HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

MACHINE LEARNING

Hate Comments Detection on Vietnamese Social Media

ĐINH QUANG HIỂN
20224281

NGUYỄN HUY HOÀNG
20224313

NGUYỄN HOÀNG HIỆP
20224282

HỒ SỸ HIẾU
20224271

INSTRUCTOR: VŨ HẢI

Hanoi, 15/12/2024

TABLE OF CONTENTS

1. Overview

With the rapid growth of the internet, the number of users on social networks has risen dramatically, leading to an exponential increase in data generated from these platforms. Managing user posts and comments has become increasingly challenging. As a result, tools for categorizing posts and comments have become essential. This need was highlighted in the VLSP Shared Task 2019, which introduced the Hate Speech Detection on Social Networks task to classify Vietnamese social media text into predefined categories.

In this task, we focus on a solution for predicting hate speech in Vietnamese which is a low-resource language for natural language processing. In particular, we have implemented deep learning to classify comments or posts on social networks.

The problem is stated as:

· Input: Given a Vietnamese post/comment on social networks.

· Output: One of three labels (**HATE**, **OFFENSIVE**, or **CLEAN**) which is predicted by our system.

The table below shows several examples of this task:

NO.	COMMENT/POST	LABEL
1	mấy thằng ngu này cứ thích sủa	**HATE** (2)
2	buồn gì rồi cũng qua nhưng buồn_ngủ là ngày nào cũng bám đéo tha bạn	**OFFENSIVE** (1)
3	quán này đâu_vậy em cho minh xin dia chỉ di	**CLEAN** (0)

· **HATE** (Hate Speech): a comment or post is identified as hate speech if it (1) targets individuals or groups on the basis of their characteristics; (2) demonstrates a clear intention to incite harm, or to promote hatred; (3) may or may not use offensive or profane words. For example: “Assimilate? No they all need to go back to their own countries. #BanMuslims Sorry if someone disagrees too bad.". See the definition of Zhang et al. [2]. In contrast, “All you perverts (other than me) who posted today, needs to leave the O Board. Dfasdfdasfadfs" is an example of abusive language, which often bears the purpose of insulting individuals or groups, and can include hate speech, derogatory and offensive language.

· **OFFENSIVE** (Offensive but not hate speech): a post or comment MAY contain offensive words but it does not target individuals or groups on the basis of their characteristics. For instance, “WTF, tomorrow is Monday already."

· **CLEAN** (Neither offensive nor hate speech): normal comments or posts on social networks, it does not contain offensive or hate speech. For example, “She learned how to paint very hard when she was young"

2. Methods

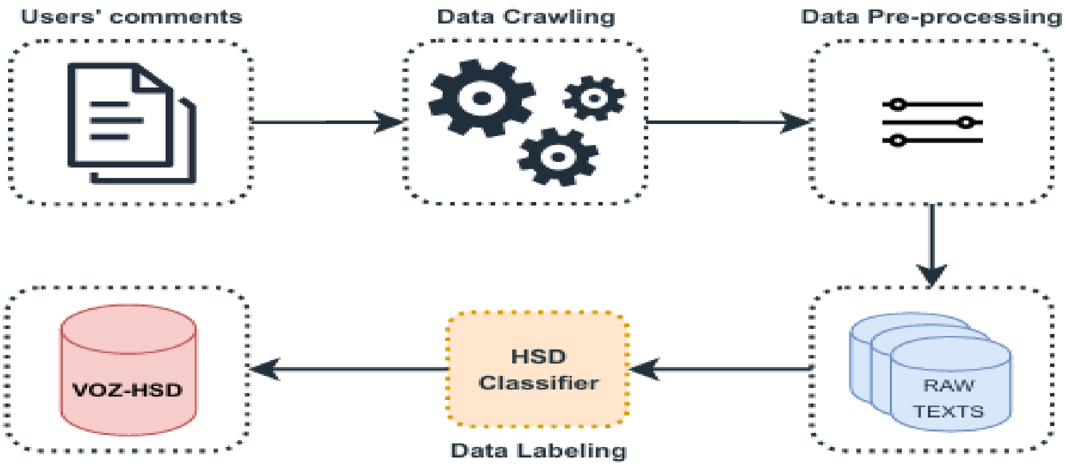
2.1 Enhancing Hate Speech Detection in Vietnamese With a Unified Text-to-Text Transformer Model (Deep-Learning)

https://www.researchgate.net/publication/380820655_ViHateT5_Enhancing_Hate_Speech_Detection_in_Vietnamese_With_A_Unified_Text-to-Text_Transformer_Model

This article does not focus on constructing a prediction model but rather on methodologies for the creation of pre-training data, the pre-training techniques utilized, and the fine-tuning procedures undertaken to assemble the unified VIHATET5 model.

2.1.1 Automated Pre-training Data Creation:

In this research, we present a significant Vietnamese hate speech classification dataset alongside an automated data annotation system



Data crawling: Data was crawled from VOZ forums. The crawling process involved the utilization of the BeautifulSoup4 tool.

Data Pre-processing: The data pre-processing approach, which was outlined by Nguyen et al. (2023), was adopted. This process involves tasks such as eliminating mentioned links, @username, retaining emojis and emoticons, and further excluding quotes.

AI Data Annotator: we initially convert the ViHSD dataset into two labels: CLEAN ⇒ NONE, and (OFFENSIVE, HATE) ⇒ HATE, and employ it as a training dataset for training our classifier. Following this, we fine-tuned several pre-trained models designed for Vietnamese to identify the best-performing ones. ViSoBERT-based fine tuned model was selected as the HSD Classifier.

Automated Data Labeling: Utilizing the selected HSD Classifier, we automatically label all textual data within the raw dataset.

2.1.2 Model Pre-training:

The constructed dataset is employed as the pre-training dataset, comprising samples extracted from real-life comments.

2.1.3 Model Fine-tuning:

To evaluate the trained model, we proceed to fine tune the pre-trained VIHATET5 on various hate speech-based datasets such as Hate Speech Detection (ViHSD), Toxic Speech Detection (ViCTSD) and Hate Spans Detection (ViHOS).

2.1.4 Model setup:

We follow the original pre-training strategy outlined for the T5 model (Raffel et al., 2023) to pre train our VIHATET5. Both training and validation are conducted with a batch size of 128. The pre training process is executed over 20 epochs, employing the Adam optimizer with a lower learning rate set at 5e-3. Additionally, a weight decay of 0.001 is applied, with the initial 2,000 steps designated for warm-up during training. In the fine-tuning phase, we maintain uniform settings for all BERT-based baseline models across specific tasks. Similarly, the same model settings are applied to T5-based models. For detailed information regarding the model settings for fine-tuning downstream tasks, please refer to Appendix B.2. It is worth noting that all experiments are carried out with a limited resource setup utilizing a single NVIDIA A6000 GPU.

2.2 A Large-scale Dataset for Hate Speech Detection on Vietnamese Social Media Texts

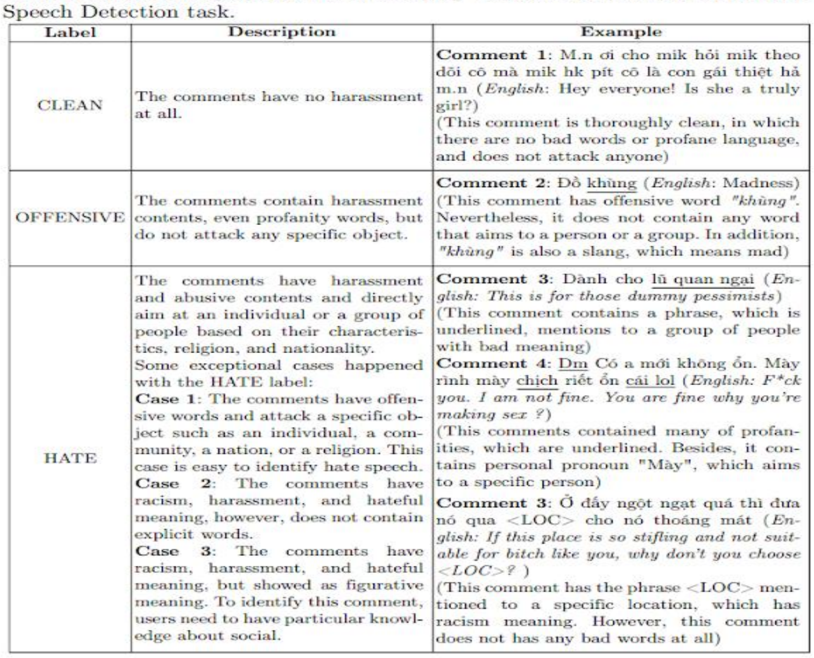
https://arxiv.org/pdf/2103.11528v4

Dataset:

