VIETNAM NATIONAL UNIVERSITY HCMC
UNIVERSITY OF INFORMATION TECHNOLOGY

PROJECT REPORT

CS221.N11.KHCL

SENTIMENT ANALYSIS OF LODGING SERVICE REVIEW

Ngày 1 tháng 3 năm 2023

Member:

Cao Tuấn Anh 19520008

Nguyễn Gia Thống 19520993

TABLE OF CONTENTS

1 Introduction

In recent years, people have increasingly shared their opinions and attitudes about products and services on social media platforms, including e-commerce websites, social networking sites, and news outlets. As a result, there is now an overwhelming amount of data being transmitted through these channels, and extracting subjective information from this data has become a significant challenge. This is where sentiment analysis comes into play, as it has emerged as a valuable tool in understanding and improving customer experiences in various industries.

Sentiment analysis involves the process of identifying and extracting subjective information from text data. It has various applications, including social listening tools for sentiment, trend, and product/campaign impact monitoring, as well as data analysis. This tool helps businesses gain insights, have a vision, and make data-driven decisions. The lodging sector is one industry where sentiment analysis has shown great potential. Customer satisfaction is critical to the success of a business in this sector, and sentiment analysis can provide insights into customer opinions and feedback. As a result, the group's report will focus on sentiment analysis specifically for hotel and lodging reviews.

The report is structured into five parts. Part 1 provides an introduction to the sentiment analysis problem and explains the data flow, input-output pipeline, and data preprocessing techniques. We also implement, compare, and evaluate the Natural Language Processing (NLP) task Part-Of-Speech (POS) tagging as our Word Segmentation process. We explain the theory and implementation of maximum matching, introduce two libraries (underthesea and VNCoreNLP), and compare their performance in the POS tagging problem with our handcrafted dataset. This section also covers the word representation techniques used in our analysis, including Count Vectorization and TF-IDF, and the classifiers we used, including Random Forest, SVM, and Naive Bayes.

Part 2 focuses on exploratory data analysis (EDA) of the real-world dataset for the sentiment analysis problem. We analyze a lodging service review dataset from FPT Quy Nhon AI Challenge and present our observations.

Part 3 summarizes our findings from the EDA dataset and highlights the issue of data imbalance. To address this issue, we propose a data augmentation approach using PhoBERT's MLM task to generate meaningful review sentences by randomly replacing 15% of the tokens with Noun POS tags. This section provides a quick overview of the BERT and PhoBERT models, our pipeline solution, and the results of the data augmentation process.

Finally, Part 4 will present the results from our experimentation, comparing and evaluating the effectiveness of classifier models with word representation feature extraction, tokenization, and data augmentation. The aim of this report is to provide valuable insights and recommendations for improving customer experiences in the lodging sector through sentiment analysis.

Task Assignment

Bảng 1: Task Assignment

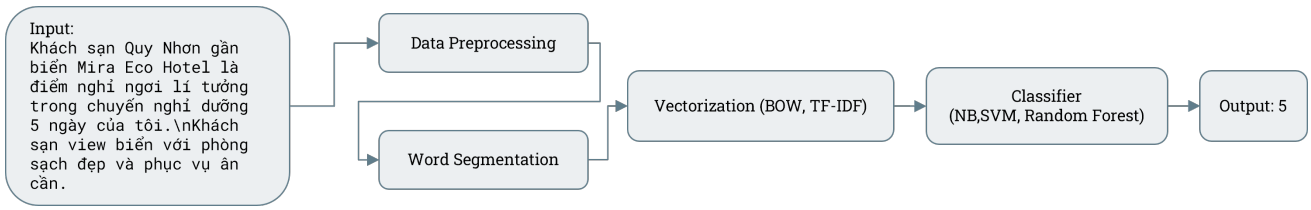
ID	Name	Work
19520008	Cao Tuấn Anh	Word Segmentation using Longest Matching algorithm, VNCoreNLP and Underthesea library, Comparison and evaluation of word segmentation results, Data Augmentation, Data Preprocessing, Observation on result of vectorization methods and POS tagging method on sentiment analysis
19520993	Nguyễn Gia Thống	Word Segmentation using Longest Matching algorithm, Data Analysis, Vectorization, Classifcation models, Data Augumentation, Create dataset with 50 manual POS labeled sentences, Observation on result of data augmentation on sentiment analysis

Bảng 2: Mức độ hoàn thành công việc

ID	Name	Tỷ lệ hoàn thành	Notes
19520008	Cao Tuấn Anh	100%	
19520993	Nguyễn Gia Thống	100%	

2 Sentiment Analysis

Sentiment classification is a form of text classification in which a piece of text has to be classified into one of the predefined sentiment classes. It is a supervised machine learning problem. In binary sentiment classification, the possible classes are positive and negative. With our work, this can be defined as a fine-grained sentiment classification, where there are five classes (very negative, negative, neutral, positive, and very positive) represent by a corresponding numerical range from 1 to 5.
Our sentiment analysis pipeline consists of 4 main steps: data preprocessing, word segmentation, word representation, and classification. In the first step, we perform several preprocessing techniques on the input text. The resulting preprocessed text is then fed into the next step, word segmentation, where we use our own maximum matching implementation as well as two external libraries, Underthesea and VNCoreNLP for part-of-speech tagging.
After word segmentation, we move on to the word representation step, where we use classical methods such as Count Vectorization and TF-IDF to represent each word in the preprocessed text as a numerical vector. Finally, in the classification step, we use three different machine learning models - Random Forest, SVM, and Naive Bayes - to classify the sentiment of the input text based on the vectorized word representation.



