AIP490 Project

FPT UNIVERSITY

HATE TEXT DETECTION SERVICE: VIETNAMESE SENSITIVE WORDS DETECTION PLATFORM

by

Duy Le Nguyen Minh - DuyLNM

Hai Hoang Thanh - HaiHT

Huy Le Gia - HuyLG

THE FPT UNIVERSITY HO CHI MINH CITY

HATE TEXT DETECTION SERVICE: VIETNAMESE SENSITIVE WORDS DETECTION PLATFORM

by

Duy Le Nguyen Minh - DuyLNM

Hai Hoang Thanh - HaiHT

Huy Le Gia - HuyLG

Supervisor: M.S Vu Hoang Anh

A final year capstone project submitted in partial fulfillment of the requirement for the Degree of Bachelor of Artificial Intelligence in Computer Science

DEPARTMENT OF ITS

THE FPT UNIVERSITY HO CHI MINH CITY

December 2023

ACKNOWLEDGMENTS

Big shoutout to our supervisor, M.S Vu Hoang Anh, for dropping all that knowledge and expertise on us throughout this journey. Seriously, we couldn't have wrapped up this thesis without his endless patience, killer guidance, and spot-on feedback. A huge thanks for backing us up on the financials too, covering the costs for the remote server and domain. And hey, props to the Referee committee for giving our thesis the once-over to make sure we were on track.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: Conceptualization, Duy Le. and Hai Hoang.; methodology, Duy Le. and Huy Le.; software, Huy Le. and Hai Hoang.; formal analysis, Hai Hoang., Duy Le. and Huy Le.; investigation, Duy Le., Huy Le. and Hai Hoang.; data curation, Huy Le. and Hai Hoang.; writing—original draft preparation, Huy Le. and Duy Le.; writing—review and editing, Hai Hoang.; visualization, Huy Le. and Hai Hoang.; project administration, Duy Le. All authors have read and agreed to the Final Capstone Project document.

ABSTRACT

In the digital age, social media's pervasive influence has inadvertently escalated the prevalence of hate speech and offensive comments, with alarming implications for mental health. There is increasing evidence indicating a clear correlation between two factors. exposure to such toxic online content and the onset of depression among users, particularly affecting vulnerable groups like content creators and channel owners. Addressing this critical issue, our research introduces XBert, a model for detecting hostile and provocative language in Vietnamese. We propose an approach related to data preprocessing, improved tokenization, and model fine-tuning. We have modified the architecture of the Roberta model, used the EDA technique, and added a dropout parameter to the tokenizer. Our model achieved an accuracy of 99.75% and an F1-Macro score of 98.05%. This is a promising result for a model detecting provocative and hostile language in Vietnamese.

Keywords: Hate speech detection, transformer, model, Xbert, Vietnamese

CONTENTS

List of Figures

List of Tables

1.INTRODUCTION

In today’s digital era, online space like Facebook, Instagram, Youtube, v.v. has become an indivisible part of daily life. Social networks and online forums expand opportunities for people to connect and share ideas and information. However, the strong development of online platforms also brings new challenges, especially the issue of hate speech.

Hate speech in words, actions, and images that incite, defame, discriminate, insult, or harm the honor and dignity of individuals or groups based on race, religion, gender, ethnicity, political orientation, etc. Hate speech can exist in many platforms, such as: Distorting and defarming the reputation and honor of an individual or group people; Inciting violence or hatred against individuals or groups; Discrimination or stigmatization of individuals or groups of people; Insulting or harming the honor or dignity of an individual or group of people.

Recent data highlights the severity of this problem. The alarming statistics on Instagram's struggle against hate speech [1] serve as a stark reminder of the gravity of this issue. Facebook, the digital leviathan with over 2.9 billion active users [2], faces an even more formidable challenge. The sheer volume of content and the ever-evolving nature of language make it a formidable task to detect and remove hate speech effectively. The statistics paint a bleak picture: Microsoft's 2023 report [3] found that over a third of all online content is tainted with hate speech. This insidious trend threatens to erode the foundations of digital civility and undermine the principles of open dialogue and democratic discourse.

The linguistic diversity of social media platforms further complicates the issue of hate speech detection. With its support for over 100 languages [4], Facebook faces the challenge of deploying effective hate speech detection mechanisms in a multilingual context. This is particularly concerning for languages like Vietnamese, where the prevalence of hate speech is particularly acute [5].

The societal impact of hate speech extends far beyond online vitriol. Studies have shown a correlation between online hate speech and real-world violence [9]. The psychological toll is equally severe, with victims of online hate experiencing anxiety, depression, and even suicidal ideation [10].

Given the increase in hate speech in cyberspace, developing a platform to identify and detect hate speech in the Vietnamese language is an urgent issue. This platform will contribute to creating a safe and healthy online environment, protecting network users' legitimate rights and interests. The application of AI and natural language processing (NLP) technology in detecting hate speech is a major technological challenge. Hate language can manifest in many different forms, from offensive and insulting words to threats and incitement of violence. To accurately detect hate speech, the platform is to be built on a comprehensive database of the Vietnamese language while using advanced deep learning algorithms.

Considering these challenges, we propose a novel approach to hate speech detection in the Vietnamese language. We leverage the innovative BPE-Dropout algorithm [6], which introduces a stochastic element to the segmentation process, to accurately capture the nuances of the Vietnamese language. This method, as demonstrated by Nguyen and Tran [11] in their study on language processing, provides a more flexible and robust framework for understanding the fluid nature of online speech.

Furthermore, we adapt the XBert (based on the Roberta model)[7], a state-of-the-art natural language processing model, to specifically target the nuances of hate speech. By tailoring this model, we create a tool with the precision of a surgeon's scalpel, capable of identifying and classifying hate speech with enhanced accuracy.

In the subsequent sections of this article, we will delve into the intricacies of the BPE-Dropout algorithm's application to Vietnamese, the architectural tweaks to the XBERT (based on the Roberta model), and the implications of our findings. We will also explore the broader context of hate speech detection, drawing on sociology, linguistics, and computer science insights. This endeavor is not merely an academic exercise; it is a mission to safeguard the integrity of our digital interactions and foster a more inclusive and respectful online environment.

2. RELATED WORK

Hate speech detection identifies if a piece of communication, such as text or audio, contains hatred or encourages violence towards a person or a group of people [16]. Discrimination is often based on preconceived biases against certain traits or characteristics that are protected by law, such as race, gender, age, religion, or disability. Ethnicity, gender, sexual orientation, religion, age, etc. The United Nations defines *hate speech* as any communication in speech, writing, or behavior. It is unacceptable to use language that attacks or discriminates against individuals or groups based on their identity [17].

Numerous studies have been carried out to tackle this problem. For instance, MacAvaney et al. Discussed the challenges faced by online automatic approaches for hate speech detection in text [12]. They proposed a multi-view SVM approach that achieves near state-of-the-art performance while being more straightforward and producing more easily interpretable decisions than neural methods [12]. In recent years, the ways in which people receive news and communicate with one another has been revolutionized by the Internet, and especially by social networks [13]. This has led to the development of numerous methods for hate speech detection, including a recent proliferation of deep-learning-based approaches [18]. These methods have been utilized to solve past challenges and are expected to shape the prospects for hate speech detection [19]. They stressed the relevance of machine learning approaches to assess different forms of online speech correctly.

