**Phân loại sắc thái bình luận**

Xác định một bình luận là tích cực hay tiêu cực

Các bước thực hiện:

1. Đọc dữ liệu từ file train.crash

2. Tiền xử lý dữ liệu cho dữ liệu đầu vào

* Đưa các ký tự in hoa về chữ thường
* Loại bỏ các con số
* Loại bỏ các dấu câu
* Loại bỏ khoảng trắng thừa
* Loại bỏ các chữ cái giống nhau liên tiếp: ví dụ: “quaaaaaaaaaaaa” sẽ thành “qua”
* Tách từ tiếng Việt tử dụng thư viện underthesea
* Loại bỏ các stopwords tiếng Việt
* Chuẩn hóa từ để tăng độ chính xác cho tập dữ liệu: ví dụ: các từ tks, thanks,…. đều có nghĩa là cảm ơn trong tiếng Việt

3. Vector hóa dữ liệu sau khi đã thực hiện bước tiền xử lý

Vector hóa dữ liệu sử dụng kỹ thuật TF-IDF có trong thư viện sklearn

TF-IDF (Term Frequency – Inverse Document Frequency) được sử dụng để đánh giá tầm quan trọng của một từ trong văn bản.

* TF: tần suất xuất hiện của từ, là số lần từ xuất hiện trong văn bản,

TF(t,d) = số lần từ t xuất hiện trong văn bản d/tổng số từ trong văn bản d

* IDF: dùng để đánh giá mức độ quan trọng của một từ trong văn bản

IDF(t, D) = log\_e(số văn bản trong tập D/số văn bản chứa từ t trong tập D)

TF-IDF = TF x IDF

4. Tiến hành train model sử dụng các mô hình thường gặp như Logistic Regression và MultinomialNB

* Chia tập dữ liệu train/test theo tỷ lệ 80/20
* Sử dụng model Logistic Regression và MultinomialNB
* Đánh giá model sử dụng F1\_score và confusion matrix
* Confusion matrix giúp ta có cái nhìn chi tiết hơn trong quá trình chọn lọc model dựa trên tập dữ liệu có sẵn
* Bài toán này thuộc loại binary classification (có 2 lớp dữ liệu) nên ta sử dụng phương pháp True/False Positive/Negative, ta định nghĩa phương pháp này dựa trên confusion matrix
* Tham khảo: https://machinelearningcoban.com/2017/08/31/evaluation/#-truefalse-positivenegative
* Để đánh giá chất lượng của model ta sử dụng khái niệm F1-score, khái niệm này dựa trên 2 khái niệm khác là Precision và Recall
* Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP)
* Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN)
* F1-score là harmonic mean của precision và recall

5. Train model sử dụng deep learning

* Sử dụng hàm tách từ Tokenizer của thư viện Keras với số lượng từ vựng tối đa là 8000
* Sử dụng kỹ thuật word embedding
* Word embedding là một không gian vector dùng để biểu diễn dữ liệu có khả năng miêu tả được mối liên hệ, sự tương đồng về mặt ngữ nghĩa, văn cảnh (context) của dữ liệu. Không gian này bao gồm nhiều chiều và các từ trong không gian đó mà có cùng văn cảnh hoặc ngữ nghĩa sẽ có vị trí gần nhau
* Ở đây, ta sử dụng Glove word embedding, Glove word embedding là một dự án mã nguồn mở của Stanford nhằm tạo ra các vector biểu diễn cho các từ. Sử dụng các vector được huấn luyện sẵn bởi Glove, các mô hình có thể tận dụng thông tin về mối quan hệ ngữ nghĩa giữa các từ tốt hơn, từ đó có kết quả tốt hơn trong bài toán NLP. Trong bài toán phân loại sắc thái bình luận ở đây, ta sử dụng vector có chiều là 100 (glove.6B.100d.txt)
* Sử dụng mô hình RNN (Recurrent Neural Network: mạng thần kinh hồi quy), cụ thể ở đây sử dụng LSTM layer với hàm kích hoạt là tanh, đầu ra 1 Dense (Fully-connected layer)
  + RNN là mô hình deep learning với ý tưởng sử dụng một bộ nhớ để lưu lại thông tin từ những bước xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại

Công thức hàm tanh:

* Hàm tanh(x) nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1; 1). Cũng như sigmoid, hàm tanh bị bão hòa ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm tanh lại đối xứng qua 0 nên khắc phục được một nhược điểm của sigmoid.

Ta tiếp tục train model với tỉ lệ train/validation = 80/20, với batch\_size = 32, epochs = 10

Notebook: https://colab.research.google.com/drive/16BJVHlsabA8qSMCY0VNJMj2ZHYd-9OQQ#scrollTo=zocqEdE\_PMXB