- We collect users’ comments about entertainment, celebrities, social issues, and politics from different Vietnamese Facebook pages and YouTube videos
- Our annotation process contains two main phases:
- The first one is the training phase, which annotators are given a detailed guidelines, and annotate for a sample of data after reading carefully. Then we compute the inter-annotator agreement by Cohen Kappa index (κ) [10]. If the inter-annotators agreement not good enough, we will re-train the annotator, and re-update the annotation guidelines if necessary. After all annotators are well-trained, we go to annotation phase. Our annotation phase is inspired from the IEEE peer review process of articles [15]. Two annotators annotate the entire dataset. If there are any different labels between two annotators, we let the third annotators annotate those labels. The

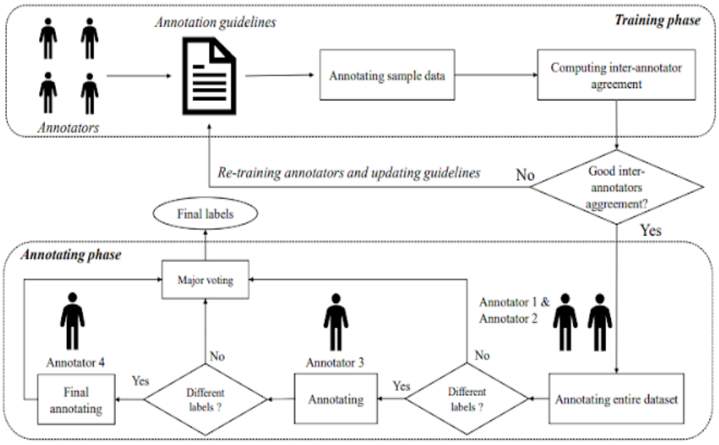
fourth annotators annotate if all three annotators are disagreed. The final label are defined by Major voting.

Table 1: Annotation guidelines for annotating Vietnamese comments in the Hate Speech Detection task.



Label	Description	Example
CLEAN	The comments have no harassment at all.	**Comment 1**: M.n ơi cho mik hỏi mik theo dõi cô mà mik hk pít cô là con gái thiệt hả m.n (*English*: Hey everyone! Is she a truly girl?) (This comment is thoroughly clean, in which there are no bad words or profane language, and does not attack anyone)
OFFENSIVE	The comments contain harassment contents, even profanity words, but do not attack any specific object.	**Comment 2**: Đồ khùng (*English*: Madness) (This comment has offensive word *"khùng"*. Nevertheless, it does not contain any word that aims to a person or a group. In addition, *"khùng"* is also a slang, which means mad)
HATE	The comments have harassment and abusive contents and directly aim at an individual or a group of people based on their characteristics, religion, and nationality. Some exceptional cases happened with the HATE label: **Case 1**: The comments have offensive words and attack a specific object such as an individual, a community, a nation, or a religion. This case is easy to identify hate speech. **Case 2**: The comments have racism, harassment, and hateful meaning, however, does not contain explicit words. **Case 3**: The comments have racism, harassment, and hateful meaning, but showed as figurative meaning. To identify this comment, users need to have particular knowledge about social.	**Comment 3**: Dành cho lũ quan ngại (*English: This is for those dummy pessimists*) (This comment contains a phrase, which is underlined, mentions to a group of people with bad meaning) **Comment 4**: Dm Có a mới không ổn. Mày rình mày chịch riết ổn cái lol (*English: F*ck you. I am not fine. You are fine why you're making sex ?*) (This comments contained many of profanities, which are underlined. Besides, it contains personal pronoun "Mày", which aims to a specific person) **Comment 3**: Ở đấy ngột ngạt quá thì đưa nó qua <LOC> cho nó thoáng mát (*English: If this place is so stifling and not suitable for bitch like you, why don't you choose <LOC>? *) (This comment has the phrase <LOC> mentioned to a specific location, which has racism meaning. However, this comment does not has any bad words at all)

- We randomly take 202 comments from the dataset and give them to four different annotators, denoted as A1, A2, A3 and A4, for annotating. Table 2 shows the inter-annotator agreement between each pair of annotators. Then, we compute the average inter-annotator agreement. The final inter-annotator agreement for the dataset is κ = 0.52.



Based on the ViHSD dataset, two approaches were introduced for constructing prediction models:

- Deep Neural Network Models (DNN Models): Convolutional Neural Network (CNN) can also be applied to Natural Language

Processing (NLP), in which a filter W relevant to a window of h words. Besides, the pre-trained word vectors also influence the performance of the CNN model. On the other hand, Gated Recurrent Unit (GRU) is a variant of the RNN network. It contains two recurrent networks: encoding sequences of texts into a fixed-length word vector representation, and another for decoding word vector representation back to raw input. With the implementation as well as evaluation of the Text-CNN and the GRU models with the fasttext pre-trained word embedding, the embedding transforms a word into a 300 dimension vector.

- Transformer models: The transformer model is a deep neural network architecture based entirely on the attention mechanism, replacing the recurrent layers in auto encoder decoder architectures with special multi-head self-attention layers. Yang et al. [31] found that the transformer blocks improved the performance of the classification model. In this paper, we implement BERT [13]- the SOTA transformer model with multilingual pre-training such as bert-base-multilingual-uncased (m-BERT uncased) and bert-base-multilingual-cased (m-BERT cased), Distil BERT [25]- a lighter but faster variant of BERT model with multilingual cased pre-trained model6, and XML-R- a cross-lingual language model with xlm roberta-base pre-trained . Those multilingual pre-trained models are trained in various languages including Vietnamese.

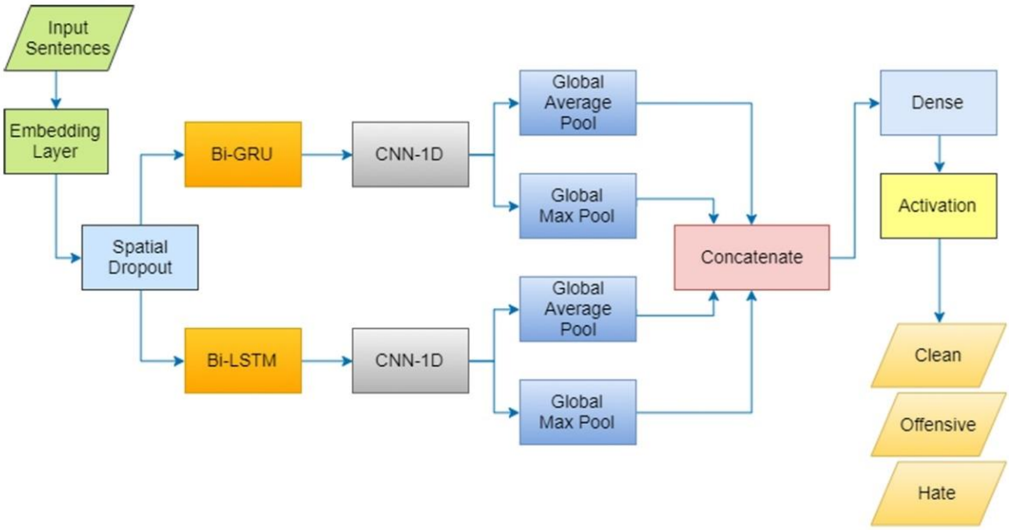
2.3 Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model

The basic architecture in this paper is Convolutional Neural Network (CNN) with 1D convolutions.In addition, we also study about two other deep neural models which are Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). The details of all these neural models are presented in next sub-sections.

In this model, there are several common parts:

- **Word embedding layer**: The input is a matrix of 220x300 dimensions. In particular, each sentence has only 220 words and each word is represented by a 300 dimensional word embedding. Pre-training word level vector already is a kind of word representations for deep neural network models since Word2Vec . In our experiments, we choose FastText as our pre-training model.

- **CNN-1D layer**: We use a 1D spatial drop out with 0.2 dropout rate. It can prevent the model from over-fitting and to get better generalizations.
- **Bidirectional LSTM**: The model uses two parallel blocks of Bidirectional Long Short Term Memory (Bi-LSTM) where the term Bidirectional is that the input sequence is given to the LSTM in two different ways. LSTM is a variation of a recurrent neural network that has an input gate, an output gate, a forget gate and a cell. In our experiment, we used two parallel bidirectional LSTM blocks having 112 units for each. We used sigmoid and tanh for recurrent activations and hidden units respectively.
- **Bidirectional GRU**: Different from LSTMs, gated recurrent units (GRU) is without output gate. In addition, GRUs have an update gate and a reset gate which is responsible of combining new input with the previous one. Finally, the update gate is responsible of how much the previous memory is required to be saved.