Several key studies have been conducted to develop models for hate speech in various languages. Hate speech detection: Challenges and Solution [20] this study identifies and examines challenges faced by online automatic approaches for hate speech detection in text; Hate Speech Recognition in Multilingual Text: Highlish Documents [21] this paper evaluates various machine learning and deep learning techniques to detect hate speech on various social media platforms in the Hinglish (English - Hindi code-mix) languages; A Literature Review of Textual Hate Speech Detection Methods and Datasets [22] this study systematically review textual hate speech detection systems and highlights their primary datasets, text features, and machine learning models.

The methodologies used in these studies vary, but they generally involve the use of machine learning and deep learning techniques. For instance, the study on Hinglish documents applied and evaluated several machine learning and deep learning methods, along with various feature extraction and word-embedding techniques [21]. The experimental results revealed that deep learning models generally performed better than machine learning models [21].

In terms of outcomes, these studies have contributed significantly to the field of hate speech detection. For example, the study on challenges and solutions proposed a multi-view SVM approach that achieved near state-of-the-art performance [20]. Meanwhile, the literature review provided insights and empirical evidence on the intrinsic properties of hate speech, helping the research community identify topics for future work [22].

These studies are relevant to your work as they provide valuable insights into the methodologies and outcomes of hate speech detection. They highlight the importance of using advanced techniques such as machine learning and deep learning, and the need for continuous research and development in this field.

Hate speech detection in the Vietnamese language is a growing field of research, with several notable studies and models developed specifically for this purpose. ViHOS: Hate Speech Spans Detection for Vietnamese[14] this study presents the ViHOS(Vietnamese Hate and Offensive Spans) dataset, the first human-annotated corpus containing 26k spans on 11k comments. The researchers also provide definitions of hateful and offensive spans in Vietnamese comments as well as detailed annotation guidelines. They conducted experiments with the various state of the art models. ViHSD- Vietnamese Hate Speech Detection dataset[15] This dataset is used for hate speech detection in the Vietnamese language. It contains 33,400 annotated comments used for hate speech detection on social network sites. The labels include CLEAN, OFFENSIVE, and HATE. Vietnamese Hate and Offensive Detection using PhoBERT-CNN[23]: This study proposed an efficient pre-processing technique to clean comments collected from Vietnamese social media. A novel hate speech detection (HSD) model, which is the combination of a pre-trained PhoBERT model and a Text-CNN model, was proposed for solving tasks in Vietnamese.

The unique challenges in the Vietnamese context include the rise in hateful and offensive language directed at other users, which is one of the adverse side effects of the increased use of social networking platforms[14]. The Vietnamese language's complexity, including its tonal nature and the use of diacritics, also presents unique challenges for hate speech detection.

The RobertaForSequenceClassification model is a Roberta[24] Model transformer with a sequence classification/regression head on top (a linear layer on top of the pooled output). It is a powerful transformer-based model developed by Facebook AI, which follows the same architecture as BERT[25] (Bidirectional Encoder Representations from Transformers) but is trained on more data and for a longer time than BERT.

The Xbert model extends the RobertaForSequenceClassification model by adding an additional sequential layer, named "Xbert", which consists of a linear layer, a Layer Normalization [26], and a Dropout layer [27]. This additional layer can help in better capturing the contextual information in the input data and thus improve the performance of the model in hate speech detection tasks.

There are several other models that have been used for hate speech detection: BERT-based models [28]: BERT-based models have been widely used for hate speech detection. For instance, a BERT-based transfer learning approach was proposed for hate speech detection in online social media; HateBERT [29]: HateBERT is a re-trained BERT model specifically for abusive language detection in English; BERT-based ensemble learning[30]: This approach combines the pre-trained BERT model with Deep Learning (DL) models to compose several ensemble learning architectures for multi-aspect hate speech detection.

Strengths of Xbert : The Xbert model leverages the power of the Roberta model, which is known for its robust performance on various NLP tasks. The additional sequential layer (Xbert) could potentially help in capturing more complex patterns in the data. The model is designed for sequence classification tasks, making it suitable for tasks like sensitive words detection.

Weakness of the Xbert : The model might require a large amount of labeled data for training to achieve good performance. The complexity of the model could lead to longer training times.

3. PROJECT MANAGEMENT PLAN

This presentation will provide a detailed overview of our project management strategy, including a breakdown of tasks and a schedule outlining the timelines for each segment of our project.

3.1 Data preparation

In the initial phase, our focus was on planning the research process, which involved strategizing how to extract data, preprocess it, and analyze the dataset. This stage was critical in laying the groundwork for our project, as it entailed developing a comprehensive approach for handling and interpreting the data effectively. Through this meticulous planning, we aimed to ensure that our data extraction and analysis methods were robust and tailored to meet the specific needs and objectives of our research.

Table 1. *Data preparation plan*

Task name	Owner	Start date	End date	Status
Find and analyze public dataset	Huy	09/05	09/08	Done
Research Facebook API	Hai + Huy	09/05	09/07	Done
Research tools to crawl comments from Facebook (selenium, bs4)	Hai + Huy	09/07	09/08	Done
Research and develop to extract comments from Facebook pages, groups	Hai	09/09	09/20	Done
Ask for standard datasets from VLSP & Vi-HSD	Duy	09/07	09/08	Done
Ask for dataset from external company	Hai	09/13	09/13	Done

Develop preprocessing class	Hai	09/21	09/30	Done
Preprocessing VLSP & Vi-HSD dataset	Hai	09/30	10/11	Done

3.2 Research model & papers

We divided to find and research to understand available papers and models to find relevant research and, based on this, to implement for further works, thereby enhancing our knowledge and capabilities in the field, and ultimately contributing to the development of innovative solutions and advancements.

Table 2.*Models research plan*

Task name	Owner	Start date	End date	Status
Review NLP knowledge	Duy + Huy + Hai	09/05	09/05	Done
Find relevant papers, documents (minimum 7 papers each member)	Duy + Huy + Hai	09/06	09/06	Done
Research theory of Bert, Roberta, PhoBert models	Duy + Huy	09/07	09/10	Done
Research frameworks, library methods to support to train NLP model	Duy	09/11	09/13	Done
Research subword tokenization algorithims (BPE, WordPiece, Unigram, Sentence Piece)	Huy	09/11	09/12	Done
Research K-fold cross-validation	Hai + Duy	10/14	10/16	Done

Research Easy Data Augmentation (EDA) to enrich dataset	Huy	10/15	10/16	Done

3.3 Experiments:

Following our study of the algorithms, we applied these models and conducted experiments with them to assess the outcomes, thereby gaining insights into their effectiveness and potential for application in real-world scenarios. This process allowed us to refine our approach and develop a deeper understanding of the models' capabilities and limitations, leading to more informed decisions in future applications.

