

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH**



**MÔN HỌC
CS221 - XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**BÁO CÁO ĐỒ ÁN
HATE SPEECH DETECTION BASED ON
SENTIMENT KNOWLEDGE SHARING**

Giảng Viên: TS. Nguyễn Thị Quý

Lớp: CS221.O11.KHCL

Nhóm 4:

- | | | |
|---------------------|---|----------|
| 1. Bùi Mạnh Hùng | - | 21522110 |
| 2. Lê Trần Bảo Lợi | - | 21522295 |
| 3. Bùi Đình Quân | - | 21522487 |
| 4. Huỳnh Công Thiện | - | 21522621 |
| 5. Nguyễn Minh Trí | - | 21522706 |

TP.HCM, ngày 26 tháng 12 năm 2023

MỤC LỤC

1	Giới thiệu bài toán	1
1.1	Dataset sử dụng trong bài báo	1
2	Phương pháp thực hiện	1
2.1	Tạo môi trường thực nghiệm	1
2.2	Định nghĩa đường dẫn thư mục gốc	1
2.3	Clone github repository	3
2.4	Cài đặt các thư viện cần thiết	4
2.5	Tải dữ liệu	4
2.6	Chuẩn bị dữ liệu	5
2.7	Huấn luyện mô hình	6
3	Demo	7
4	Chạy thử	8
5	Tài liệu tham khảo	9

1 Giới thiệu bài toán

1.1 Dataset sử dụng trong bài báo

Về giai đoạn chuẩn bị dữ liệu, nhóm thực nghiệm trên hai bộ dataset chính:

- **SemEval2019 task5 (SE)**
- **Davidson dataset (DV).**

Ngoài hai bộ dữ liệu dùng để thực nghiệm trên. Cần thêm 2 bộ dataset phục vụ cho thực hiện bài toán :

- **Sentiment dataset:** bộ dữ liệu được cung cấp cho tác vụ sentiment analysis. Bản gốc dùng để huấn luyện và kiểm thử của bộ dữ liệu được cung cấp sẵn trên **Kaggle**
- Một tập dữ liệu chứa các từ vựng xúc phạm (Dictionary of derogatory words)

Ngoài ra, để đạt kết quả tốt hơn trong tác vụ **word embedding** nhóm tải thêm bộ dữ liệu GloVe (một dự án mã nguồn mở của Stanford nhằm hỗ trợ việc tạo các vector embedding biểu diễn cho từ). Với file **GloVe.txt** bạn có thể tải tại đây **GloVe.txt**. Ngoài ra, bạn có thể tải thêm bộ dữ liệu lớn hơn tại đây **Pickled glove-840B-300d**

2 Phương pháp thực hiện

2.1 Tạo môi trường thực nghiệm

Để tăng tốc độ tính toán nhóm tiến hành thực nghiệm trên google colab với loại runtime là T4-GPU

Ngoài ra, các bạn có thể thay thế cuda bằng cpu trên colab nhưng tốc độ tính toán sẽ chậm hơn. Hoặc thay cuda bằng cpu để có thể chạy trên cục bộ. Phương pháp chạy với cpu sẽ tương tự với gpu. Tuy nhiên, nhóm khuyến khích các bạn sử dụng gpu

2.2 Định nghĩa đường dẫn thư mục gốc

Khi các bạn chạy trên colab và kết nối google drive của mình. Nhóm chúng mình sẽ đặt vị trí của thư mục tại:

```
ROOT_PROJECT = "/content/drive/MyDrive/Project_CS221"
```

Khi đó tất cả các đường dẫn được trình bày dưới đây đều có đường dẫn bắt đầu từ <ROOT_PROJECT>. Khi các bạn thực hiện lại thực nghiệm của nhóm chúng mình, thì các bạn có thể điều chỉnh lại <ROOT_PROJECT> trên Drive của của các bạn

Chi tiết hơn về thư mục của nhóm sẽ được vẽ như hình bên dưới:

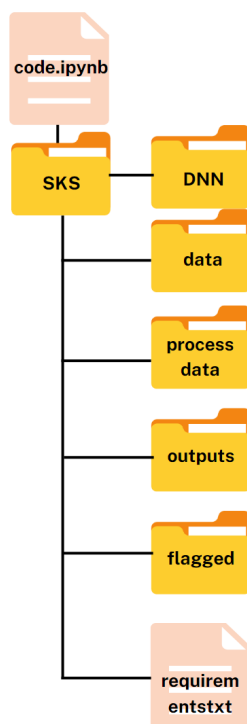


Figure 1: Full folder

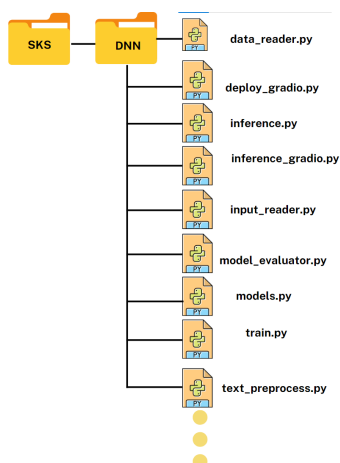


Figure 2: DNN folder

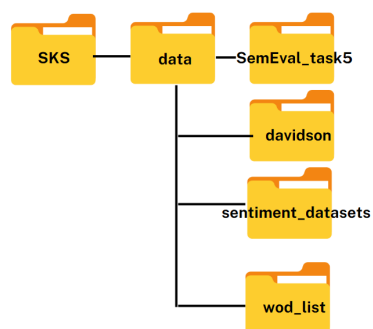


Figure 3: data folder

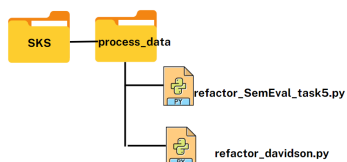


Figure 4: preprocess folder

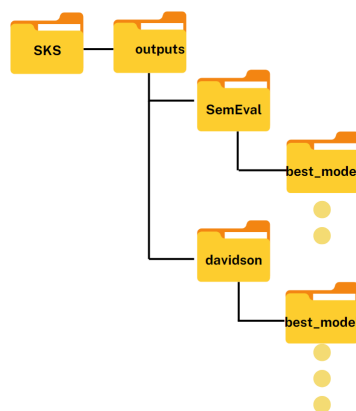


Figure 5: output folder

2.3 Clone github repository

Chúng ta phải clone repo github tại url bên dưới về thư mục <ROOT_PROJECT>:

```
1 %cd {ROOT_PROJECT}
2 !git clone https://github.com/bmhungqb/SKS.git SKS
3
```

bmhungqb hz		1971e72 · 2 hours ago	🕒 37 Commits
📁 .idea	fix bug train davidson	5 days ago	
📁 DNN	hz	2 hours ago	
📁 data	fix refactor dataset davidson + add train kfolds cross validati...	yesterday	
📁 process_data	fix refactor david	5 hours ago	
📄 .gitignore	fix refactor dataset davidson + add train kfolds cross validati...	yesterday	
📄 figure1.jpg	add test	5 days ago	
📄 readme.md	Update readme.md	13 hours ago	
📄 requirements.txt	deploy gradion	5 days ago	

Figure 6: Cấu trúc trang github repository

Trong đó:

- Folder **"DNN"** chứa các file code về đọc dữ liệu, xử lý dữ liệu, kiến trúc mô hình, huấn luyện mô hình, kiểm thử mô hình, triển khai mô hình thành ứng dụng web với gradio,...
- Folder **"data"** chứa các bộ dữ liệu phục vụ cho việc đánh giá, kiểm thử, xử lý dữ liệu text và huấn luyện mô hình. Cụ thể hơn:
 - **SemEval_task5**: Chứa bộ dữ liệu SE dùng cho việc huấn luyện và đánh giá mô hình
 - **Davidson**: Chứa bộ dữ liệu DV dùng cho việc huấn luyện và đánh giá mô hình
 - **word_list**: là danh sách các từ ngữ mang ý nghĩa xúc phạm
 - **sentiment_datasets**: dữ liệu sentiment phục vụ cho việc huấn luyện mô hình
- Folder **"process_data"** chứa các file tiền xử lý dữ liệu cụ thể như sau:
 - **refactor_SemEval_task5**: tác dụng chuyển format file dữ liệu dạng .tsv thành .csv
 - **refactor_davidson**: định dạng dữ liệu DV thành đúng định dạng trước khi đưa vào huấn luyện