(PDF) Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model

3. Implementation using Bi-LSTM-CNN

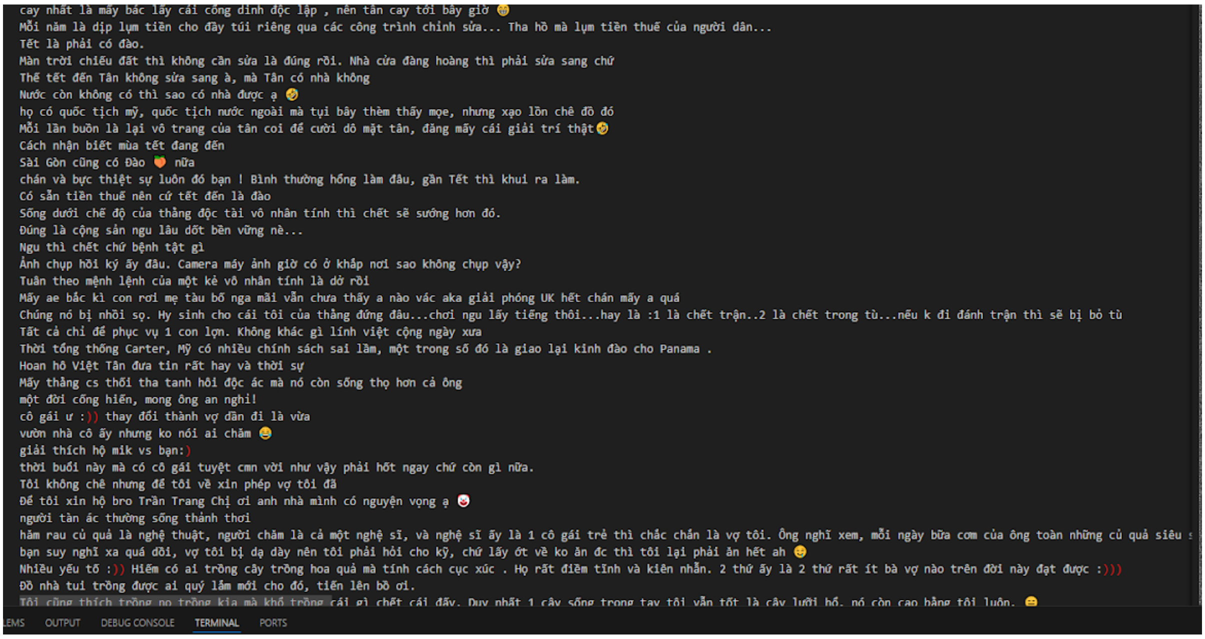
3.1 Dataset Preparation

Our data was crawled from a vast majority of Vietnamese forums, focusing mainly on Facebook and VOZ communities. The result is over 800 comments, some of which contain emojis or special characters.

Total Clean Comments: 607

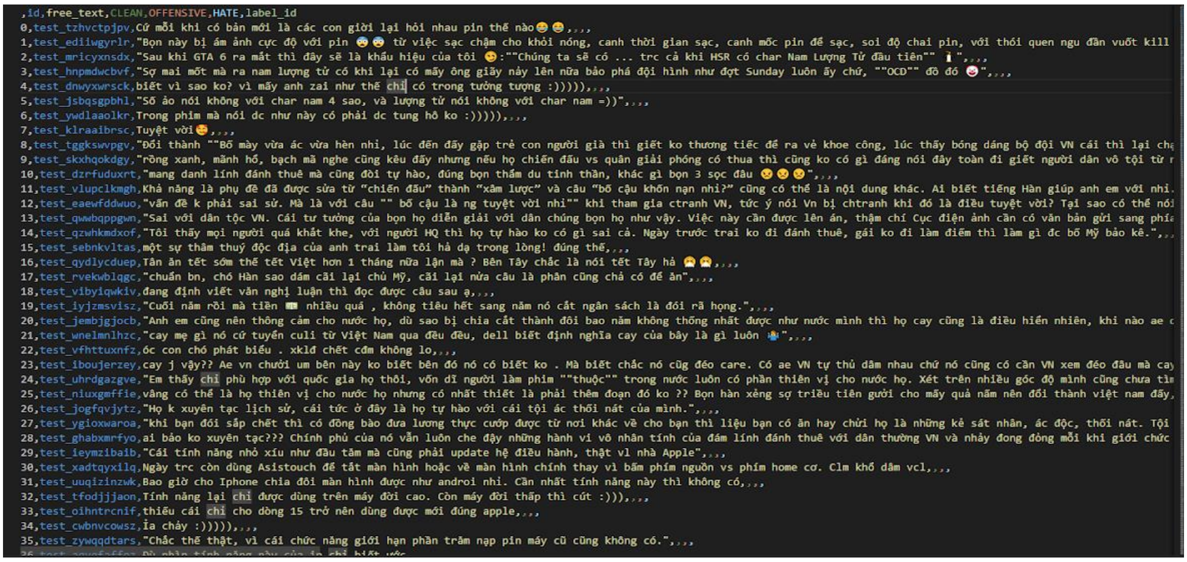
Total Offensive Comments: 150

Total Hate Comments: 107



```
cay nhất là mấy bác lấy cái cổng dinh độc lập , nên tân cay tới bây giờ 😁
Mỗi năm là dịp lụm tiền cho đầy túi riêng qua các công trình chỉnh sửa... Tha hồ mà lụm tiền thuế của người dân...
Tết là phải có đào.
Màn trời chiếu đất thì không cần sửa là đúng rồi. Nhà cửa đàng hoàng thì phải sửa sang chứ
Thế tết đến Tân không sửa sang à, mà Tân có nhà không
Nước còn không có thì sao có nhà được ạ 🤣
họ có quốc tịch mỹ, quốc tịch nước ngoài mà tụi bây thèm thấy mọe, nhưng xạo lồn chê đồ đó
Mỗi lần buồn là lại vô trang của tân coi để cười dô mặt tân, đăng mấy cái giải trí thật🤣
Cách nhận biết mùa tết đang đến
Sài Gòn cũng có Đào 🍑 nữa
chán và bực thiệt sự luôn đó bạn ! Bình thường hổng làm đâu, gần Tết thì khui ra làm.
Có sẵn tiền thuế nên cứ tết đến là đào
Sống dưới chế độ của thằng độc tài vô nhân tính thì chết sẽ sướng hơn đó.
Đúng là cộng sản ngu lâu dốt bền vững nè...
Ngu thì chết chứ bệnh tật gì
Ảnh chụp hồi ký ấy đâu. Camera máy ảnh giờ có ở khắp nơi sao không chụp vậy?
Tuân theo mệnh lệnh của một kẻ vô nhân tính là dở rồi
Mấy ae bắc kì con rơi mẹ tàu bố nga mãi vẫn chưa thấy a nào vác aka giải phóng UK hết chán mấy a quá
Chúng nó bị nhồi sọ. Hy sinh cho cái tôi của thằng đứng đâu...chơi ngu lấy tiếng thôi...hay là :1 là chết trận..2 là chết trong tù...nếu k đi đánh trận thì sẽ bị bỏ tù
Tất cả chỉ để phục vụ 1 con lợn. Không khác gì lính việt cộng ngày xưa
Thời tổng thống Carter, Mỹ có nhiều chính sách sai lầm, một trong số đó là giao lại kinh đào cho Panama .
Hoan hô Việt Tân đưa tin rất hay và thời sự
Mấy thằng cs thối tha tanh hôi độc ác mà nó còn sống thọ hơn cả ông
một đời cống hiến, mong ông an nghỉ!
cô gái ư :)) thay đổi thành vợ dần đi là vừa
vườn nhà cô ấy nhưng ko nói ai chăm 😂
giải thích hộ mik vs bạn:)
thời buổi này mà có cô gái tuyệt cmn vời như vậy phải hốt ngay chứ còn gì nữa.
Tôi không chê nhưng để tôi về xin phép vợ tôi đã
Để tôi xin hộ bro Trần Trang Chị ơi anh nhà mình có nguyện vọng ạ 🤡
người tàn ác thường sống thảnh thơi
hâm rau củ quả là nghệ thuật, người chăm là cả một nghệ sĩ, và nghệ sĩ ấy là 1 cô gái trẻ thì chắc chắn là vợ tôi. Ông nghĩ xem, mỗi ngày bữa cơm của ông toàn những củ quả siêu s
bạn suy nghĩ xa quá đồi, vợ tôi bị dạ dày nên tôi phải hỏi cho kỹ, chứ lấy ớt về ko ăn đc thì tôi lại phải ăn hết ah 😊
Nhiều yếu tố :)) Hiếm có ai trồng cây trồng hoa quả mà tính cách cục xúc . Họ rất điềm tĩnh và kiên nhẫn. 2 thứ ấy là 2 thứ rất ít bà vợ nào trên đời này đạt được :)))
Đồ nhà tui trồng được ai quý lắm mới cho đó, tiến lên bồ ơi.
Tôi cũng thích trồng no trồng kia mà khó trồng cái gì chết cái đấy. Duy nhất 1 cây sống trong tay tôi vẫn tốt là cây lưỡi hổ. nó còn cao bằng tôi luôn. 😑
```