Table 3. *Experiments plan*

Task name	Owner	Start date	End date	Status
Re-train bases models on 3 datasets (VLSP, Vi-HSD, social media)	Duy + Huy	09/14	09/20	Done
Evaluate base models when tunning parameters	Duy + Huy + Hai	09/21	09/22	Done
Build XBert model based on RobertaForSequenceClassification	Duy	09/07	09/15	Done
Build system writing log & monitoring for model	Hai	09/08	09/13	Done
Create XBert_Tokenize to optimize model training time & accuracy	Huy	09/11	10/15	Done
Evaluate XBert model on VLSP & Vi-HSD datasets	Hai	09/17	10/05	Done

Compare tokenizer of 2 methods: XBert & PhoBert	Huy	09/17	09/25	Done
Apply EDA to train XBert model	Duy + Huy	09/24	09/26	Done
Implement K-fold method with the same hyperparameters and datasets with relevant papers	Duy	10/13	10/15	Done

3.4 Writing research journal

To demonstrate our accomplishments and contribute to the wider community, we have authored a research paper. This document not only serves as proof of our work but also aims to share our findings and methodologies with others in the field. By publishing this paper, we hope to inspire further research and collaboration, enriching the collective knowledge base and fostering advancements in our area of study.

Table 4. *Writing research plan*

Task name	Owner	Start date	End date	Status
Abstract + Introduction	Huy	10/16	10/16	Done
Preprocessing data	Hai + Duy	10/16	10/17	Done
Tokenizer	Huy	10/17	10/17	Done
Model Architecture	Duy	10/17	10/17	Done
Data description	Huy	10/18	10/18	Done
Experiments & Result	Hai	10/17	10/18	Done
Discussion + Conclusion	Hai	10/19	10/19	Done

3.5 Deployment

To showcase this project, we created a prototype in the form of a website application. This interactive platform serves as a practical example of the project's capabilities, allowing users to engage with its features and functionalities directly. It not only demonstrates the practical application of our research but also provides a user-friendly interface for testing and feedback, which is crucial for further development and refinement of the project.

Table 5. *Demo Solution plan*

Task name	Owner	Start date	End date	Status
Create solution, pipeline of demo platform	Hai	10/25	10/26	Done
Research Google API to extract data from Youtube	Hai	10/27	10/28	Done
Research frameworks to create UI to visualize data (Streamlit, Matplotlib)	Huy	10/28	10/30	Done
Research FlaskAPI to create API	Duy	10/25	10/27	Done
Build function to extract comments of all videos of an Youtube channel	Hai + Huy	10/29	11/10	Done
Build model API	Duy	10/28	11/02	Done
Build platform for user to interact and visualize data	Huy	11/01	11/17	Done

3.6 Document

The final component of our project is this report, in which we have meticulously documented our efforts for the benefit of the readers.

Table 6. *Writing document plan*

Task name	Owner	Start date	End date	Status
Abstract + Introduction	Duy	11/19	11/20	Done
Related Work	Duy	11/19	11/24	Done
Project Management Plan	Hai	11/23	11/25	Done
Materials and methods	Huy	11/19	11/21	Done
Result	Huy	11/23	11/24	Done
Discussion	Hai	11/19	11/19	Done
Conclusion	Hai	11/20	11/20	Done

4.MATERIALS AND METHODS

4.1. Project Pipeline

Data Preparation: Data preparation is usually the first step in an AI project. We will collect data from two sources and process them before moving to the next step.

- The first dataset was introduced in the paper "HSD Shared Task in VLSP Campaign 2019: Hate Speech Detection for Social Good." This dataset, curated to support research in hate speech detection, particularly in Vietnamese, includes 25,431 samples collected from Facebook, a widely used social media platform in Vietnam. The second dataset is featured in the article "A Large-scale Dataset for Hate Speech Detection on Vietnamese Social Media Texts." It comprises user comments from Vietnamese Facebook posts and highly interactive YouTube videos, containing over 30,000 comments. Both datasets are labeled as CLEAN (0), ATTACK (1), and HATE (2).

- Preprocessing Data: The very first step, before fitting the raw data into the model, is preprocessing it. Data preprocessing, a component of data preparation, involves any type of processing performed on raw data to prepare it for another processing stage. It has traditionally been a crucial preliminary step in the data mining process. Here, we focus on data preprocessing for NLP (Natural Language Processing).

Fitting the Model to the Data: After cleaning the data, the next step is to use a machine learning model to identify important keywords in that data. The machine learning model will process this data, using its algorithms to "select" key keywords. For more detailed information, please refer to the section ...., where we have discussed this in detail.

Evaluation: After obtaining the output results, it's essential to ensure they are reliable enough for use. Evaluation is an indispensable stage in an AI project, aimed at reviewing the quality of the project and comparing this method with others. The evaluation phase is detailed in the fifth section (Results) of this report, which includes all the methods we used to evaluate the model's results.

4.2. Data Preprocessing

In this stage, we do the following work to preprocess the raw data to be usable.

Stop words Removal: Stop words are common words that, while structurally important, often carry little meaningful content for analysis. They are frequently removed to reduce the dimensionality of text data, thereby enhancing the efficiency of data processing. In our approach, we utilize a curated list of Vietnamese stop words to filter out these words from the input text.

Addressing the placement of diacritics in Vietnamese is crucial, as there are currently two prevalent styles: the old style and the new style. This variation can lead to inconsistencies in text processing. To ensure uniformity, we will synchronize and convert any words using the old style to the new style. This standardization is vital for maintaining consistency in our data processing. The rules for the new formatting style are as follows:

- **Diacritic Placement for Single Vowels:** For syllables where the main vowel is a single vowel, the placement of diacritics will be at the position of the letter representing this main vowel. This rule helps in standardizing the text and is crucial for accurate interpretation and processing.
- **Diacritic Placement for Double Vowels:** The approach varies slightly for syllables with double vowels. For combinations like iê, yê, uô, ươ, where the ending vowels are followed by consonants such as p, t, c, ch, m, n, ng, nh, o, u, i, the diacritic is placed on the second letter of the main vowel. Conversely, for combinations like ia, ya, ua, ưa, the diacritic is placed on the first letter. This nuanced approach to diacritic placement is essential for maintaining the linguistic integrity of the Vietnamese language in our data.

Next, we focus on cleaning up the input text. This involves several steps:

- **Replacing Emojis:** Emojis are replaced with spaces. While emojis can convey emotions and ideas, they are often not directly relevant to text-based analysis and can introduce noise into the data.

- **Removing Unicode Characters:** We eliminate Unicode characters that are not part of the standard Vietnamese character set to maintain consistency in encoding.
- **Deleting Special Characters and URLs:** Special characters and URLs are removed. These elements often do not contribute to the overall meaning of the text and can skew analysis if left untreated.
- **Removing Extra Spaces and Lowercasing:** We also remove extra spaces and convert all letters to lowercase. This normalization step is crucial for ensuring that the text is uniformly processed, reducing the chances of the same words being treated differently due to case differences.

