ỨNG DỤNG DỮ LIỆU LỚN

PROJECT 2: SENTIMENT ANALYSIS

1.Giới thiệu:

1.1 Mô tả đồ án:

Xây dựng mô hình phân tích ý kiến câu.

1.2 Ngôn ngữ:

Python

1.3 Source code:

Link source code:

1. Thu thập dữ liệu
   Crawl_data.ipynb
2. Tiên xử lý dữ liệu:
   Preprocessing.ipynb
3. Huấn luyện
   Vietnamese-Sentiment Analysis.ipynb - Colaboratory (google.com)

1.4 Thông tin nhóm :

MSSV	Họ và tên
18120505	Đào Quốc Phong
18120525	Đoàn Thanh Quang
18120528	Nguyễn Như Quang
18120533	Dương Đoàn Bảo Sơn

2.Nội dung thực hiện:

2.1 Thu thập dữ liệu:

- Sản phẩm mà nhóm em chọn là: Điện thoại di động
- Ngôn ngữ ở đây tụi em chọn là Tiếng Việt.
- Ban đầu nhóm định lấy dữ liệu từ 2 trang đó là thế giới di động và FPT Shop nhưng vì lượng review có hướng tiêu cực quá ít không đủ điều kiện nên nhóm đã crawl thêm bình luận tiêu cực từ trang Shopee.

- Với thế giới di động và Shopee nhóm dùng Selenium và BeautifulSoup để crawl. Còn FPT Shop thì nhóm dùng requests của python để kéo Api của FPT Shop về để lấy dữ liệu. Trong Api của FPT Shop có key bảo mật, nhóm có lên mạng tìm hiểu thì có người đã giải được cái key này bằng cách debug file JS với HTML, sau đó dùng JS để mô phỏng lại quá trình encrypt ra cái key này. Nhóm đã đọc hiểu đoạn code JS này và chuyển đoạn code sang Python để chạy trên file python.

  Tổng số review mà nhóm lấy được là: 45.597 bình luận có phân bổ theo rating như sau:

rating	name	review
1	1971	2674
2	2024	2256
3	3646	4065
4	4479	4479
5	25808	25808

2.2 Xử lý dữ liệu:

1. Từ dữ liệu nhóm dùng Regex bỏ các kí tự đặc biệt và emoji trong bình luận, và tách các bình luận thành câu. Nhóm tự làm mà không dùng thư viện vì thấy thư viện tách câu không tốt lắm.
2. Để loại bỏ các câu không liên quan đến sản phẩm, nhóm list ra các từ thường dùng để nói về shop ví du như: nhân viên, vận chuyển, giao hàng,... để loại bỏ các câu này khỏi review.
3. Nhóm liệt kê danh sách các từ viết tắt để thực hiện chuyển đổi các từ này về dạng đầy đủ.
4. Nhóm dùng công cụ google doc để sửa lỗi chính tả, lỗi đánh máy của các câu.
5. Nhóm dùng số sao để đánh dấu label cho câu.
6. Từ bộ dữ liệu lớn, nhóm đọc lại và chọn ra 6000 câu để làm tập dữ liệu train và 2000 câu test, số lượng label tích cực và tiêu cực là ngang nhau trong từng bộ dữ liệu.

2.3 Word Embeddings:

1. TF-IDF : Dùng TfidfVectorizer của sklearn để chuyển câu thành vector tf-idf

   $tf\text{-}idf(t, d) = tf(t, d) * idf(t)$

   $idf(t) = \log [(1 + n) / (1 + df(t))] + 1$

Với *tf(t,d)* là số lần xuất hiện của từ *t* trong câu *d*, *idf(t)* số lần xuất hiện của từ *t* trong tổng số các câu trong tập dữ liệu. Các hằng số "1" trong *idf(t)* để tránh trường hợp chia cho 0.

2. Word2Vec : Đưa các từ lên không gian vector embedding bằng cách dự đoán "ngữ cảnh" của từ, thể hiện mối quan hệ giữa các từ trong tập từ điển.