Chi tiết hơn bạn có thể truy cập vào đường gấn sau để truy cập vào github của nhóm **Github**

2.4 Cài đặt các thư viện cần thiết

Các thư viện cần thiết được lưu ở file requirements.txt nên chỉ cần dùng câu lệnh như bên dưới, các thư viện sẽ tự động được cài đặt

```
1 %cd {ROOT_PROJECT}
2 !pip install -r {ROOT_PROJECT}/SKS/requirements.txt
3
nltk==3.8.1
numpy==1.26.2
pandas==1.5.3
pyenchant==3.2.2
scikit_learn==1.3.2
scipy==1.11.4
tensorflow==2.15.0
tensorflow_intel
wordninja==2.0.0
langdetect==1.0.9
gradio
```

Figure 7: Nội dung file "requirements.txt"

Tiếp theo, cài đặt một số thư viện cần thiết khác:

```
1 !python -m nltk.downloader 'punkt'
```

Trong thư viện Natural Language Toolkit (NLTK) của Python, "punkt" là một trình dự đoán ngôn ngữ tự nhiên được sử dụng để phân loại văn bản thành các câu

Lưu ý: Khi sử dụng google colab thì khi cài đặt pyenchant sẽ không khai báo được thư viện enchant. Vì vậy thay vì sử dụng lệnh **!pip install pyenchant** ta sẽ sử dụng 2 câu lệnh sau

```
1 !sudo apt-get update
2 !sudo apt-get install python3-enchant -y
3
```

Thư viện enchant: thư viện chứa các từ vựng tiếng anh theo kiểu Anh Anh (UK) và Anh Mỹ (UK)

2.5 Tải dữ liệu

Các bộ dataset sau khi tải về sẽ được lưu trữ trong thư mục: **data** Nếu thư mục **data** chưa có sẵn bạn có thể chạy câu lệnh bên dưới để tạo thư mục

```
1 %cd {ROOT_PROJECT}/'SKS'
```

```
2 !mkdir data
```

Sau đó, lưu các file **SemEval2019 task5 (SE)**, **Davidson dataset (DV)**, **Sentiment analysis dataset**, **GloVe.txt** vào thư mục data

Ngoài cách tải GloVe từ đường dẫn đã cung cấp ở trên. Bạn có thể tải bằng đường dẫn với **!wget** để tải về dữ liệu GloVe dưới dạng .zip

Sau đó, sử dụng lệnh **!unzip** để giải nén dữ liệu

Đối với nhóm, do không đủ dữ liệu nên nhóm sẽ sử dụng tập GloVe6b300d

```
1 %cd {ROOT_PROJECT}/ 'SKS/data'
2 !wget https://huggingface.co/stanfordnlp/glove/resolve/main/glove
  .6B.zip
3 !unzip glove.6B.zip
```

Thêm vào đó, trong repo của paper đã có một danh sách các từ xúc phạm được lưu trong thư mục **SKS/data/word_list** với tên word_all.txt

2.6 Chuẩn bị dữ liệu

Để có thể huấn luyện mô hình, chúng ta cần phải chuẩn bị các bộ dữ liệu cần thiết bằng cách: ở mỗi bộ dữ liệu SE & DV, ta cần chuyển định dạng của file dữ liệu phù hợp với mô hình và chia thành 2 tập test và train

2.6.1 Dataset: SemEval2019 task5

Để chuẩn bị dữ liệu cần thiết cho việc huấn luyện mô hình từ tập **SemEval2019 task5**, ta chạy dòng code bên dưới

Lưu ý: khác với tập dataset DV, tập dataset SE đã chia sẵn tập train, test nên ta không cần chia nữa mà tập trung vào các thao tác chuẩn bị dữ liệu khác

```
1 %cd {ROOT_PROJECT}/ 'SKS'
2 !python process_data/refactor_SemEval_task5.py
```

Sau khi dòng code trên thực thi xong:

1. Kết hợp tập train và dev thành một tập train lớn hơn
2. Dữ liệu sẽ chuyển từ **.tsv** format sang **.csv** format
3. Cột **text** sẽ được đổi tên thành **tweet**, cột **HS** sẽ được đổi tên thành **label**
4. Chọn ra các cột trong DataFrame cần thiết cho quá trình huấn luyện. Cụ thể sẽ là các cột ('tweet', 'label')