Finally, for special characters that are not letters and are repeated multiple times in a sentence, we will streamline the text by removing these repetitions and keeping only a single instance of each special character. This step is important to avoid overemphasis on certain punctuation marks or special characters, which could otherwise lead to skewed analysis or processing errors.

By meticulously following these steps, we ensure that the text data is clean, consistent, and ready for further processing, such as machine learning modeling or advanced text analysis. This thorough preparation is key to the success of any data-driven project, especially in a language as complex and nuanced as Vietnamese. It also sets the foundation for accurate and reliable linguistic analysis, enabling more meaningful insights to be drawn from the data.

4.3. Easy Data Augmentation

Data augmentation techniques provide a straightforward approach to enhancing performance in text classification tasks. These methods encompass a range of general and easily applicable strategies, particularly beneficial for addressing challenges associated with small datasets. By leveraging data augmentation, it becomes possible to amplify the effectiveness of text classification models, thereby improving their overall accuracy and robustness.

Techniques Involved:

- **Synonym Replacement (SR):** Randomly replace words with synonyms.
- **Random Insertion (RI):** Insert random synonyms at random positions.
- **Random Swap (RS):** Swap random words in the sentence.
- **Random Deletion (RD):** Randomly delete words in the sentence.

Table 7. *EDA Example*

Kỹ thuật	Câu gốc	Câu đã sửa
Synonym Replacement (SR)	Con mèo đang ngồi dưới cái cây.	Con mèo đang ngồi dưới cái thân cây.
Random Insertion (RI)	Con mèo đang ngồi dưới cái cây.	Con mèo đang ngồi dưới cái cây xanh.
Random Swap (RS)	Con mèo đang ngồi dưới cái cây.	Con đang mèo ngồi dưới cái cây.
Random Deletion (RD)	Con mèo đang ngồi dưới cái cây.	Con mèo ngồi dưới cái cây.

Below are some of the main benefits:

When dealing with limited training data, it has been observed that performing Exploratory Data Analysis (EDA) on just 50% of the available data can yield accuracy levels that are comparable to those achieved by training on the entire dataset without EDA. This finding underscores the potential of EDA in maximizing the utility of limited data resources, thereby enhancing the efficiency and effectiveness of machine learning models, particularly in the realm of text classification.

One of the key advantages of EDA lies in its ability to diversify text data by introducing variations through techniques such as synonym replacement, random word insertion, swap, and deletion. By leveraging these methods, EDA facilitates the exposure of models to a broader spectrum of textual patterns, thereby bolstering their capacity to generalize and make accurate predictions across diverse inputs. This diversification of data is pivotal in enabling models to capture the nuances and intricacies inherent in natural language, ultimately contributing to improved performance and robustness.

Table 8. *Overview statistics before & after data augmentation*

Dataset	Label	Original dataset			Augmented dataset		
		Num comments	Avg word length	Vocab size	Num comments	Avg word length	Vocab size
Vi-HSD	CLEAN	19,886	6.55	130,238	19,886	6.55	130,238
	OFFENSIVE	1,606	7.24	11,624	10,147	7.57	76,802
	HATE	2,556	12.08	30,883	16,849	11.64	196,086
VLSP	CLEAN	18,614	14.85	276,557	18,614	14.85	276,557
	OFFENSIVE	1,022	8.87	9,063	8,461	8.05	68,093
	HATE	709	14.23	10,087	6,392	13.41	85,713

These techniques are applied to the ViHSD training set and the HSD-VLSP dataset with the percentage of replacement word in the sentence (α) set to 0.2. Table shows that after applying exploratory data analysis (EDA) techniques, the number of data and vocabulary size on the HATE and OFFENSIVE labels increased significantly. By utilizing these EDA techniques, the system aims to effectively manage the imbalance in the data, thereby improving the overall performance and accuracy of the model

However, despite its numerous benefits, the EDA technique is not without its limitations and potential drawbacks. Specifically, there are concerns regarding the risk of altering the context of sentences, as methods like random word replacement, insertion, or deletion may inadvertently distort the original meaning or information conveyed. Additionally, the accuracy of synonym selection during the replacement process poses a potential challenge, as inappropriate choices may compromise the quality and relevance of the augmented data.

To address these drawbacks, it is imperative to adopt mitigation approaches that can help alleviate the potential risks associated with EDA. For instance, limiting the number of changes made to each sentence and conducting thorough testing and evaluation on EDA-processed datasets are essential strategies for mitigating the concerns and ensuring the integrity and contextual fidelity of the augmented data. These mitigation measures are crucial in safeguarding the quality and reliability of the augmented data, thereby reinforcing the efficacy of EDA as a valuable tool for enhancing the performance of text classification models.

4.4. Tokenizer

Sub-word tokenization, a method that breaks down words into smaller units – or sub-words – has gained significant traction in recent years. This approach strikes a balance between the granularity of character-level tokenization and the context-awareness of word-level tokenization. By decomposing words into sub-word units, such as morphemes, syllables, or character n-grams, sub-word tokenizers can effectively mitigate the challenges posed by inflectional and agglutinative languages, where the number of possible word forms can be exponentially large.

The Impact of Sub-word Tokenization on Language Models

The advent of sub-word tokenization has not only addressed the limitations of traditional tokenization methods but has also paved the way for more robust and versatile language models. Models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), and their derivatives have demonstrated remarkable performance improvements across a range of NLP tasks, largely attributable to their underlying sub-word tokenization mechanisms.

One of the techniques used in NLP is **Byte Pair Encoding (BPE)**, which is a method for segmenting text into 'sub-words'. Originally developed for data compression, BPE has found widespread applications in machine learning models, particularly in natural language processing and machine translation.

Table 9. *Tokenizer methods comparison*

Model	BPE	WordPiece	Unigram
Training	**Starts from a small vocabulary and learns rules to merge tokens**	Starts from a small vocabulary and learns rules to merge tokens	Starts from a large vocabulary and learns rules to remove tokens
Training step	Merges the tokens corresponding to the most common pair	Merges the tokens corresponding to the pair with the best score based on the frequency of the pair, privileging pairs where each individual token is less frequent	Removes all the tokens in the vocabulary that will minimize the loss computed on the whole corpus
Learning	**Merge rules and a vocabulary**	Just a vocabulary	A vocabulary with a score for each token
Encoding	Splits a word into characters and applies the merges learned during training	Finds the longest subword starting from the beginning that is in the vocabulary, then does the same for the rest of the word	Finds the most likely split into tokens, using the scores learned during training

The reason we choose the BPE algorithm is that this algorithm can be applied to small-sized datasets, and it can combine words based on pre-trained rules, which is very important. For Vietnamese words, when they stand together, they have one meaning, but when separated, they have a different meaning. For example, the word "Tiền mặt" means money made of paper or metal, often used for buying and selling. But when separated, "Tiền" is a type of material used to exchange and buy. "Mặt" is a part of an object, usually the appearance.

To understand how BPE works, let's delve into its step-by-step process. BPE starts with a base vocabulary, which consists of all individual characters or bytes in the dataset. The algorithm then identifies the most frequently occurring adjacent pair of characters (or bytes) in the dataset. This pair of characters is merged into a new unit and added to the vocabulary. This process is repeated until a fixed number of merges is reached or the desired vocabulary size is achieved.