Afterwards, we used idlabel4.py to give each comment an ID as well as transform them into the correct format to preprocess. This is the result:

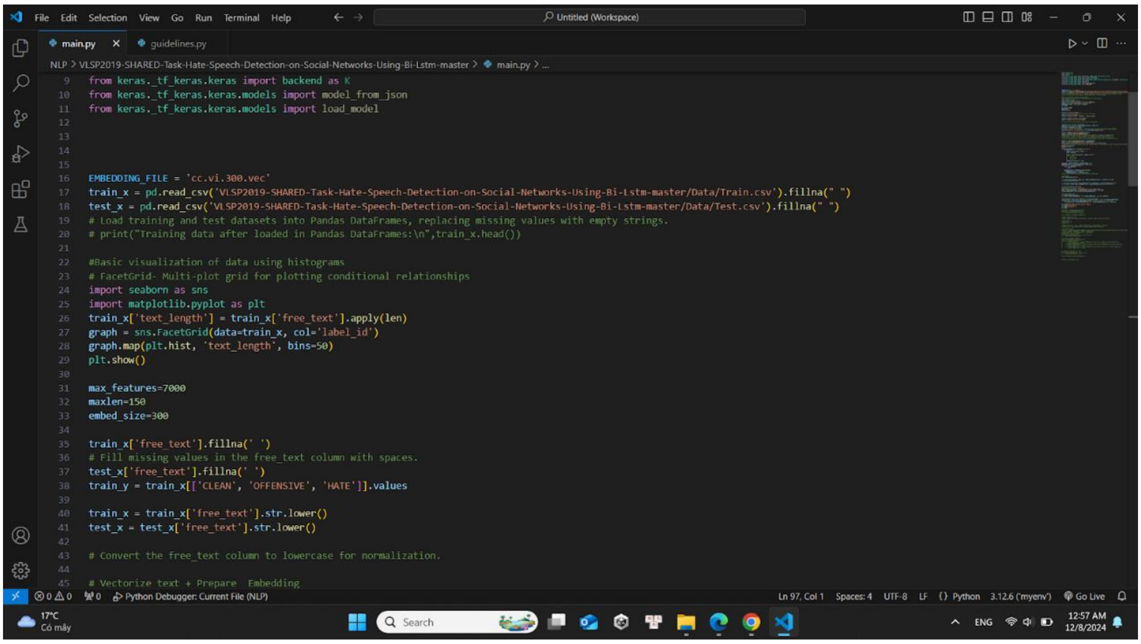


```
,id,free_text,CLEAN,OFFENSIVE,HATE,label_id
0,test_tzhvctpjpv,Cứ mỗi khi có bản mới là các con giời lại hỏi nhau pin thế nào😂😂,,,,
1,test_ediiwgyrlr,"Bọn này bị ám ảnh cực độ với pin 😨😨 từ việc sạc chậm cho khỏi nóng, canh thời gian sạc, canh mốc pin để sạc, soi độ chai pin, với thói quen ngu đần vuốt kill
2,test_mricyxnsdx,"Sau khi GTA 6 ra mắt thì đây sẽ là khẩu hiệu của tôi 😏:""Chúng ta sẽ có ... trc cả khi HSR có char Nam Lượng Tử đầu tiên"" 🕯",,,,
3,test_hnpmdwcbvf,"Sợ mai mốt mà ra nam lượng tử có khi lại có mấy ông giãy nảy lên nữa bảo phá đội hình như đợt Sunday luôn ấy chứ, ""OCD"" đồ đó 🤡",,,,
4,test_dnwyxwrsck,biết vì sao ko? vì mấy anh zai như thế chỉ có trong tưởng tượng :))))),,,,
5,test_jsbqsgpbhl,"Số ảo nói không với char nam 4 sao, và lượng tử nói không với char nam =))",,,,
6,test_ywdlaaolkr,Trong phim mà nói dc như này có phải dc tung hô ko :))))),,,,
7,test_klraaibrsc,Tuyệt vời🥰,,,,
8,test_tggkswvpgv,"Đổi thành ""Bố mày vừa ác vừa hèn nhỉ, lúc đến đấy gặp trẻ con người già thì giết ko thương tiếc để ra vẻ khoe công, lúc thấy bóng dáng bộ đội VN cái thì lại chạ
9,test_skxhqokdgy,"rồng xanh, mãnh hổ, bạch mã nghe cũng kêu đấy nhưng nếu họ chiến đấu vs quân giải phóng có thua thì cũng ko có gì đáng nói đây toàn đi giết người dân vô tội từ r
10,test_dzrfuduxrt,"mang danh lính đánh thuê mà cũng đòi tự hào, đúng bọn thẩm du tinh thần, khác gì bọn 3 sọc đâu 😠😠😠",,,,
11,test_vlupclkmgh,Khả năng là phụ đề đã được sửa từ “chiến đấu” thành “xâm lược” và câu “bố cậu khốn nạn nhỉ?” cũng có thể là nội dung khác. Ai biết tiếng Hàn giúp anh em với nhỉ.
12,test_eaewfddwuo,"vấn đề k phải sai sử. Mà là với câu "" bố cậu là ng tuyệt vời nhỉ"" khi tham gia ctranh VN, tức ý nói Vn bị chtranh khi đó là điều tuyệt vời? Tại sao có thể nói
13,test_qwwbqppgwn,"Sai với dân tộc VN. Cái tư tưởng của bọn họ diễn giải với dân chúng bọn họ như vậy. Việc này cần được lên án, thậm chí Cục điện ảnh cần có văn bản gửi sang phía
14,test_qzwhkmdxof,"Tôi thấy mọi người quá khắt khe, với người HQ thì họ tự hào ko có gì sai cả. Ngày trước trai ko đi đánh thuê, gái ko đi làm điếm thì làm gì đc bố Mỹ bảo kê.",,,
15,test_sebnkvltas,một sự thâm thuý độc địa của anh trai làm tôi hả dạ trong lòng! đúng thế,,,,
16,test_qydlycduep,Tân ăn tết sớm thế tết Việt hơn 1 tháng nữa lận mà ? Bên Tây chắc là nói tết Tây hả 😤😤,,,,
17,test_rvekwblqgc,"chuẩn bn, chó Hàn sao dám cãi lại chủ Mỹ, cãi lại nửa câu là phần cũng chả có để ăn",,,,
18,test_vibyiqwkiv,đang định viết văn nghị luận thì đọc được câu sau ạ,,,,
19,test_iyjzmsvisz,"Cuối năm rồi mà tiền 💵 nhiều quá , không tiêu hết sang năm nó cắt ngân sách là đói rã họng.",,,,
20,test_jembjgjocb,"Anh em cũng nên thông cảm cho nước họ, dù sao bị chia cắt thành đôi bao năm không thống nhất được như nước mình thì họ cay cũng là điều hiển nhiên, khi nào ae c
21,test_wnelmnlhzc,"cay mẹ gì nó cứ tuyển culi từ Việt Nam qua đều đều, dell biết định nghĩa cay của bây là gì luôn 🤷",,,,
22,test_vfhttuxnfz,óc con chó phát biểu . xklđ chết cđm không lo,,,,
23,test_iboujerzey,cay j vậy?? Ae vn chưởi um bên này ko biết bên đó nó có biết ko . Mà biết chắc nó cũg đéo care. Có ae VN tự thủ dâm nhau chứ nó cũng có cần VN xem đéo đâu mà cay
24,test_uhrdgazgve,"Em thấy chỉ phù hợp với quốc gia họ thôi, vốn dĩ người làm phim ""thuộc"" trong nước luôn có phần thiên vị cho nước họ. Xét trên nhiều góc độ mình cũng chưa tìm
25,test_niuxgmffie,vâng có thể là họ thiên vị cho nước họ nhưng có nhất thiết là phải thêm đoạn đó ko ?? Bọn hàn xẻng sợ triều tiên gưởi cho mấy quả nấm nên đổi thành việt nam đấy,
26,test_jogfqvjytz,"Họ k xuyên tạc lịch sử, cái tức ở đây là họ tự hào với cái tội ác thối nát của mình.",,,,
27,test_ygioxwaroa,"khi bạn đói sắp chết thì có đồng bào đưa lương thực cướp được từ nơi khác về cho bạn thì liệu bạn có ăn hay chửi họ là những kẻ sát nhân, ác độc, thối nát. Tội
28,test_ghabxmrfyo,ai bảo ko xuyên tạc??? Chính phủ của nó vẫn luôn che đậy những hành vi vô nhân tính của đám lính đánh thuê với dân thường VN và nhảy đong đỏng mỗi khi giới chức
29,test_ieymzibaib,"Cái tính năng nhỏ xíu như đầu tăm mà cũng phải update hệ điều hành, thật vl nhà Apple",,,,
30,test_xadtqyxilq,Ngày trc còn dùng Asistouch để tắt màn hình hoặc về màn hình chính thay vì bấm phím nguồn vs phím home cơ. Clm khổ dâm vcl,,,,
31,test_uuqizinzwk,Bao giờ cho Iphone chia đôi màn hình được như androi nhỉ. Cần nhất tính năng này thì không có,,,,
32,test_tfodjjjaon,Tính năng lại chỉ được dùng trên máy đời cao. Còn máy đời thấp thì cút :))),,,,
33,test_oihntrcnif,thiếu cái chỉ cho dòng 15 trở nên dùng được mới đúng apple,,,,
34,test_cwbnvcowsz,Ỉa chảy :))))),,,,
35,test_zywqqdtars,"Chắc thế thật, vì cái chức năng giới hạn phần trăm nạp pin máy cũ cũng không có.",,,,
36,test_aquofaffoz,Đủ nhận tính năng này của ip chỉ biết ước
```