Hình 1: Pipeline architecture for our sentiment analysis system

Overall, our sentiment analysis pipeline takes in an input text and outputs a sentiment classification, which can fall into one of five classes ranging from very negative to very positive. This pipeline can be seen here, where an example input text is shown going through each step of the process before arriving at a final sentiment classification. In the following sections, we will delve into each step of the pipeline in more detail.

2.1 Data Preprocessing

In order to extract meaningful insights from the text data in hotel and resort reviews, it is necessary to preprocess the text. Text preprocessing involves replacing emoticons and removing emojis, removing special characters, removing punctuation, handling numbers and normalizing annotations, lowercasing all strings, replacing elongated characters, and removing stop words. These techniques help to reduce noise in the data in order to extract meaningful feature and improve the classification task result.

```
#emoji
text="5,Cái giường nó bị dính qá 😋 ngủ mà k tỉnh dc kkk"
print(text)

print(strip_emoji(text))
```

```
5,Cái giường nó bị dính qá 😋 ngủ mà k tỉnh dc kkk
5,Cái giường nó bị dính qá  ngủ mà k tỉnh dc kkk
```

- Replacing Emoticons and removing Emojis: Replacing emoticons with their associated meanings is a common technique used in text preprocessing for sentiment analysis. However, replacing emojis may not be as straightforward due to the large number of possible emojis and the difficulty in identifying their sentiment meanings. Therefore, it is common practice to remove emojis instead of replacing them with their meanings. In our case, we have a list of 19 specific emoticons to encode and replace in our function, but we have chosen not to replace emojis in order to avoid the challenge of encoding a large number of emojis and their potential sentiment meanings.
- Removing special characters: Special characters such as leftover symbols and unicode characters can be considered as unwanted noise during the crawling process and are not useful for machines in the classification of text data.
- Removing punctuation: Punctuation marks are used to express their sentiments and emotions. But while these may be useful for humans, they can be less useful for machines performing sentiment analysis. Removing punctuation marks is a common practice in sentiment analysis tasks.

```
text="Phòng gia đình nhưng bị tách thành 2 phòng ghép với 4 giường nên kg thích lắm với cùng 1 phòng"
print(text)
print(remove_number(text))
```

```
Phòng gia đình nhưng bị tách thành 2 phòng ghép với 4 giường nên kg thích lắm với cùng 1 phòng
Phòng gia đình nhưng bị tách thành  phòng ghép với  giường nên kg thích lắm với cùng  phòng
```

- Removing numbers: Text reviews may contain unwanted numbers that are not useful for machines performing sentiment analysis. The standard approach is to remove them, though this can result in the loss of useful information.

```
text="Chấm 10đ cho Fleur Hotel Quy Nhơn* Ưu điểm: ks đẹp, đối diện biển,
print(normalize_annotatation(text))
```

```
Chấm 10đ cho Fleur Hotel Quy Nhơn* Ưu điểm: khách sạn đẹp, đối diện biển, buffet sáng
```

- Normalizing annotations: Reviewers often use abbreviations and shorthand to express their opinions, which can pose a challenge for accurate sentiment analysis. To address this, we apply a normalization step to replace these informal expressions with their actual meanings. In our approach, we leverage a list of common Vietnamese abbreviations and shorthand, as well as popular terms used in typical Vietnamese lodging service discourse, such as 'ks' for 'khách sạn'.

- Lowercasing all strings: In a corpus, sentences often contain words with different capitalizations. However, having different copies of the same word with different capitalizations can create problems during the classification task and lower the overall performance. To address this issue, a common preprocessing step is to convert all letters to lowercase. By doing so, we can avoid the issue of diversity of capitalization within the corpus and improve the performance of our classification model.

```
text="Phòng sạch sẽ, thơm, chị lao công cũng dễ thương hoà đồng. Cô chú chủ cực kỳ cute luôn á mọi người. View phòng đẹp, xung quanh yên tĩnh. Mọi người nên ghé nhaaaa."
print(text)
text=remove_similarletter(text)
print(text)
```

```
Phòng sạch sẽ, thơm, chị lao công cũng dễ thương hoà đồng. Cô chú chủ cực kỳ cute luôn á mọi người. View phòng đẹp, xung quanh yên tĩnh. Mọi người nên ghé nhaaaa.
Phòng sạch sẽ, thơm, chị lao công cũng dễ thương hoà đồng. Cô chú chủ cực kỳ cute luôn á mọi người. View phòng đẹp, xung quanh yên tĩnh. Mọi người nên ghé nha.
```