However, there is a variant of BPE called **BPE-Dropout** that introduces randomness into the segmentation process. This technique randomly omits some merges during segmentation, resulting in various sub-word segments for the same word. The purpose of BPE-Dropout is to create diversity in sub-word segmentation, which helps the model better understand the structure and meaning of words. This is particularly important in sentiment analysis, where accurately interpreting the sentiment behind words is crucial.

BPE-Dropout also offers other benefits. The diversity in segmentation increases the model's resilience to linguistic variations, making it more effective in classifying sentiments in texts that use slang, colloquialisms, or misspellings. Additionally, by enriching word representations, BPE-Dropout helps the model not only learn from training data but also better generalize when encountering new data.

There are a few points to note when using BPE-Dropout. Finding the right dropout rate is crucial to balance diversity and stability in word segmentation. Adjusting the dropout rate allows for fine-tuning the level of randomness introduced during segmentation. Furthermore, it is important to conduct experiments to assess the effectiveness of BPE-Dropout in sentiment classification tasks. Comparing it with standard BPE and other word segmentation methods can provide insights into its performance and suitability for different applications.

4.5. Model

4.5.1 RCNNs

In the field of deep learning [30], Recurrent Neural Networks (RNNs) [31] are a pivotal model, especially for processing sequential data. RNNs are distinguished by their ability to maintain a form of 'memory', making them particularly effective in tasks where understanding the sequence and context of data is crucial. This feature is enabled by the network's architecture, where output from a layer is fed back into the same layer as input. This recurrent nature allows the network to hold onto previous information, a trait not found in traditional feedforward neural networks.

RNNs have demonstrated significant success in various applications such as natural language processing, speech recognition, and time series analysis. However, they are also known to struggle with long-term dependencies [32] due to issues like vanishing and exploding gradients. These challenges have led to the development of more advanced versions of RNNs, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which are designed to better capture long-range dependencies in data.

4.5.2 LSTM

Long Short-Term Memory (LSTM) networks are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter and Schmidhuber in 1997[34] and have since been refined and improved by numerous researchers. LSTMs [35] are particularly known for their effectiveness in tasks where the context can span over long sequences of data, such as in language modeling and translation, speech recognition, and time series prediction.

The key innovation in LSTMs is the use of a cell state, which acts like a conveyor belt, carrying relevant information throughout the sequence of the data. The information flow through the cell state is regulated by structures called gates:

- **Forget Gate**: Decides what information should be discarded from the cell state.
- **Input Gate**: Updates the cell state with new information.
- **Output Gate**: Determines what the next hidden state should be, which is used for predictions.

These gates allow LSTMs to selectively remember and forget information [36], making them much more efficient and powerful for handling long-range dependencies compared to traditional RNNs.

LSTM networks have been instrumental in advancing the performance of various sequence modeling tasks. They are a cornerstone in the development of more complex sequence learning models and have laid the groundwork for many state-of-the-art deep learning architectures.

4.5.3 Bert (Bidirectional Encoder Representations from Transformers)

BERT, developed by researchers at Google in 2018, represents a groundbreaking approach in the realm of natural language processing (NLP) [37]. It's a deep learning model based on the Transformer architecture, introduced by Vaswani et al. in 2017 [38]. Unlike previous models that read text input sequentially (either left-to-right or right-to-left), BERT is designed to read in a bidirectional way, allowing it to capture context from both directions. This bidirectional nature is a key distinguishing feature of BERT and a primary contributor to its effectiveness.

BERT is pre-trained on a large corpus of text (like Wikipedia and BooksCorpus) using two novel strategies:

Masked Language Modeling (MLM): Randomly masking words in a sentence and predicting them based only on their context.

Next Sentence Prediction (NSP): Predicting whether a given sentence logically follows another sentence.

This pre-training enables BERT to develop a deep understanding of language context and flow. Once pre-trained, BERT can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, like question answering, sentiment analysis, and language inference, without substantial task-specific architecture modifications.

BERT has been influential in NLP, leading to the development of various derivatives and improvements, such as RoBERTa, DistilBERT, and ALBERT, each offering variations in terms of training data, model size, and optimization techniques.

4.5.4 Roberta (Robustly Optimized BERT Approach)

RoBERTa [39], introduced by Facebook AI in 2019 [40], is an optimized version of the BERT model, specifically aimed at improving its performance and robustness. It builds upon BERT's foundation, but with several key modifications to the training procedure and model architecture that enhance its effectiveness. These changes include:

- **Training on More Data**: RoBERTa is trained on a significantly larger dataset compared to BERT, including more diverse and extensive text corpora. This extensive training helps it to better understand and process various linguistic nuances.

- **Longer Training with Larger Batches:** The model undergoes longer training with larger mini-batches, allowing for more comprehensive learning and adaptation.

- **Elimination of Next Sentence Prediction (NSP):** Unlike BERT, RoBERTa removes the NSP task during training, focusing solely on the Masked Language Modeling (MLM) task.

- **Dynamic Masking:** RoBERTa introduces dynamic masking, where the masking pattern is changed during the training process, as opposed to the static masking in BERT.

- **More Careful Hyperparameter Tuning:** The developers of RoBERTa experimented with different hyperparameters, leading to a more finely-tuned model.

4.5.6 Xbert

Our model, based on the RoBERTa architecture, is a prime example of this cutting-edge technology. RoBERTa itself is an evolution of BERT (Bidirectional Encoder Representations from Transformers), optimized for more robust performance. It's known for its deep understanding of context and nuance in language, which makes it a powerhouse for a wide range of NLP tasks.