We also did the same procedure for another 100 comments but they were used for evaluating the model and the dataset instead.

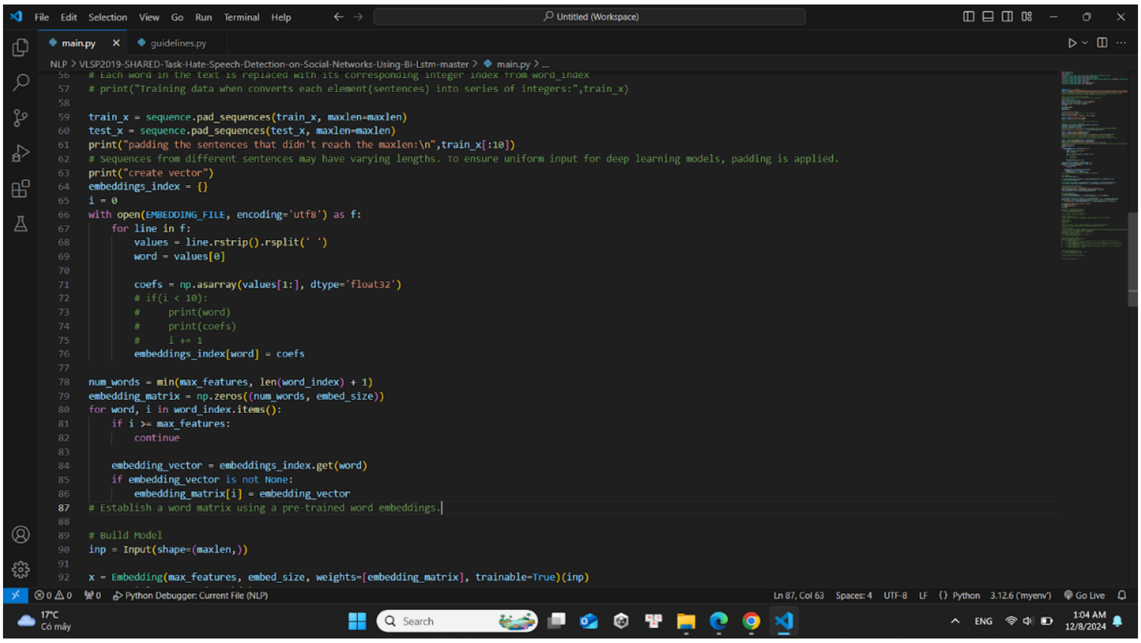
 We use several simple techniques in text pre-processing in all models for this task as follows:

- **Converting all words to lower case.**
- **Removing extra white spaces, punctuation marks.**
- **Replacing all numbers with "number".**
- **Word tokenization using the pivy library**



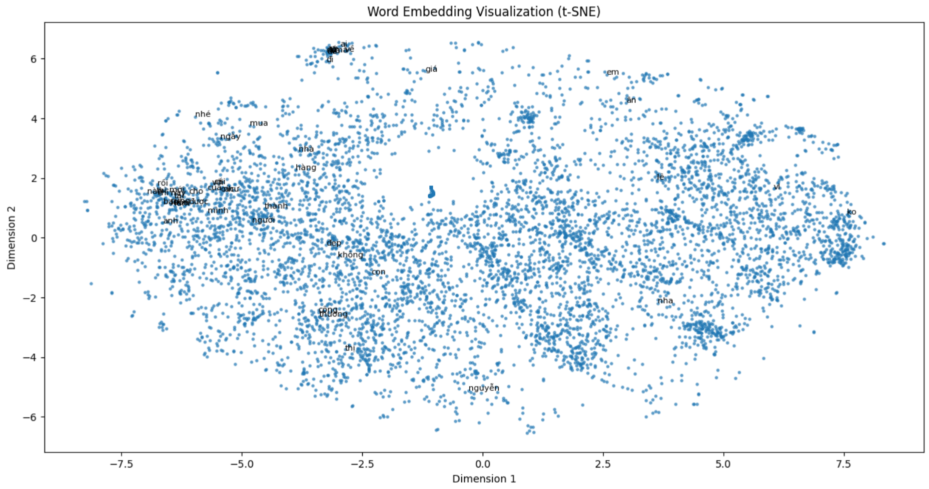
```python
from keras._tf_keras.keras import backend as K
from keras._tf_keras.keras.models import model_from_json
from keras._tf_keras.keras.models import load_model




EMBEDDING_FILE = 'cc.vi.300.vec'
train_x = pd.read_csv('VLSP2019-SHARED-Task-Hate-Speech-Detection-on-Social-Networks-Using-Bi-Lstm-master/Data/Train.csv').fillna(" ")
test_x = pd.read_csv('VLSP2019-SHARED-Task-Hate-Speech-Detection-on-Social-Networks-Using-Bi-Lstm-master/Data/Test.csv').fillna(" ")
# Load training and test datasets into Pandas DataFrames, replacing missing values with empty strings.
# print("Training data after loaded in Pandas DataFrames:\n",train_x.head())

#Basic visualization of data using histograms
# FacetGrid- Multi-plot grid for plotting conditional relationships
import seaborn as sns
import matplotlib.pyplot as plt
train_x['text_length'] = train_x['free_text'].apply(len)
graph = sns.FacetGrid(data=train_x, col='label_id')
graph.map(plt.hist, 'text_length', bins=50)
plt.show()

max_features=7000
maxlen=150
embed_size=300

train_x['free_text'].fillna(' ')
# Fill missing values in the free_text column with spaces.
test_x['free_text'].fillna(' ')
train_y = train_x[['CLEAN', 'OFFENSIVE', 'HATE']].values

train_x = train_x['free_text'].str.lower()
test_x = test_x['free_text'].str.lower()

# Convert the free_text column to lowercase for normalization.

# Vectorize text + Prepare  Embedding
```