   Sử dụng thư viện gensim để pretrained word embedding đưa các từ thành vector 128 chiều

   Sau đó chuyển các câu trong tập dữ liệu thành các ma trận với mỗi từ là 1 vector, ta có tập các ma trận số n x 128 các câu review đưa vào CNN để huấn luyện.

3. PhoBert : Sử dụng pre-trained PhoBert của tranformers và tách từ theo tiếng Việt bằng VnCoreNLP

   Tải pre-trained của PhoBert trong đó bao gồm config của model, pretrained weight của model và từ điển có sẵn của PhoBert

   Với mỗi câu được tách các từ,dùng bpe để chuyển thành các từ rồi ánh xạ vào từ điển để lấy index của nó và câu được thêm <s> ở đầu câu và </s> ở cuối câu

   Ta có tập các list là id của từ trong từ điển giới hạn mỗi câu là 100 những câu ngắn hơn sẽ được gán bằng 0 ở cuối.

   Tiếp theo tạo mask để làm đầu vào cho transformer, mask cho biết giá trị nào được gán bằng 0 ở trên.

   Chuyển tất cả dữ liệu đào tạo đã chuẩn bị trước thành tensor để tạo dataloader. Dùng model đã tải ở trên để train với dữ liệu đã xử lý.

2.4 Xây dựng model:

1. Logistic Regression cho TF-IDF : Dữ liệu khi đã được chuyển thành các vector tf-idf và labels là 0 : NEG và 1 : POS, được đưa vào Logistic Regression để huấn luyện.

2. CNN cho Word2Vec: Dữ liệu bây giờ của 1 câu là một ma trận mà hàng là một từ được vector hóa với số chiều là embedding size = 128, label được chuyển về dạng onehot, khi áp dụng CNN cho W2V ta phải trượt các filter có kích thước dựa vào embedding size của từ để mô hình có thể hiểu hết được ngữ nghĩa của một từ.

2.5 Kết quả đạt được:

6000 câu cho train và validation. 2000 câu cho test.

1. TF-IDF + Logistic Regression : train :75%, val:25%

   Kết quả Val:

	precision	recall	f1-score	support
0	0.83	0.89	0.86	745
1	0.88	0.83	0.85	755
accuracy			0.86	1500
macro avg	0.86	0.86	0.86	1500
weighted avg	0.86	0.86	0.86	1500

   Kết quả khi predict tập test:

	precision	recall	f1-score	support
0	0.84	0.88	0.86	1000
1	0.88	0.83	0.85	1000
accuracy			0.85	2000
macro avg	0.86	0.85	0.85	2000
weighted avg	0.86	0.85	0.85	2000

2. Word2Vec + CNN : train :80%, val :20%

   Kết quả accuracy cuối cùng :train : 0.91, val :0.84

   Kết quả test:

	precision	recall	f1-score	support
0	0.82	0.84	0.83	1000
1	0.84	0.82	0.83	1000
accuracy			0.83	2000
macro avg	0.83	0.83	0.83	2000
weighted avg	0.83	0.83	0.83	2000

3. PhoBert pretrained : train :80%, val :20%

   Kết quả accuracy cuối cùng :
   train:
      acc : 0.99
      f1-score: 0.99
   val :

acc : 0.89

f1-score: 0.89

test:

acc : 0.89

f1-score: 0.89

3.Tài liệu tham khảo:

- 1. Installation — Selenium Python Bindings 2 documentation (selenium-python.readthedocs.io)
- Sentiment analysis - Wikipedia
- sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.1.3 documentation
- BERT, RoBERTa, PhoBERT, BERTweet: Ứng dụng state-of-the-art pre-trained model cho bài toán phân loại văn bản (viblo.asia)
- Code để giải mã key FPT
- https://teamcodedao.com/forum/index.php?/topic/8435-fptshop-co-ban-nao-da-lay-duoc-thong-tin-api-cua-fptshop-chua/