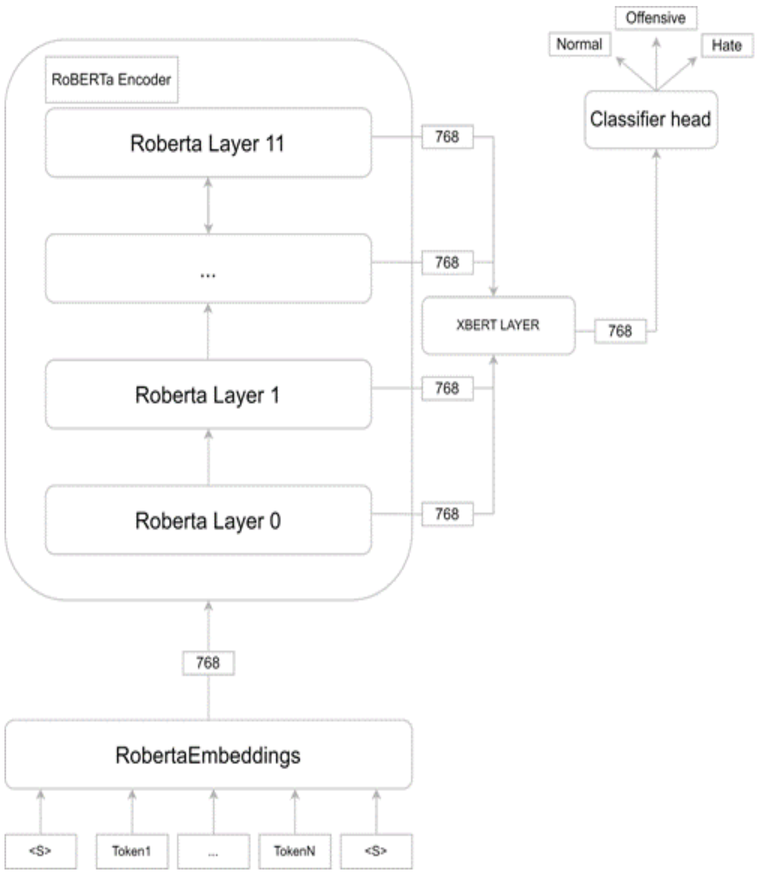


Figure 1. *XBert architecture*

Let's dive into the specifics of your model's architecture and understand how each component contributes to its overall capabilities. We'll explore the foundational layers provided by RoBERTa and then focus on how the Xbert layer integrates with and enhances this base model. This will give us a comprehensive view of not just how our model works, but why it's structured the way it is, and how this structure makes it well-suited for the tasks we are tackling.

a. Roberta Model

At its core, our model is built upon the RoBERTa architecture, a variant of BERT that has been specifically optimized to deliver enhanced performance and robustness. RoBERTa is widely recognized for its exceptional ability to handle a diverse range of natural language processing (NLP) tasks, making it an ideal choice for your model.

To facilitate the model's comprehension of the input data, it begins with embedding layers. These layers play a critical role in converting raw text into numerical vectors, enabling the model to effectively process and interpret the information. Within these embedding layers, you have word embeddings, position embeddings, and token type embeddings. Word embeddings capture the semantic meaning of each word, while position embeddings provide the model with a sense of word order. Additionally, token type embeddings, which serve as placeholders in RoBERTa, ensure that the structure remains consistent with BERT.

The word embeddings within the embedding layers are responsible for capturing the semantic meaning of each word. By assigning numerical vectors to words, the model gains a deeper understanding of the context in which they are used. This allows the model to grasp the nuances and subtleties of language, enabling it to accurately process and interpret the input text.

Position embeddings, on the other hand, provide the model with a sense of word order. By assigning unique numerical vectors to each word based on its position in the sentence, the model can understand the sequential nature of language. This helps the model to differentiate between words that appear at different positions within a sentence, allowing it to develop a more comprehensive understanding of the text.

Token type embeddings, which serve as placeholders in RoBERTa, ensure that the structure of the model remains consistent with BERT. By assigning distinct numerical vectors to different types of tokens, such as words from the input text or special tokens like [CLS] and [SEP], the model can maintain the integrity of the original architecture. This consistency is crucial for the model to effectively process and interpret the input text.

Following the embedding layers, the input text progresses through a series of encoder layers, with your model employing a total of 12 layers. Each encoder layer consists of two primary components: a self-attention mechanism and a feed-forward neural network. The self-attention mechanism empowers the model to assign varying degrees of importance to different words within a sentence, enabling it to develop a contextual understanding of the text. This contextual understanding is further refined and processed by the feed-forward network, which leverages the information obtained from the self-attention mechanism.

By incorporating the RoBERTa architecture, your model benefits from its robustness and versatility in handling a wide array of NLP tasks. The embedding layers and encoder layers work in tandem to ensure that the model can effectively comprehend and process the input text, ultimately enabling it to deliver accurate and meaningful results. With its exceptional ability to handle a diverse range of NLP tasks, RoBERTa serves as the foundation for our model's success in delivering enhanced performance and robustness.

b. XBert Model

This is where things get custom. To enhance the processing capabilities, our have incorporated an Xbert layer into the system. The Xbert layer consists of a sequence of layers that are specifically designed to perform additional processing tasks. This incorporation of the Xbert layer allows for a more advanced and sophisticated approach to handling the data. By taking a closer look at each component of the Xbert layer, we can gain a deeper understanding of its functionality and the benefits it brings to the system.

Firstly, let's examine the linear layer. This component serves as a fully connected neural network layer, which means that it establishes connections between every input and output neuron. Its main function is to take the 768-dimensional output from each token that has been processed by RoBERTa and transform it. By doing so, it has the potential to capture more complex relationships and patterns that are specific to the task at hand. This transformation process allows for a more nuanced analysis of the data, enabling the system to extract more meaningful insights.

Moving on to the next component of the Xbert layer, we have the layer normalization technique. This technique plays a crucial role in stabilizing the training process. It achieves this by normalizing the outputs of the previous layer, ensuring that they have a mean of 0 and a standard deviation of 1. This normalization step helps the model to train faster and more effectively, especially in deep networks like the one being used here. Deep networks often face challenges with unstable training behavior due to the sheer number of layers involved, and layer normalization helps to mitigate this issue. By ensuring that the outputs are within a standardized range, the system can maintain a more consistent and reliable training process.

Lastly, we encounter the dropout layer, which serves as a form of regularization. The dropout layer randomly "drops out" a proportion of the features by setting them to zero. This prevents the model from becoming overly reliant on any specific feature, thereby reducing the risk of overfitting to the training data. Overfitting occurs when a model becomes too specialized in capturing the patterns present in the training data, making it less effective in generalizing to new, unseen data. By promoting diversity in feature usage, the dropout layer enables the model to generalize better to new, unseen data. This regularization technique adds an extra layer of flexibility and adaptability to the system, allowing it to handle a wider range of tasks and datasets.

In summary, the Xbert layer, consisting of the linear layer, layer normalization, and dropout layer, adds customizability and enhances the processing capabilities of the system. It enables the model to capture more intricate relationships, stabilize the training process, and prevent overfitting, ultimately leading to improved performance on various tasks. The incorporation of these components into the system represents a significant advancement in the field of natural language processing, as it allows for a more sophisticated and nuanced analysis of textual data. With the Xbert layer in place, our system is equipped to handle complex tasks and deliver accurate and reliable results.

c. Classifier

The final part of our model is the classification head, which plays a crucial role in the overall architecture of the model. Its main function is to take the intricate representations and transformations generated by the preceding layers and convert them into concrete predictions for the specific task at hand. The implementation of the classification head involves a relatively straightforward process that consists of several steps.

Firstly, a dense layer is employed in the classification head. This dense layer is fully connected, ensuring comprehensive information flow throughout the model. By connecting all the neurons in the dense layer, the model can capture complex patterns and relationships in the data, leading to more accurate predictions.

To enhance the model's generalization capabilities and prevent overfitting, a dropout layer is added after the dense layer. The dropout layer randomly sets a fraction of the input units to zero during training, which helps to regularize the model and reduce its reliance on specific features or patterns in the data. This regularization technique improves the model's ability to generalize well to unseen data.

Finally, the predictions are produced by a linear layer in the classification head. This linear layer maps the 768-dimensional representation generated by the preceding layers to the number of classes being predicted, which in this case is 3. By applying a linear transformation, the model can produce output that corresponds to the specific task at hand, providing predictions for each class.