After that, we would vectorize all the sentences by assigning every word with a unique integer and prepare our word embeddings which basically is start retrieving the vector values of the pre-trained word-embeddings and assign it to all the words that was found in the dataset.



```python
# Each word in the text is replaced with its corresponding integer index from word_index
# print("Training data when converts each element(sentences) into series of integers:",train_x)

train_x = sequence.pad_sequences(train_x, maxlen=maxlen)
test_x = sequence.pad_sequences(test_x, maxlen=maxlen)
print("padding the sentences that didn't reach the maxlen:\n",train_x[:10])
# Sequences from different sentences may have varying lengths. To ensure uniform input for deep learning models, padding is applied.
print("create vector")
embeddings_index = {}
i = 0
with open(EMBEDDING_FILE, encoding='utf8') as f:
    for line in f:
        values = line.rstrip().rsplit(' ')
        word = values[0]

        coefs = np.asarray(values[1:], dtype='float32')
        # if(i < 10):
        #     print(word)
        #     print(coefs)
        #     i += 1
        embeddings_index[word] = coefs

num_words = min(max_features, len(word_index) + 1)
embedding_matrix = np.zeros((num_words, embed_size))
for word, i in word_index.items():
    if i >= max_features:
        continue

    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
# Establish a word matrix using a pre-trained word embeddings.

# Build Model
inp = Input(shape=(maxlen,))

x = Embedding(max_features, embed_size, weights=[embedding_matrix], trainable=True)(inp)
```

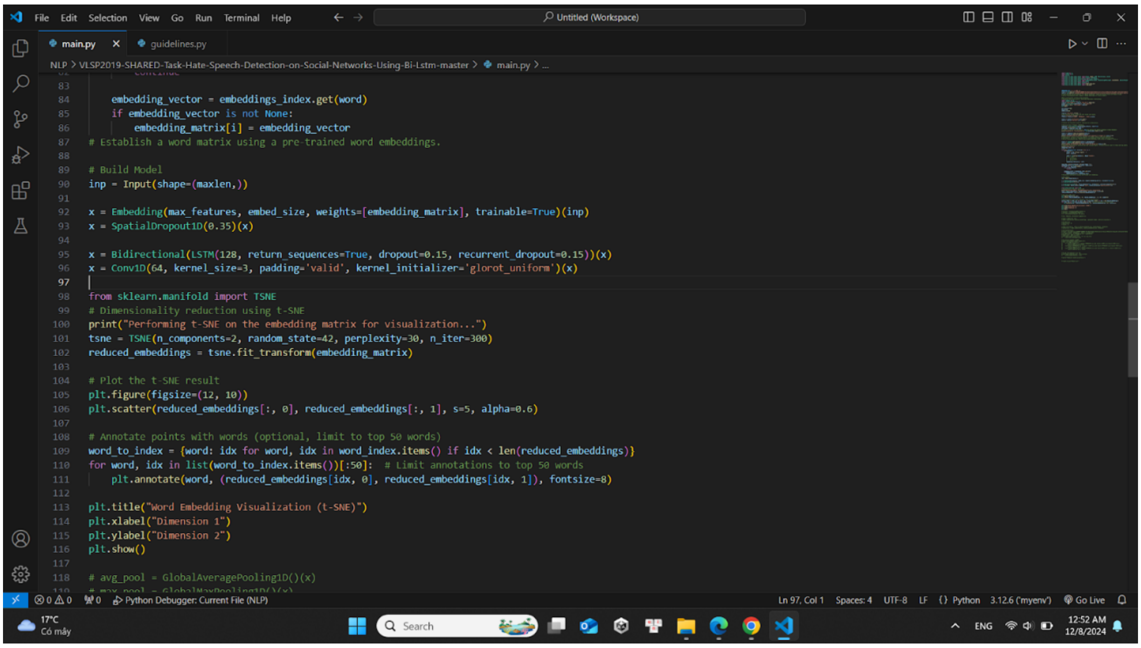
Word Matrix Visualization:



3.2 Model Creation

Now we setup the model:

- Embedding: Initializing the embedding layer with the weights we retrieved from the pre-trained embedding_matrix (cc.vi.300.vec)
- SpatialDropout1D: Randomly drops 35% of embed_size (300) to avoid overfitting.
- Bi-LSTM: applies a bidirectional wrapper around an LSTM layer to capture dependencies in both forward and backward directions.
- Conv1D: applies filters(Kernel) to detect patterns in the sequence of embeddings, it aims to generalize the model further so that it can accurately predicts dataset with similar patterns.


File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
main.py
guidelines.py
NLP > VLSP2019-SHARED-Task-Hate-Speech-Detection-on-Social-Networks-Using-Bi-Lstm-master > main.py > ...
⊗ 0 ⚠ 0
0
Python Debugger: Current File (NLP)
Ln 97, Col 1
Spaces: 4
UTF-8
LF
{} Python
3.12.6 ('myenv')
Go Live
17°C
Có mây
Search
ENG
12:52 AM
12/8/2024

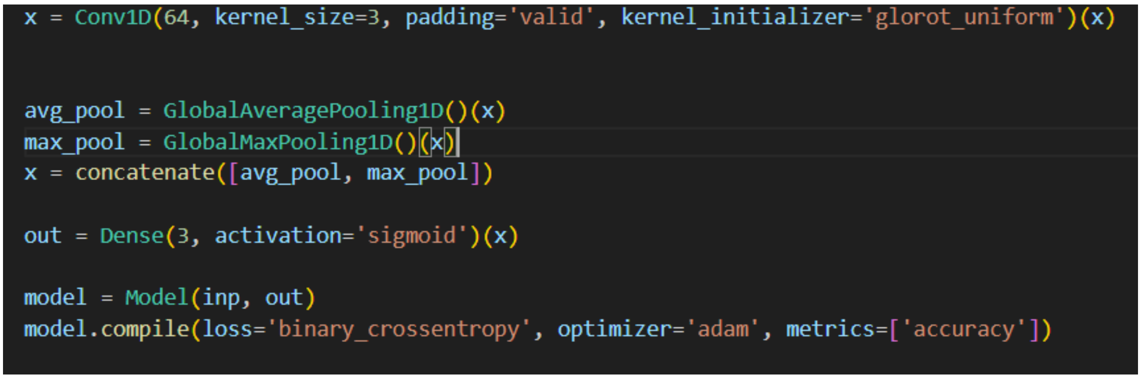
```
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
# Establish a word matrix using a pre-trained word embeddings.

# Build Model
inp = Input(shape=(maxlen,))

x = Embedding(max_features, embed_size, weights=[embedding_matrix], trainable=True)(inp)
x = SpatialDropout1D(0.35)(x)

x = Bidirectional(LSTM(128, return_sequences=True, dropout=0.15, recurrent_dropout=0.15))(x)
x = Conv1D(64, kernel_size=3, padding='valid', kernel_initializer='glorot_uniform')(x)

from sklearn.manifold import TSNE
# Dimensionality reduction using t-SNE
print("Performing t-SNE on the embedding matrix for visualization...")
tsne = TSNE(n_components=2, random_state=42, perplexity=30, n_iter=300)
reduced_embeddings = tsne.fit_transform(embedding_matrix)

# Plot the t-SNE result
plt.figure(figsize=(12, 10))
plt.scatter(reduced_embeddings[:, 0], reduced_embeddings[:, 1], s=5, alpha=0.6)

# Annotate points with words (optional, limit to top 50 words)
word_to_index = {word: idx for word, idx in word_index.items() if idx < len(reduced_embeddings)}
for word, idx in list(word_to_index.items())[:50]:  # Limit annotations to top 50 words
    plt.annotate(word, (reduced_embeddings[idx, 0], reduced_embeddings[idx, 1]), fontsize=8)

plt.title("Word Embedding Visualization (t-SNE)")
plt.xlabel("Dimension 1")
plt.ylabel("Dimension 2")
plt.show()

# avg_pool = GlobalAveragePooling1D()(x)
```

- GlobalAveragePooling1D: acts as a layer in neural networks used for reducing the dimensionality of data while retaining the most important features. It operates on 1D sequence data (like time series or text data represented as sequences).

- GlobalMaxPooling1D: also a layer as a layer in neural networks much like global average but instead of selecting the average values of each sentences, it only retrieve the maximum values.