- Replacing elongated characters: reviewers may intentionally use elongated words to express or exaggerate their emotion, such as 'nhaaaa' or 'yêêêuuu'. It is important to replace these elongated words with their base word to ensure that sentiment analysis treats them as the same word.

```
#remove stopword
remove_VN_stopwords("Phòng gia đình nhưng bị tách thành 2 phòng ghép với 4 giường nên kg thích lắm với cùng 1 phòng")
```

```
'Phòng gia đình tách thành 2 phòng ghép 4 giường kg lắm 1 phòn'
```

- Removing stop words: In the dataset, there may be many words that do not have critical significance and are present in high frequency. These words do not improve the accuracy of sentiment analysis and are recommended to be removed. Common stop words include "là", "và", "của", "cho", and "làm", among others.

2.2 Word Segmentation

Word segmentation is a process of dividing a continuous sequence of written text into separate words or units of meaning. It is a crucial step in natural language processing and text analysis as it helps to extract the meaning from the text and enables the computer to understand and process human language.
Word segmentation is an important technique used in many areas such as information retrieval, machine translation, speech recognition, and text-to-speech synthesis. It helps to improve the accuracy of these applications and allows computers to understand and process human language more effectively.
In this project, we experimented with three methods of word segmentation:
Each method was evaluated and compared based on their accuracy and performance in segmenting Vietnamese text.

1. Word segmentation using Longest Matching algorithm
2. Word segmentation using the VNCoreNLP library
3. Word segmentation using the Underthesea library

2.2.1 Data Preparing:

The dataset consists of 50 sentences.

> *"Đồ ăn hải sản tươi ngon, các món canh nấu rất tuyệt."*
>
> *"Mình hay ghé nhà hàng Ngon ở Trần Hưng Đạo để dùng bữa với bạn bè và đối tác."*
>
> *"Nhà xe dễ thương, bác tài dễ thương, xe chạy đúng giờ, nằm đúng chỗ đã đặt, có thêm sữa uống cho hành khách."*
>
> *"Vị trí đẹp, quản lý bộ phận lễ tân nhiệt tình, tận tình, chu đáo."*
>
> *"Bán rất nhiều đồ đặc sản không chỉ của Bình Định mà cả các tỉnh khác."*

The sentence with the highest word count is 28.

> *"Vị trí nhà khách thuận lợi rất gần biển nên buổi sáng sớm bạn có thể đi bộ ra biển tập thể dục và tắm biển thoải mái."*

The sentence with the lowest word count is 8.

> *" Giao xe nhanh chóng, thủ tục đơn giản."*

2.2.2 Manual tokenization:

Using the Vietnamese dictionary from the VLSP library, we will search each word and if it is a compound word, we will separate the words with _ symbol.

For example:
Đồ ăn hải sản tươi ngon , các món canh nấu rất tuyệt. $\longrightarrow$ *Đồ_ăn hải_sản tươi ngon , các món canh nấu rất tuyệt .*
Moreover, during the word segmentation process, there are some ambiguities: *đồ ăn, thoáng mát, người dân, xin phép,...*

2.2.3 Word segmentation using Longest Matching algorithm:

Longest Matching algorithm is a simple word segmentation technique that works by finding the longest matching word in a given dictionary or vocabulary. The algorithm starts from the beginning of a sentence and tries to match the longest possible word from the dictionary. If there is no match, it moves one character to the right and tries again. This process continues until the end of the sentence is reached.

```
// Chỉ số của từ hiện tại ở đầu câu
current_index = 0:
    // Kết thúc khi lặp qua các từ của câu
    while (current_index + 1) != len(text):
        // Xét từ hiện tại với 2 từ sau nó trong từ điển tri_grams
        check if {"word, word+1, word+2"} is in tri_grams
            // 3 từ đó tạo thành từ ghép
            if true: take {word, word1, word2; current_word_id += 3}
        // Xét từ hiện tại với tiếng sau trong từ điển trong từ điển bi_grams
        else check if: take {"word, word+1"} is in bi_grams
            // 2 tiếng đó tạo thành từ ghép
            if true: take {word, word1; current_word_id += 2}
        // tiếng đó là từ đơn
        else => {take word; current_word_id += 1}
```

After inputting the data into the Longest Matching algorithm, we will get the following result:

```
Quần_áo
thời_trang
,
nhân_viên
nhiệt_tình
,
cửa_hàng
đẹp
,
có
nhiều
hình_thức
thanh_toán
tiện_lợi
.

Chị
chủ
shop
nhiệt_tình
,
thân_thiện
,
gói
quà
tặng
theo
nhiều
phong_cách
tuyệt_vời
.
```

However, there are still some advantages and disadvantages of using the Longest Matching algorithm:

- Advantages:
  - Simple implementation.
  - Does not require training data.
- Disadvantages:
  - Dependent on the dictionary.
  - Cannot handle ambiguity issues.