In summary, the classification head is a crucial component of the model architecture, responsible for converting the complex representations and transformations into concrete predictions. By employing a dense layer, dropout layer, and linear layer, the classification head ensures comprehensive information flow, regularization, and accurate predictions for the specific task.

5. RESULTS

XBert, the hate speech detection model, achieves an impressive accuracy rate of 99.75% and an F1-Macro score of 98.05%. These exceptional results highlight the model's exceptional capability in identifying and flagging hateful and offensive speech accurately.

5.1 Methods and Improvements

Data Preprocessing: The article delves into the importance of data preprocessing in enhancing the accuracy of the model. Specifically, it focuses on the processing of Vietnamese accents and the removal of special characters, emojis, and URLs from the text. These preprocessing techniques play a crucial role in improving the model's overall performance.

Data Augmentation (EDA): To further enhance the model's performance, the researchers employ data augmentation techniques known as EDA (Easy Data Augmentation). This technique involves various strategies such as word substitution, word insertion, word position change, and word deletion. By applying these techniques, the model becomes more robust and better equipped to handle different types of hate speech.

Tokenizer BPE-Dropout: Another key aspect of the XBert model is the utilization of the BPE-Dropout technique. This technique aids in increasing the model's generalization ability by creating multiple word segments within the same fixed BPE (Byte Pair Encoding) frame. This approach allows the model to handle a wide range of hate speech instances effectively.

The XBert Layer is a recent addition to the encoder's output, designed to improve the detection of hate speech in Vietnamese. This new contribution layer fine-tunes the encoder's output to better suit the specific task at hand.

5.2 Methods and Improvements

In a comprehensive evaluation, the XBert model has been found to outperform other baseline models, including PhoBERT and PhoBERT-CNN, in terms of both accuracy and F1-Macro score. This superiority has been consistently observed across all tests conducted during the 10-fold cross-validation process. These findings not only demonstrate the impressive performance of the XBert model but also emphasize the significant advancements it has made in the field of hate speech detection. The XBert model's exceptional results highlight its potential to contribute to the ongoing efforts in combating hate speech.

Table 10. *Comparison performance of methods*

Model	**VLSP**		**VI-HSD**	
	Accurracy	Macro-F1	Accuracy	Macro-F1
PhoBERT	94.1	66.03	86.61	53.0
PhoBERT - CNN	98.26	90.89	87.17	64.43
XBERT	**99.75**	**98.05**	**87.44**	**61.18**

5.3 Topic trend

The current trends in using hate speech detection tools in daily life, especially in the context of social media and online platforms, show increasing relevance and importance. As society becomes more interconnected through the internet, the prevalence of hate speech has become a pressing issue that needs to be addressed. A study conducted to analyze over one million comments on YouTube shed light on the growing presence of hate speech in online spaces, emphasizing the urgent need for countermeasures to mitigate its harmful effects.

Interestingly, the study found that hate speech on YouTube was slightly more prevalent than on other social media platforms. This finding suggests that YouTube, with its vast user base and diverse content, may be more susceptible to the spread of hate speech. Furthermore, the study revealed that hate speech did not exhibit specific temporal patterns, indicating that it is a persistent problem that requires continuous monitoring and intervention.

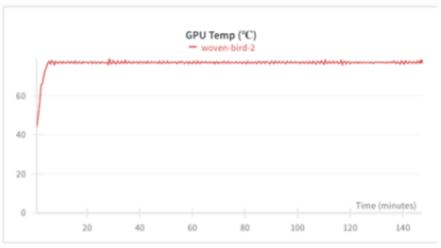
The focus on hate speech detection aligns with the broader trend of integrating advanced AI and machine learning techniques to combat the spread of hate speech in everyday online interactions. By leveraging these technologies, society can strive towards creating a safer and more inclusive online environment for all users.

6. DISCUSSION

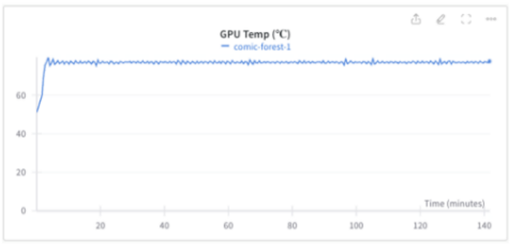
After conducting a comprehensive and in-depth analysis of the data, we have discovered that the implementation of the Easy Data Augmentation (EDA) technique has yielded remarkable and noteworthy results, leading to a significant improvement in accuracy. The findings we have presented above serve as concrete evidence of the effectiveness and efficacy of this approach. Moreover, it is worth mentioning that our observations have revealed an additional benefit in the form of the BPE-Dropout method, which has successfully reduced the model's processing time by approximately 2 minutes. This reduction in processing time is undeniably a crucial factor that contributes to making the model more efficient and practical for real-world applications.

In our relentless pursuit of enhancing the model's performance, we have taken a step further by incorporating an extra processing layer into the architecture. This strategic addition has proven to be highly promising and has yielded impressive results, aligning perfectly with our initial expectations. The inclusion of this supplementary layer has played a pivotal role in further augmenting the accuracy of the model, solidifying our confidence in its ability to handle a wide range of real-world scenarios with utmost precision and reliability.

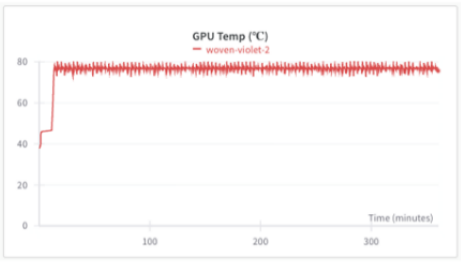
However, the cost of reaching heightened accuracy with this model comes in the form of longer training times and heightened consumption of GPU and disk resources. This necessitates not only a careful balance between the pursuit of precision and the efficient use of computational resources but also a strategic planning in resource allocation to optimize both the performance and cost-effectiveness of the model's operation over time.



PhoBert

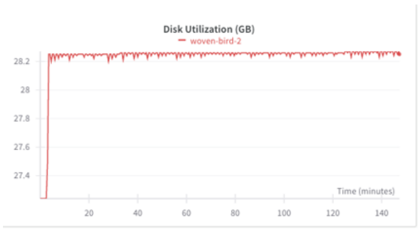


PhoBert-CNN

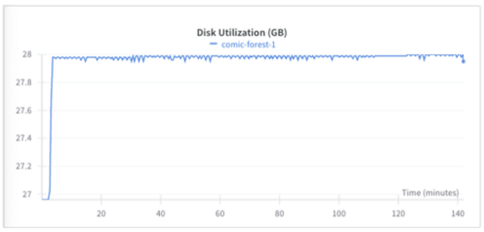


XBert

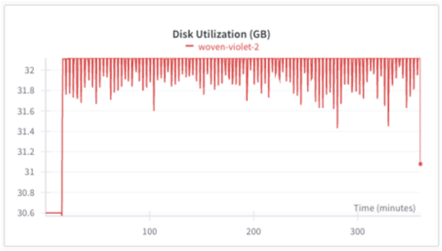
Figure 2. *GPU Capacity*



PhoBert



PhoBert-CNN



XBert

Figure 3. *Disk utilization*

Looking ahead, the research direction for this model revolves around the continuous pursuit of increasing its accuracy or reducing its processing time to the maximum extent possible. We firmly believe that these ongoing improvements will undoubtedly render the model even more applicable and indispensable in real-world projects. The prospect of exploring new avenues and innovative techniques to optimize its performance fills us with great excitement and motivation as we strive to push the boundaries of what is achievable in the field of machine learning and data analysis.