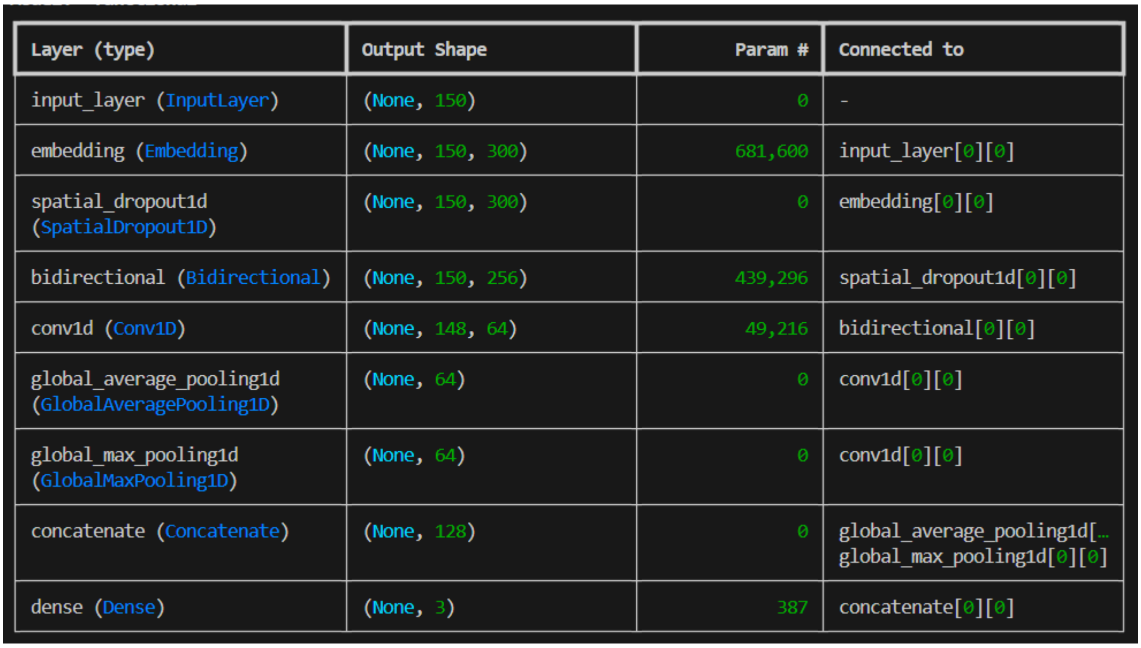
- Concatenate: Combine the results of both GlobalAveragePooling and GlobalMaxPooling.