2.2.4 Word segmentation using the VNCoreNLP library:

VNCoreNLP is an open-source natural language processing (NLP) toolkit for the Vietnamese language. It is developed and maintained by the Vietnamese Language and Speech Processing (VLSP) research group.
VNCoreNLP provides a wide range of NLP functionalities, including word segmentation, part-of-speech tagging, named entity recognition, dependency parsing, and sentiment analysis. It is built on top of Stanford CoreNLP, a popular NLP toolkit for the English language, and uses machine learning algorithms to perform various NLP tasks.
VNCoreNLP has been widely used in many applications, such as:

- Word Segmentation
- Pos tagging
- Dependency parsing
- Text classification
- Information extraction
- Machine translation
- Speech recognition

The results we obtained after using the VNCoreNLP library to process the data are as follows:

```
Nhà_xe dễ_thương , bác tài dễ_thương , xe chạy đúng giờ , nằm đúng chỗ đã đặt , có thêm sữa uống cho hành_khách .
Mình đã đến cửa_hàng mình thấy hàng_hoá rất đa_dạng , giá_cả rất phải_chăng , đồ_ăn được .
Nhà_hàng hải_sản Hoàng_Thao cũng rất là ổn và đẹp .
Tất_niên cuối năm , tiệc thôi_nôi , sinh_nhật hay bất_cứ tiệc gì đi_chăng_nữa .
Đây là một nhà_hàng địa_phương và rất nổi_tiếng .
Đồ_ăn rất vừa_miệng , không bị ngọt như khách_sạn khác .
```

2.2.5 Word segmentation using the Underthesea library:

Underthesea is a Vietnamese natural language processing (NLP) toolkit developed by AI VIETNAM Research Development Joint Stock Company. It provides a wide range of features for Vietnamese text processing such as word segmentation, part-of-speech tagging, named entity recognition, dependency parsing, and text classification.
Underthesea uses a combination of machine learning algorithms and rule-based methods to achieve high accuracy in its NLP tasks. It is based on the CRF (Conditional Random Field) algorithm and uses a variety of pre-trained models to process Vietnamese text.
Similarly to VNCoreNLP, after installing the underthesea library and processing the data, we will obtain the result as follows:

```
'Đồ_ăn hải_sản tươi ngon , các món canh nấu rất tuyệt .',
'Mình hay ghé nhà_hàng Ngon ở Trần_Hưng_Đạo để dùng bữa với bạn_bè và đối_tác .',
'Đồ_ăn tươi ngon chế_biến vừa_vặn , giá_cả hợp_lý cạnh quán Nghĩa_Ghẹ .',
'Đồ_ăn ngon , hải_sản tươi , quán sạch , nhân_viên  dễ_thương  .',
'Lần đầu_tiên ghé ăn nhưng rất hài_lòng , chắc_chắn là sẽ quay lại .',
'Quán bán đủ món cháo lươn miến lươn , bún phở đầy_đủ , có nhiều lựa_chọn .',
'Sân_vận_động đã được đầu_tư cải_tạo lại , sân_cỏ đẹp , khán_đài được làm lại đẹp hơn trước rất nhiều .',
'Quảng_trường xanh , rộng và thoáng mát của thành_phố biển Quy_Nhơn , rất thích_hợp cho việc đi bộ .',
'Một tổ_hợp khoa_học , nơi tổ_chức hội_thảo khoa_học thu_hút nhân_tài thế_giới .',
'Khu_vui_chơi dành cho mọi_người có nhiều dụng_cụ tập_thể_dục , cây hoa được chăm_sóc tốt .',
'Thiên_nhiên hoang_sơ , chưa được khai_phá nhiều nên vẫn còn nhiều nét mộc_mạc .',
'Một địa_điểm hay cho các học_sinh thăm quan để hiểu sâu hơn giữa lý_thuyết và thực_tế các môn_học .',
'Không_gian hoạt_động giáo_dục khoa_học , sáng_tạo cho thiếu_nhi thật tuyệt_vời !',
```

2.2.6 Result

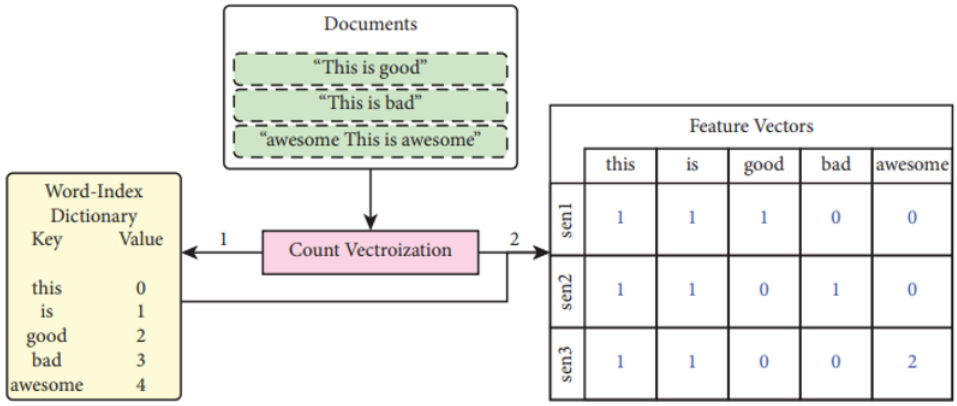
	Thủ công	Longest Matching	**VNCoreNLP**	Underthesea
Số từ ghép	206	197	**211**	213
Accuracy		0.93	**0.94**	0.89

Bảng 3: Comparison of word segmentation methods

2.3 Word Representation

After the preprocessed text went through the word segmentation step, our next goal is to transform each word into a numerical vector. This is necessary as machine learning algorithms require numerical inputs to work properly. In this section, we present two classical methods: Count Vectorization and TF-IDF, which enable us to capture the meaning and significance of every word in the text.