7. CONCLUSIONS AND PERSPECTIVES

The broad context of our project is to introduce a novel model, XBERT, which is specifically designed to detect hate speech in Vietnamese social media texts. This model stands out due to its high accuracy and efficiency, surpassing previous models like PhoBERT and PhoBERT-CNN. The key to XBERT's success lies in its advanced preprocessing methods and a specialized architectural approach. The model utilizes Byte Pair Encoding (BPE) with BPE-Dropout for subword regularization, a technique crucial for managing the complexities of the Vietnamese language. Additionally, the authors have innovatively integrated a unique XBert layer into the RobertaForSequenceClassification framework from the Hugging Face Transformer library.

The superior performance of XBERT is also a result of its effective handling of Vietnamese text. This includes the management of diacritics and the application of Easy Data Augmentation (EDA) techniques, which bolster the training dataset, making the model more robust and capable of understanding the nuances of hate speech in Vietnamese. Despite its strengths, the model has some limitations. Its focus on Vietnamese text limits its direct application to other languages, and there's a possibility that the datasets used might not cover all manifestations of hate speech, possibly affecting the model's comprehensiveness and applicability.

Looking forward, there are several avenues for improvement and expansion. Broadening the dataset to include a more diverse range of hate speech instances could improve the model’s scope and accuracy. Adapting the model for other languages would increase its utility and applicability in a global context. Moreover, integrating newer NLP techniques and algorithms could further enhance the model's performance, making it more efficient and accurate. The research conducted in this paper not only sets a new benchmark in Vietnamese hate speech detection but also provides a framework for developing language-specific hate speech detection models, opening new possibilities for research and development in this critical area of natural language processing.

CONFLICTS OF INTEREST: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

8. REFERENCES

References

[1]. Instagram's 2022 Hate Speech Report (https://www.statista.com/statistics/1275933/global-actioned-hate-speech-content-instagram/)

[2]. Facebook's First Transparency Report on Content Removal URL: https://transparency.fb.com/reports/

[3]. Microsoft's 2023 Digital Civility Index (https://www.microsoft.com/en-us/online-safety/digital-civility)

[4]. Facebook's Language Support (Facebook Support)

[5]. A Study of Hate Speech in Vietnamese Online Communities (https://ieeexplore.ieee.org/document/10084222).

[6]. Sennrich, Rico, et al. "Neural machine translation by joint learning of segmentation and translation." arXiv preprint arXiv:1609.08899 (2016).

[7]. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[8]. Ahmed, Muhammad, et al. "Hate speech detection in online social media: A review." arXiv preprint arXiv:1910.11079 (2019).

[9]. Goodman, Ellen. "The link between online hate speech and real-world violence." The Conversation (2017).

[10]. Awan, Osman, and Mark Dredze. "A quantitative analysis of the emotional impact of online hate speech." arXiv preprint arXiv:1811.02003 (2018).

[11]. Nguyen, Duc, and Minh Tan Tran. "Exploring the impact of BPE-dropout on Vietnamese word segmentation and language modeling." https://arxiv.org/pdf/2003.00744.pdf

[12]. MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions | PLOS ONE

[13]. Tontodimamma, A., & Chakraborty, P. (2020). Knowledge Structure of Hate Speech Research: A Bibliometric Analysis. Journal of Information Science, 46(5), 652-668.https://link.springer.com/article/10.1007/s11192-020-03737-6

[14]. Hoang, T. A., Nguyen, M. L., & Nguyen, T. V. (2021). ViHOS: A Dataset for Vietnamese Hate and Offensive Spans Detection. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing . https://arxiv.org/abs/2301.10186

[15]. Nguyen, D. T., & Nguyen, D. Q. (2022). ViHSD: A Corpus for Vietnamese Hate Speech Detection on Social Networks. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing .https://github.com/sonlam1102/vihsd

[16]. https://paperswithcode.com/task/hate-speech-detection

[17]. https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech

[18]. Jitendra Singh Malik, Guansong Pang, Anton van den Hengel (2022). Deep Learning for Hate Speech Detection: A Comparative Study. https://arxiv.org/abs/2202.09517

[19]. Sanjay Kumar; Abhishek Nagar; Akash Kumar; Amar Singh (2022). Hate Speech Detection:A Survey. https://ieeexplore.ieee.org/document/10074044/authors

[20]. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0221152

[21]. Arun Kumar Yadav, Mohit Kumar, Abhishek Kumar, Shivani, Kusum & Divakar Yadav(2023). Hate speech recognition in multilingual text: hinglish documents. https://link.springer.com/article/10.1007/s41870-023-01211-z

[22]. Fatimah Alkomah and Xiaogang Ma (2022).A Literature Review of Textual Hate Speech Detection Methods and Datasets. https://doi.org/10.3390/info13060273

[23]. Khanh Q. Tran, An T. Nguyen, Phu Gia Hoang, Canh Duc Luu, Trong-Hop Do, Kiet Van Nguyen(2022). Vietnamese Hate and Offensive Detection using PhoBERT-CNN and Social Media Streaming Data. https://arxiv.org/abs/2206.00524

[24]. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach.https://arxiv.org/abs/1907.11692

[25]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[26]. Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton (2016) Layer Normalization. https://arxiv.org/abs/1607.06450

[27]. Pierre Baldi, Peter J. Sadowski . Understanding Dropout

[28]. Marzieh Mozafari, Reza Farahbakhsh, Noel Crespi (2019) A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. https://arxiv.org/abs/1910.12574

[29]. Tommaso Caselli, Valerio Basile, Jelena Mitrović, Michael Granitzer (2021) HateBERT: Retraining BERT for Abusive Language Detection in English https://aclanthology.org/2021.woah-1.3/

[30]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org/contents/rnn.html

[31]. Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks.

[32]. Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv preprint arXiv:1506.00019

[33]. Ahmed Cherif Mazari, Nesrine Boudoukhani & Abdelhamid Djeffal (2023) BERT-based ensemble learning for multi-aspect hate speech detection. https://link.springer.com/article/10.1007/s10586-022-03956-x

[34]. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[35]. Olah, C. (2015). Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[36]. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451-2471. https://doi.org/10.1162/089976600300015015

[37]. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451-2471. https://doi.org/10.1162/089976600300015015

[38]. Vaswani, A., et al. (2017). Attention Is All You Need - The original paper on Transformers.

[39]. Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692. https://arxiv.org/abs/1907.11692

[40]. Facebook AI's official RoBERTa announcement and details . https://ai.meta.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/