5. Lưu dữ liệu từ DataFrame vào .csv file

2.6.2 Dataset: Davidson

Để chuẩn bị dữ liệu cần thiết cho việc huấn luyện mô hình từ tập **Davidson**, ta chạy dòng code bên dưới

```
1 %cd {ROOT_PROJECT}/'SKS'  
2 !python process_data/refactor_davidson.py  
3
```

Khác với dữ liệu **SemEval2019 task5** bộ dữ liệu **Davidson** đã được định dạng ở file .csv nên ta không cần phải chuyển định dạng của dữ liệu mà ta chỉ cần loại bỏ một số cột, đổi tên, tùy theo nhu cầu của bài toán. Cụ thể các bước như sau:

1. Xóa bỏ các cột không cần thiết. Ví dụ 'id', 'count', 'hate_speech', 'offensive_language', 'neither'
2. Từ bộ dữ liệu gốc ban đầu: chia dữ liệu non-hate speech và hate speech ra tập train và test với tỉ lệ 0.8:0.2
3. Kết hợp non-hate speech và hate speech vào tập train. Tương tự, kết hợp non-hate speech và hate speech vào tập test
4. Lưu DataFrame vào file với định dạng .csv

2.7 Huấn luyện mô hình

2.7.1 Dataset: SemEval2019 task5

Để thực hiện quá trình huấn luyện mô hình với tập dữ liệu **SemEval2019 task5**, ta thực hiện đoạn code dưới đây

```
1 %cd {ROOT_PROJECT}/'SKS'  
2 output_path = ROOT_PROJECT+'/'SKS'/outputs/SemEval'  
3 if not os.path.exists(output_path):  
4     os.makedirs(output_path)  
5 !python DNN/train.py -d data/SemEval_task5/df_train.csv --trial  
data/SemEval_task5/df_test.csv -s data/sentiment_datasets/  
train_E6oV3lV.csv --word_list data/word_list/word_all.txt --emb  
data/glove.6B.300d.txt -o outputs/SemEval -b 512 --epochs 50 --lr  
0.002 --maxlen 50 -t HHMM_transformer  
6
```


2.7.2 Dataset: Davidson

Để thực hiện quá trình huấn luyện mô hình với tập dữ liệu Davidson, ta chạy đoạn code dưới đây

```
1 %cd {ROOT_PROJECT}/'SKS'  
2 output_path = ROOT_PROJECT+'/SKS/outputs/davidson'  
3 if not os.path.exists(output_path):  
4     os.makedirs(output_path)  
5 !python DNN/train.py -d data/davidson/train_data.csv --trial data/  
davidson/test_data.csv -s data/sentiment_datasets/train_E6oV3lV.csv  
--word_list data/word_list/word_all.txt --emb data/glove.6B.300d.  
txt -o outputs/davidson -b 512 --epochs 30 --lr 0.002 --maxlen 50 -  
6 t HHMM_transformer --cross_validation
```

Trong đó:

- Dòng 2-4 tạo thư mục để lưu trọng số (weight) khi mô hình huấn luyện xong
- Dòng 5 sử dụng câu lệnh dưới dạng command-line thực thi file python với các tham số được khai báo
- Tiến hành huấn luyện mô hình với các tham số
 - d: đường dẫn đến tập dữ liệu huấn luyện
 - trial: đường dẫn đến tập dữ liệu kiểm thử
 - s: đường dẫn đến tập dữ liệu sentiment
 - word_list: đường dẫn đến file chứa danh sách các từ cần thu
 - emb: đường dẫn đến file word embedding
 - o: đường dẫn đến folder chứa output của quá trình train
 - b: batch size
 - epochs: số lượng epoch cho việc huấn luyện
 - lr: learning rate
 - maxlen: số lượng từ tối đa cho phép trong quá trình huấn luyện
 - t: loại model

3 Demo

Nhóm sử dụng thư viện **Gradio** để thực hiện demo kết quả của từng model trên từng tập dữ liệu

Sau khi huấn luyện xong mô hình và có được file trọng số của mô hình

Để có thể triển khai mô hình thành một thành ứng dụng và cho người dùng(có thể là những người không quan tâm code sẽ chạy như thế nào) có thể kiểm thử một cách trực quan, không phải thông qua command line theo các bước như sau:

Để chạy demo, các bạn cần download file trọng số nhóm đã cung cấp trong drive. Các bạn có thể tải **tại đây**

1. Thay thế đường dẫn đến folder lưu trọng số của mô hình(có hai mô hình chính) như đoạn code được mô tả ở bên dưới(trích trong file `deploy_gradio.py`)

```
1 if __name__ == "__main__":  
2     path_model_SemEval = "outputs/SemEval"  
3     path_model_davidson = "outputs/davidson/final"  
4
```

2. Khi mô hình đưa ra output về dự đoán của một câu text. Dữ liệu đầu ra sẽ là một vector chứa các xác suất về độ tin cậy mà mô hình tin rằng câu text là **hate** hoặc **non-hate**. Vì thế, ta sẽ sử dụng hàm *argmax* để tìm ra vị trí(nhãn) của câu text trên

```
1 %cd {ROOT_PROJECT}/ 'SKS'  
2 !python DNN/deploy_gradio.py  
3
```

Sau khi chạy xong đoạn code trên thì sẽ xuất hiện giao diện website cho người dùng câu text cần kiểm thử

Về tập dữ liệu: ta sẽ chọn một trong hai options **SemEval** hoặc **Davidson**

Sau đó, mình sẽ nhập câu text vào để kiểm thử(bởi vì bộ dữ liệu là tiếng Anh nên sẽ tốt hơn nếu bạn nhập câu text vào bằng tiếng Anh)

4 Chạy thử

Ví dụ: Khi chạy thử với câu text **u look so cute** nghĩa là "*Bạn thật là dễ thương*", vậy với câu text trên thì kết quả mong muốn sẽ là **Non-hate speech**

Để kiểm thử giả thuyết trên nhóm sẽ tiến hành các bước sau:

1. Nhập câu text **u look so cute** vào phần **Input text**
2. Chọn dataset mà mình muốn kiểm tra. Ở đây nhóm chọn SemEval dataset
3. Cuối cùng là nhấn vào nút submit để kiểm tra mô hình(bạn có thể ấn clear để xóa toàn bộ câu text)

Sau khi thực hiện các bước trên, mô hình dự đoán câu text trên là Non-hate speech \Rightarrow có thể thấy mô hình đã dự đoán đúng so với giả thuyết mà ta đã đặt ra

Dưới đây là hình ảnh minh họa cho phần chạy thử bên trên

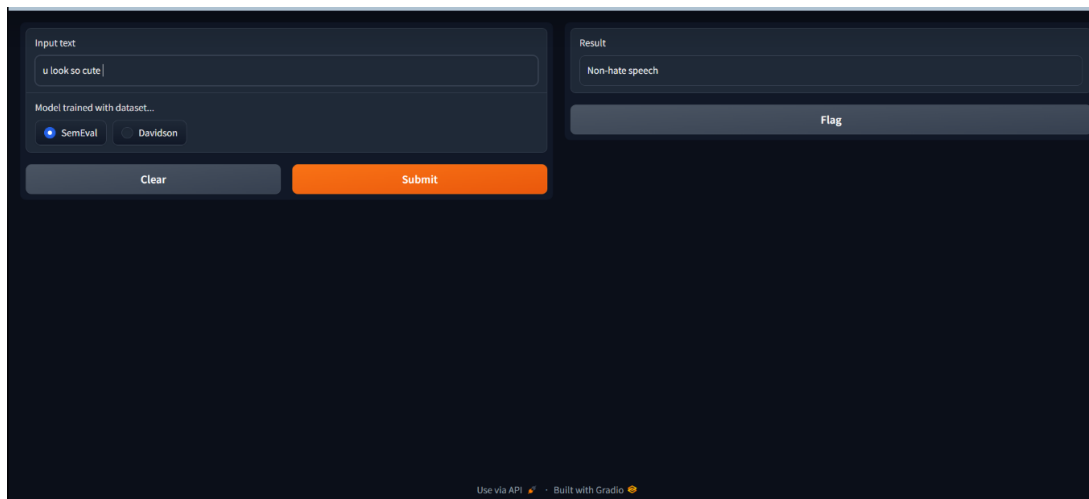


Figure 8: Test

5 Tài liệu tham khảo

Đường dẫn đến bài báo

Đường dẫn đến source code của tác giả