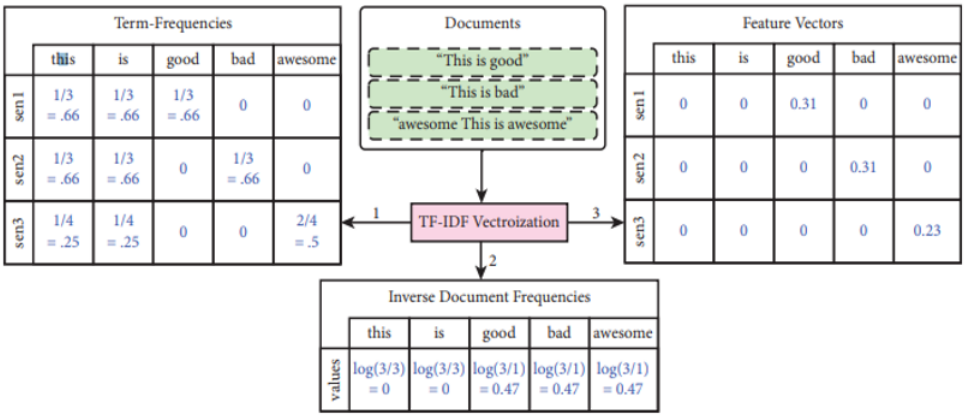
2.3.1 Count Vectorization



Hình 2: Illustration of Count vectorization

Count vectorization is a technique that involves transforming each sentence into a numerical vector by counting the frequency of each word in the sentence and setting the count as the value for that word's corresponding dimension. As the figure above illustrated, to perform count vectorization, we first generate a dictionary of word-indices and then transform each sentence into a vector in the vector space using this dictionary. However, stop words, which are the most frequent words in a language, may dominate the count and reduce the impact of less frequent words, resulting in larger values for their respective dimensions.

2.3.2 Term Frequency-Inverse Document Frequency (TF-IDF)



Hình 3: Illustration of TF*IDF vectorization.

On the other hand, TF-IDF (term frequency-inverse document frequency) offers a tradeoff between high and low-frequency words. It calculates the product of the term frequency relative to a document (TF) and the inverse document frequency of the same term in the corpus (IDF). Mathematically, we can represent the TF and IDF using equations. The figure above illustrates how these equations are used to calculate TF-IDF values. This approach enables us to capture the relative importance of each word in the text, rather than just the frequency of occurrence.

$$\mathrm{TF}(t,d) = \frac{\text{frequency of term } t \text{ in document } d}{\text{total number of terms in the document}} \tag{1}$$

$$= \frac{f_{t,d}}{\sum_{t \le d} f_{t,d}} \tag{2}$$

$$\mathrm{IDF}(t,D) = \log \frac{\text{total number of documents in the dataset}}{\text{number of documents containing the term } t} \tag{3}$$

$$= \log \frac{|D|}{\left|\{d \in D : t' \in d\}\right|} \tag{4}$$

2.4 Classification

After extracting numerical features from the text, our next goal is to build models that can classify the text into the relevant categories. In this section, we explore three popular machine learning classifiers: Naive Bayes, Random Forest, and Support Vector Machine (SVM).

2.4.1 Naive Bayes

Naive Bayes is a well-known machine learning algorithm used for classification tasks. It classifies sentences based on conditional probability, which is calculated using Bayes' theorem. However, the algorithm assumes that the features are conditionally independent of each other. The formula used by the Naive Bayes algorithm to classify a sentence X is shown below:

$$f(C_k|X) = \frac{p(X|C_k)\,p(C_k)}{p(X)}. \tag{5}$$

The cosine similarity between two document vectors (A and B) can be calculated using the following equation:

$$\text{Similarly}(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{6}$$

This is expanded with reference to the individual features ($X = \{x_0, x_1, ..., x_n\}$):

$$f(C_k|x_0, x_1, \ldots, x_n) = p(C_k)\,p(x_0|C_k)\,p(x_1|C_k)\cdots p(x_n|C_k) \tag{7}$$

$$= p(C_k)\prod_{i=0}^{n} p(x_i|C_k) \tag{8}$$

When the documents are normalized and transformed using the TF * IDF vectorization technique, the features are no longer discrete. In this case, we use a variant of Naive Bayes called Gaussian Naive Bayes, which assumes that the continuous features follow a Gaussian distribution. The substitution of $(x_0|C_k)$ in the Gaussian Naive Bayes is defined by the equation below:

$$p(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\left((v-\mu_k)^2/2\sigma_k^2\right)} \tag{9}$$

To classify the target class y, we choose the class with the highest probability, which can be represented mathematically as follows:

$$\widehat{y} = \underset{k\in\{1,\ldots,K\}}{\arg\max}\, p(C_k)\prod_{i=1}^{n} p(x_i|C_k). \tag{10}$$

Here, K represents the set of classes.

