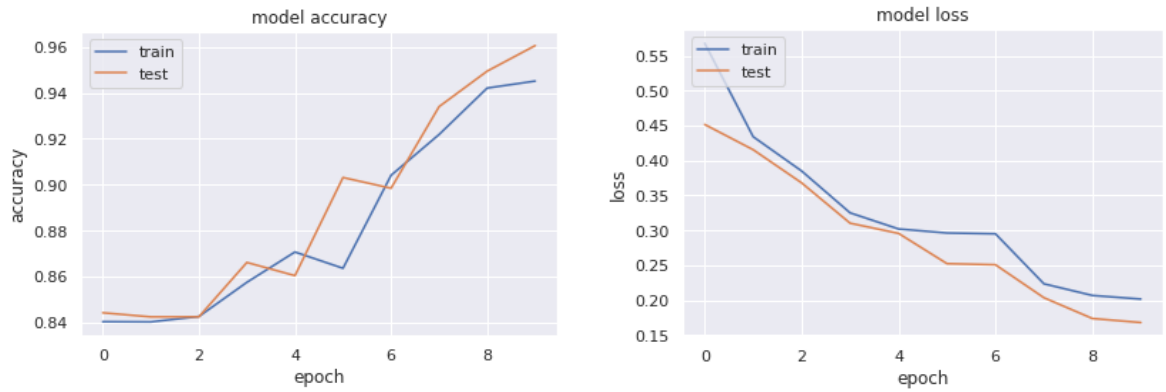
++Tốc độ hội tụ nhanh hơn hẳn

++Tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm *exp* và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.

-Kết quả thu được:

	precision	recall	f1-score	support
0	0.96	0.99	0.98	2349
1	0.94	0.80	0.86	439
accuracy			0.96	2788
macro avg	0.95	0.90	0.92	2788
weighted avg	0.96	0.96	0.96	2788

-Recall trong class 1 tăng lên 0.8.



c) Bidirectional:

-Và mô hình cuối cùng nhóm thử nghiệm là RNN LTSM nhưng có thêm Bidirectional.

- Sơ lược về **Bidirectional** hay **BiLSTM** có nghĩa là **LSTM hai chiều**, có nghĩa có là tín hiệu truyền ngược cũng như chuyển tiếp theo thời gian (2 chiều). So với LSTM, BLSTM hoặc BiLSTM có hai mạng, truy cập thông tin lớp trước đó và trong lớp kế.
- compile trong mô hình này có hàm loss là *binary_crossentropy* , optimizer được chọn là 'rmsprop' (đọc thêm tại: Understanding RMSprop — faster neural network learning | by Vitaly Bushaev | Towards Data Science)
-Kết quả thu được:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2349
1	0.96	0.81	0.88	439
accuracy			0.96	2788
macro avg	0.96	0.90	0.93	2788
weighted avg	0.96	0.96	0.96	2788

-Các chỉ số precision , recall lẫn f1-score được cải thiện khá đáng kể.

Biểu đồ cho accuracy và loss qua 10 epoch:

3) Kết luận:

- Kết quả thu được qua 3 model vẫn tồn tại 1 khuyết điểm khá lớn chính là việc chênh lệch giữa 2 class 0 (not sarcasm) và 1 (sarcasm) từ dataset thu thập. Điều này đã khiến cho model không có được recall giữa 2 class 0 và 1 tương đương nhau.

-Mặc dù dataset không quá lớn và vẫn còn hiện tượng chênh lệch class , nhưng với sự hỗ trợ của **glove embedding** (1 kho ngữ liêu được đào tạo trước từ các dữ liệu thu thập từ twitter), nhóm đã có trung bình 0.96 accuracy, f1-score không quá chênh lệch nhau.

-Những cải tiến khá đáng kể thu được sau khi sử dụng Bidirectional LSTM model so với simpleRNN và LSTM. Các chỉ số f1-score, recall đều được cải thiện.

	precision	recall	f1-score	support
0	0.96	0.99	0.97	2349
1	0.95	0.76	0.85	439
accuracy			0.96	2788
macro avg	0.95	0.88	0.91	2788
weighted avg	0.96	0.96	0.95	2788

Figure 1 SimpleRNN

0	0.96	0.99	0.98	2349
1	0.94	0.80	0.86	439
accuracy			0.96	2788
macro avg	0.95	0.90	0.92	2788
weighted avg	0.96	0.96	0.96	2788

Figure 2 LSTM

0	0.97	0.99	0.98	2349
1	0.96	0.81	0.88	439
accuracy			0.96	2788
macro avg	0.96	0.90	0.93	2788
weighted avg	0.96	0.96	0.96	2788

Figure 3 Bidirectional LSTM