After that, we convert the output by converting the dense vectors into values from 0-1 through an activation function(sigmoid). Finally we configure the model to compile it.



```
x = Conv1D(64, kernel_size=3, padding='valid', kernel_initializer='glorot_uniform')(x)


avg_pool = GlobalAveragePooling1D()(x)
max_pool = GlobalMaxPooling1D()(x)
x = concatenate([avg_pool, max_pool])

out = Dense(3, activation='sigmoid')(x)

model = Model(inp, out)
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Here’s our model summary:



Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 150)	0	-
embedding (Embedding)	(None, 150, 300)	681,600	input_layer[0][0]
spatial_dropout1d (SpatialDropout1D)	(None, 150, 300)	0	embedding[0][0]
bidirectional (Bidirectional)	(None, 150, 256)	439,296	spatial_dropout1d[0][0]
conv1d (Conv1D)	(None, 148, 64)	49,216	bidirectional[0][0]
global_average_pooling1d (GlobalAveragePooling1D)	(None, 64)	0	conv1d[0][0]
global_max_pooling1d (GlobalMaxPooling1D)	(None, 64)	0	conv1d[0][0]
concatenate (Concatenate)	(None, 128)	0	global_average_pooling1d[… global_max_pooling1d[0][0]
dense (Dense)	(None, 3)	387	concatenate[0][0]

1. Input Layer

- Type: InputLayer

- Output Shape: (None,150)
  - None: Batch size is unspecified (can vary dynamically during training or inference).
  - 150: Input sequence length (number of tokens/words per sequence).
- Parameters: 0 (No learnable weights in this layer).
- Connected to: This is the starting point of the model.

---

2. Embedding Layer

- Type: Embedding
- Output Shape: (None,150,300)
  - 150: Sequence length.
  - 300: Embedding dimension for each word.
- Parameters: 681,600
  - Calculated as: Parameters=**vocabulary size×embedding dimension**=2272×300=681,600
- Connected to: The InputLayer.

---

3. Spatial Dropout1D

- Type: SpatialDropout1D
- Output Shape: (None,150,300)
  - Applies dropout along the feature dimensions (300) while keeping the sequence structure (150) intact.
  - Purpose: Prevent overfitting by randomly zeroing out some embedding dimensions.
- Parameters: 0 (No learnable weights in dropout layers).
- Connected to: The Embedding layer.

---

4. Bidirectional LSTM

Forward LSTM Parameters:

For a single LSTM layer, the number of parameters is calculated as:

Params=4× (Input size+Hidden size+1) ×Hidden size

Where:

- Input size=300 (from embedding dimension),
- Hidden size=128,
- The extra +1 accounts for the bias terms.

Params for one direction= 4× (300+128+1) ×128=4×429×128=219,648

Bidirectional Parameters:

Since the layer is bidirectional, it doubles the parameters:

Total Params=2×219,648=439,296

This matches the table in the image.

5. Conv1D

The Conv1D layer applies a convolution operation over the sequence. The parameters are calculated as:

Params= (Kernel size×Input channels×Filters) +Filters

Where:

- Kernel size: 3
- Input channels: 256 (from the Bidirectional LSTM output)
- Filters: 64 (number of convolutional filters)

Params= (3×256×64) + 64= 49,152+64 = 49,216

6. Global Average Pooling 1D

- Type: GlobalAveragePooling1D
- Output Shape: (None,64)
  - Reduces the sequence dimension by computing the average value of each feature (64 filters) across all time steps (148).
- Parameters: 0 (No learnable weights).
- Connected to: The Conv1D layer.

---

7. Global Max Pooling 1D

- Type: GlobalMaxPooling1D
- Output Shape: (None,64)
    - Reduces the sequence dimension by computing the maximum value of each feature (64 filters) across all time steps (148).
- Parameters: 0 (No learnable weights).
- Connected to: The Conv1D layer.

8. Concatenate

- Type: Concatenate
- Output Shape: (None,128)
    - Combines the outputs of GlobalAveragePooling1D and GlobalMaxPooling1D: 64(average pooled features) +64(max pooled features) =128
- Parameters: 0 (No learnable weights).
- Connected to: GlobalAveragePooling1D and GlobalMaxPooling1D layers.

9. Dense Layer

- Type: Dense
- Output Shape: (None,3)
    - Output probabilities for 3 classes (multi-class classification).
- Parameters: 387

Calculated as: Parameters= **(input features×output units) +output units**= (128×3) +3=387

3.3 Testing and Evaluation

Now it's time to evaluate our model based on the train and test dataset that we've made. Here's a brief explanation of all the metrics that we'll be using:

Precision:

- **Definition**: Precision measures the accuracy of the positive predictions made by the model.
- **Formula**:

Precision=TP/(TP+FP)

Where:

- TP (True Positives): The number of correctly predicted positive instances.
- FP (False Positives): The number of instances incorrectly predicted as positive (i.e., negative instances incorrectly classified as positive).
- **Interpretation**: High precision indicates that when the model predicts a positive class, it is likely correct. It's especially important in scenarios where false positives are costly or undesirable (e.g., fraud detection).

Recall:

- **Definition**: Recall (also known as sensitivity or true positive rate) measures the ability of the model to identify all relevant positive instances.
- **Formula**:

Recall=TP/(TP+FN)

Where:

- FN(False Negatives): The number of instances incorrectly predicted as negative (i.e., positive instances missed by the model).
- **Interpretation**: High recall means that the model captures a large proportion of the actual positive instances. It is crucial in situations where missing positive instances is particularly harmful (e.g., disease screening)

The F1 score is a metric used to evaluate the performance of a classification model, particularly in situations where you want to balance precision and recall. It is especially useful when dealing with imbalanced datasets, where one class may be more important than the other.

Definition:

- The F1 score is the harmonic mean of precision and recall, providing a single score that reflects both metrics. It is calculated using the formula:

F1 Score=2×Precision×Recall/(Precision+Recall)

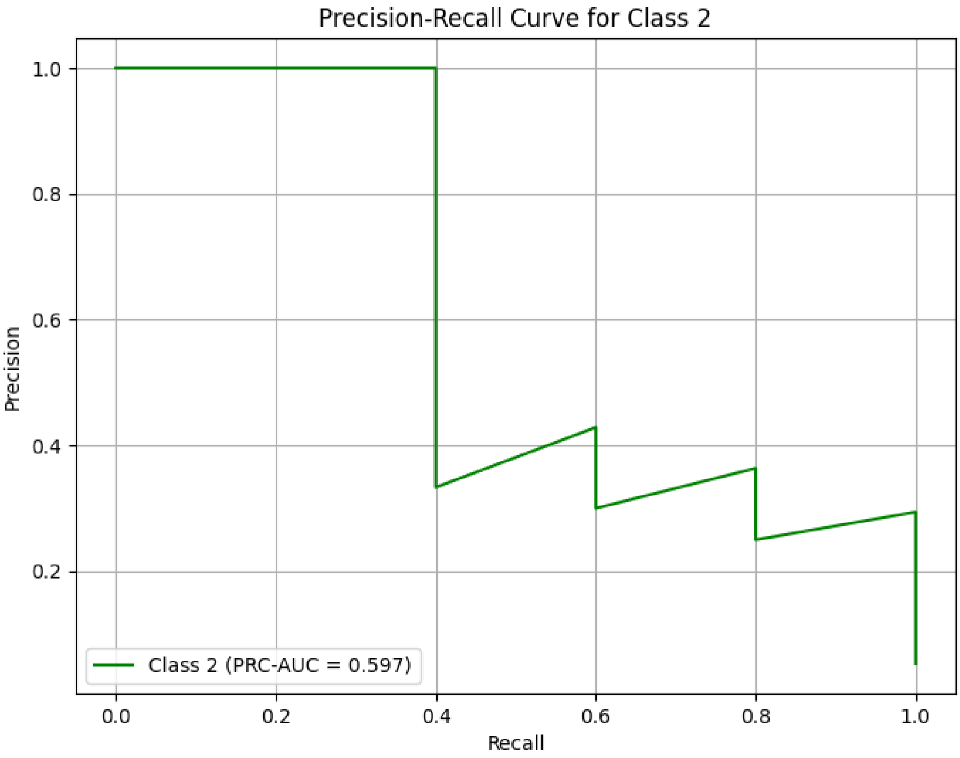
Interpretation:

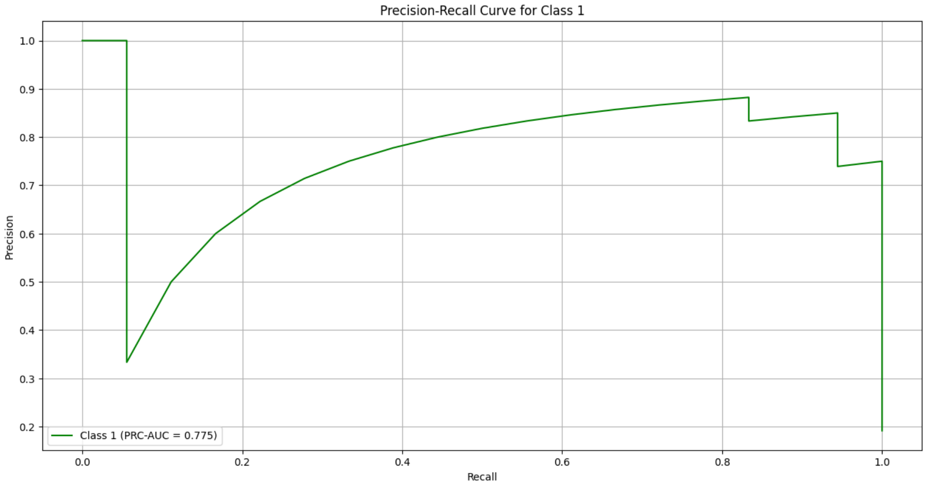
- An F1 score ranges from 0 to 1, where:

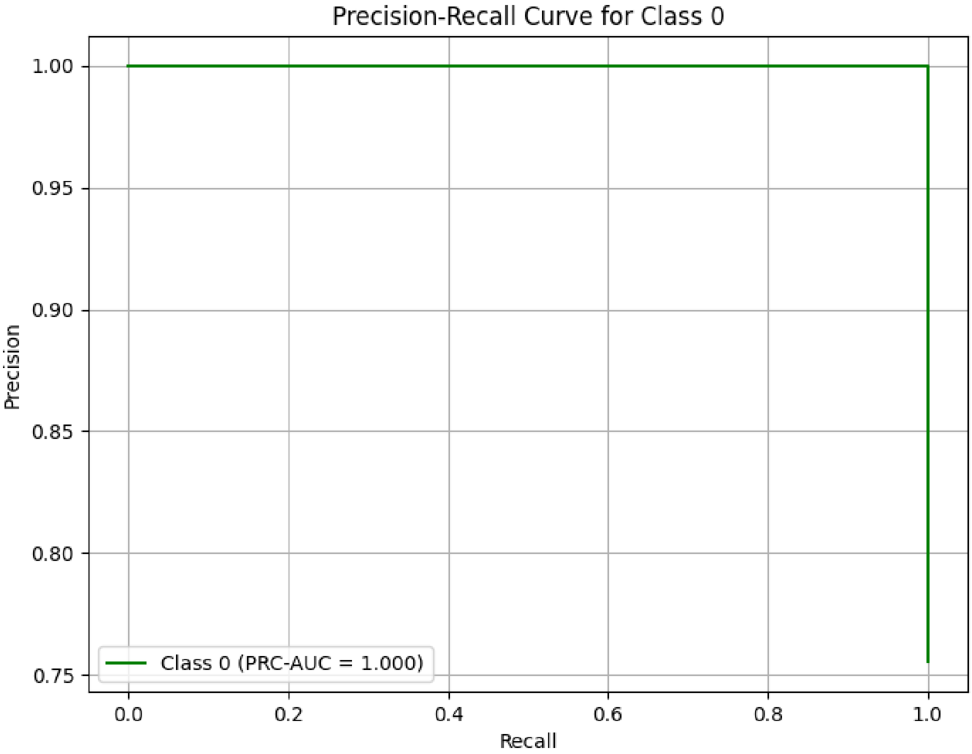
  - **1** indicates perfect precision and recall (the model makes no errors).
  - **0** indicates the worst performance (the model fails to make any correct predictions).

Below is the evaluation based on ROC-AUC (Receiver Operating Characteristic - Area Under Curve) and PRC-AUC (Precision-Recall Curve - Area Under Curve)

Since the dataset comprised of mostly CLEAN comments, it would be best if we use PRC-AUC since this evaluation metric is best suited for imbalance datasets and ROC-AUC is mainly used for binary classifiers (such as detecting spam emails). We also added an F1-score valuation purely for an overview of the overall effectiveness of the model.


Precision-Recall Curve for Class 2
Precision
1.0
0.8
0.6
0.4
0.2
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Class 2 (PRC-AUC = 0.597)


Precision-Recall Curve for Class 1
Precision
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Class 1 (PRC-AUC = 0.775)



Class 0 - PRC-AUC: 1.000, F1-Score: 0.940, TP: 63, FP: 0, TN: 23, FN: 8

Class 1 - PRC-AUC: 0.775, F1-Score: 0.571, TP: 8, FP: 2, TN: 74, FN: 10

Class 2 - PRC-AUC: 0.597, F1-Score: 0.444, TP: 2, FP: 2, TN: 87, FN: 3

Overall Observations:

1. **Class Imbalance**:
   - Class 0 is clearly the dominant class, as it has significantly more true positives and fewer errors.
   - Classes 1 and 2 might suffer from class imbalance, which could explain their lower performance metrics.
2. **Precision-Recall Tradeoff**:
   - Class 0 achieves an excellent tradeoff with high PRC-AUC and F1-Score.
   - Classes 1 and 2 show limited tradeoff, due to insufficient samples.

Conclusion and Future Directions

In conclusion, our project has provided meaningful insights into hate speech detection using deep learning techniques. Despite the challenges of working with a relatively small dataset, we have been able to design, train, and evaluate a model that shows promise in identifying different types of comments. However, we recognize that the limitations of the dataset—consisting of only around 864 comments—have significantly impacted the model's ability to generalize effectively, particularly for underrepresented classes.

Moving forward, the first and most critical step will be to expand the dataset. A larger and more diverse dataset will not only help balance the classes but also provide the model with richer context for better decision-making. This effort will be complemented by exploring advanced techniques such as attention mechanisms, transformers, and transfer learning to enhance the model's performance and robustness.

We would like to extend our heartfelt gratitude to our instructor, Mr. **Vu Hai**, for igniting a spark of curiosity within us for the fascinating world of machine learning, particularly in the realm of AI models. Your expertise, guidance, and passion for the subject have been truly inspiring, shaping not only our understanding but also our enthusiasm to delve deeper into this dynamic field. Additionally, we are immensely grateful for the collaborative and supportive environment fostered by our peers. The insightful discussions, constructive feedback, and shared learning experiences have greatly enriched our journey. Thank you for providing us with this incredible opportunity to explore and grow!