2.4.2 Random Forest

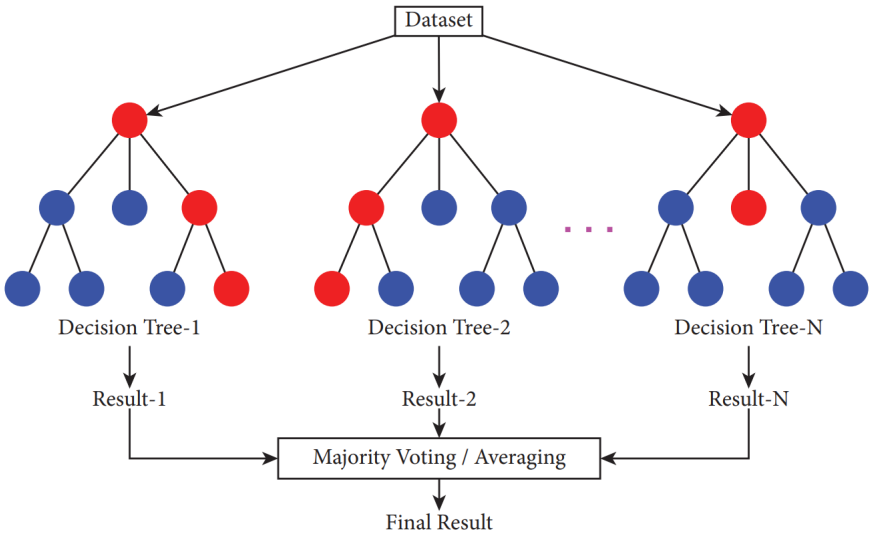
Random Forest is a popular ensemble approach for ML classification that is based on Decision Trees. Unlike using a single decision tree, Random Forest draws multiple decision trees by bootstrapping random samples from the training data. The decision tree algorithm in Random Forest can be divided into three main steps: Feature selection, Decision tree generation, and Pruning.

1. Feature selection involves selecting important features from the text vectors to be used as decision criteria. These selected features will determine the shape of the decision tree and greatly impact the classification results.

2. Decision tree generation step, the algorithm makes the main decisions. The root node is a specific word sequence that has the highest probability of correctly classifying a given category. The algorithm then creates subsequent child nodes by dividing them into left and right sub-trees based on the selected feature. If the coefficient is not zero or the word sequence has no child sequence, the decision will stop; if it is not zero and not unique, the decision will continue in the possible category.

3. Pruning: the decision tree can become very large, leading to overfitting. To overcome this issue, the size of the decision tree is controlled at each layer, the minimum number of samples required to form a leaf node is set, and the minimum weight of the leaf node is adjusted to prevent overfitting.

Figure below shows how the RF classifier works and outputs a final class from all of the DTs.



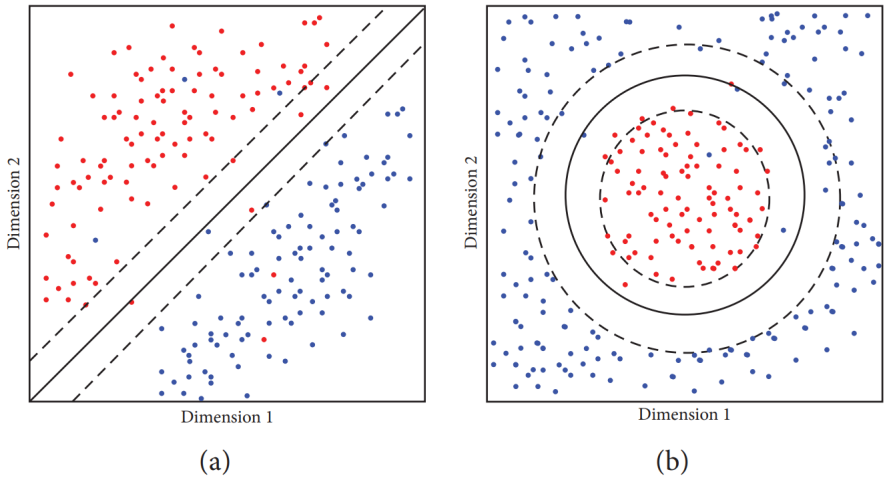
Hình 4: Illustration of random forest trees

2.4.3 Support Vector Machine

Support Vector Machine (SVM) is a common classifier in traditional ML algorithms, particularly well-suited for complex and imbalanced classification tasks with small or medium-sized datasets. SVM is a non-probabilistic binary linear classification algorithm that works by plotting the training data in multi-dimensional space, then categorizing the classes with a hyperplane. If the classes can't be separated linearly in this space, the algorithm adds a new dimension to create a linear separation. SVM can be used for linear or nonlinear classification, but the basic SVM that fits a hyperplane is commonly known as linear-SVM. The equation below provides the mathematical definition of linear-SVM.

$$\tilde{y}_i = \begin{cases} 1, & \text{if } w^T \cdot X_i - b \geq 1, \\ -1, & \text{if } w^T \cdot X_i - b < 1. \end{cases} \tag{11}$$

In our project, we used both linear-SVM and radial basis function (rbf) SVM. The main goal of rbf-SVM is to fit a circular boundary margin for nonlinear datasets.



Hình 5: Illustration of linearity and nonlinearity in data. (a) The linear-SVM corresponds to fitting a hyperplane when the data reflects its shape as depicted in the subfigure a, (b) whereas the subfigure b reflects the situation optimal for rbf-based SVM

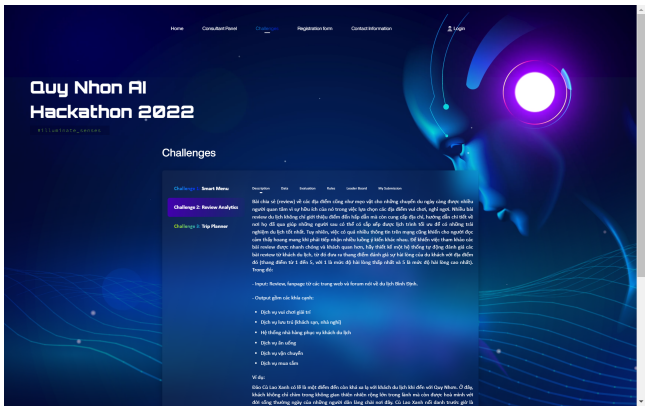
The figure above illustrates this concept, with (a) representing the linear-SVM and (b) demonstrating the need for the rbf kernel trick when a hyperplane isn't suitable for separating the data into two distinct parts. The red and blue dots in the figure represent separate classes, and the solid grey line is the decision boundary, with the dots on the dashed line referred to as support vectors.

3 Analysis of Real-world Dataset: FPT Quy Nhon AI Hackathon 2022

In this chapter, we analyze a lodging service review dataset from FPT Quy Nhon AI Challenge and present our observations.

3.1 Dataset Description

The dataset for our problem is based on the dataset from Challenge 2 Review Analytics, which is an aspect-based sentiment analysis problem in the FPT Quy Nhon AI Hackathon 2022. The data consists of around 3000-4000 reviews about 5 aspects in Quy Nhon: entertainment services, lodging services (hotels, guesthouses), restaurant systems serving tourists, food services, transportation services, and shopping services. From this dataset, our team filtered the data to fit the sentiment analysis problem about lodging services, as we found it to be a realistic and challenging dataset.



Hình 6: The competition's official website

3.1.1 Filtering Process

```
aspect_df['luu_tru']
```

	Review	giai_tri	luu_tru	nha_hang	an_uong	di_chuyen	mua_sam
15	Phòng có giường lớn, êm đẹp, sạch sẽ,... Lễ tân is the best. Buffet ngon, thanh đạm. Không gian sảnh sang chảnh tha hồ check in. Cảm ơn Phương Linh đã giúp tụi mình tìm 1 căn phòng vừa ý. Nếu có dịp mình sẽ chọn căn Pano.	0	5	0	0	0	0
23	Gần biển,gần chỗ ăn uống.Nhân viên dễ thương	0	4	0	0	0	0
31	Nhân viên nhiệt tình, phong cách hiện đại. Phòng ban công view phố đẹp, phòng mới sạch sẽ.	0	5	0	0	0	0
35	Gia đình mình có trẻ con nên ở phòng đôi k có bồn tắm, nhưng khách sạn linh động cho mình phòng đôi có bồn tắm trẻ con tắm thoải mái.Nhân viên lễ tân nhiệt tình, lễ phép. VỊ TRÍ thuận tiện, đi bộ 2p là đến vincom, 3p là ra đến biển, khá gần đường chính.	0	5	0	0	5	0
49	Phòng thoải mái, khá rộng, có bếp và dụng cụ có thể nấu ăn chung. Tiện nghi đầy đủ giá tốt	0	5	0	0	0	0
...	...	...	...	...	...	...	...
3673	Khung cảnh thoáng mát, view trực tiếp ra bãi biển. Có lối đi bộ và ghế đá sạch sẽ, không có rác tuy nhiên hơi ít thùng rác và thùng rác hơi nhỏ, cần bổ sung thêm.\n	0	5	0	0	0	0
3674	Phòng ốc sạch sẽ, view khá xịn xò, có thể ngắm trọn vẹn bình minh bsang. Ngay bãi tắm, đi vào trung tâm cũng gần	0	5	0	0	0	0
3680	Ks rất tuyệt. Nhân viên nhiệt tình. Phòng ốc rất sạch sẽ. View cực đẹp. Sẽ quay trở lại lần nữa nếu du lịch Nha Trang. Đánh giá 10 điểm.	5	5	0	0	0	0
3682	Nhân viên rất nhiệt tình, dễ thương. Phòng ốc rất sạch sẽ, giống hình ảnh đăng hoàng. Mình quên hỏi chơi không có chụp lại hình. Mong cho Green Meadow giữ vững chất lượng như này.	0	5	0	0	0	0
3685	Phòng đẹp, sạch sẽ đầy đủ tiện nghi, nằm trong khu biệt thự nên rất yên tĩnh, chủ thân thiện, sẽ quay lại nếu có dịp.	0	5	0	0	0	0

755 rows × 7 columns

Hình 7: Sample from the original dataset

```
for item in aspects:
  print('{aspect} has {count} reviews'.format(aspect = item, count = aspect_df[item].Review.count()))
  print('which consists of the follow ',query_result[item].groupby(by='review_has_other_sentiments').sum()['count'])
```

```
giai_tri has 557 reviews
which consists of the follow  review_has_other_sentiments
False    301
True     256
Name: count, dtype: int64
luu_tru has 755 reviews
which consists of the follow  review_has_other_sentiments
False    643
True     112
Name: count, dtype: int64
nha_hang has 299 reviews
which consists of the follow  review_has_other_sentiments
False     63
True     236
Name: count, dtype: int64
an_uong has 1535 reviews
which consists of the follow  review_has_other_sentiments
False    1175
True      360
Name: count, dtype: int64
di_chuyen has 768 reviews
which consists of the follow  review_has_other_sentiments
False    631
True     137
Name: count, dtype: int64
mua_sam has 314 reviews
which consists of the follow  review_has_other_sentiments
False    271
True      43
Name: count, dtype: int64
```

Hình 8: Result of the dataset's aspect survey

Our team filtered the data to fit the sentiment analysis problem about lodging services. The filtering process was based on observations of the original dataset from the competition. Reviews that provide sentiment values for the lodging aspect are also likely to be directly related to that service, such as leaving a review for a hotel, homestay, or resort, at all levels from 1 to 5.
The team's survey results showed that lodging is the aspect with the highest ratio of sentences that only mention that aspect out

of the total number of sentences among the 5 aspects, even with a very naive criterion of considering any sentence with value for another aspect as belonging to multiple aspects.

3.1.2 Dataset Details

Based on that, the team was confident in choosing the lodging aspect from the original dataset as a new dataset for their problem, and obtained 2 datasets:
Train dataset: consists of 755 reviews collected from websites and forums about lodging places (hotels, guesthouses, resorts, etc.) in Binh Dinh. The data includes the review sentence and sentiment value from 1 to 5 (Very dissatisfied / Dissatisfied / Normal / Satisfied / Very satisfied).

```
Label:  1
Mình đã đặt 11 phòng cho kì nghỉ và rất thất vọng.
khách ở từ thứ 3 đến thứ 7, mà khách sạn không hề lau dọn phòng.
Thiếu nhiều vật dụng cơ bản: khăn giấy trên phòng, bàn là hỏi nhân viên nhiều lần mới có do cả Ks có 1 bàn là duy nhất.
Label:  2
Khe vịn mở cửa tủ quần áo hơi cạn nên khi mở hơi khó, dễ gãy móng tay, nên khoét sâu hơn hoặc có nắm cửa bên ngoài.
Cách âm kém, đêm hôm phòng bên cạnh cười nghe rõ. Không có nhân viên hỗ trợ xách đồ. View biển nhưng trước mặt có công trình xây dựng
Vị trí gần biển nhưng do khách sạn nằm trong ngõ nên view biển căn góc không thực sự đẹp như trong hình
Label:  3
Cá nhân mình thấy phòng hơi thiếu sáng
Tiện nghi có một số hạn chế. Không có bàn ủi trong phòng, muốn ủi đồ phải thuê khách sạn. Đối với ks 4 sao thì tôi thấy đây là khuyết điểm
Máy lạnh Daikin nhưng điều khiển chỉ cho tắt mở và tăng giảm nhiệt chứ không điều chỉnh được chế độ quạt và làm lạnh.
Label:  4
Khách sạn gần trung tâm đi lại khá tiện. Phòng ốc xịn xò sạch sẽ. Mình book được phòng view nhìn chếch ra biển khá chill
Vị trí đẹp. Nhân viên thân thiện nhiệt tình. Gần khu vực ăn uống
Chú chú khách sạn và nhân viên siêu dễ thương
Label:  5
Phòng thoải mái, khá rộng, có bếp và dụng cụ có thể nấu ăn chung. Tiện nghi đầy đủ giá tốt
Phòng rộng. sạch sẽ. có thể ở tới 4 người
Tuy ảnh hưởng mùa covid không hỗ trợ được trực tiếp nhưng hỗ trợ online với thái độ thân thiện, nhanh chóng và chính xác.
```

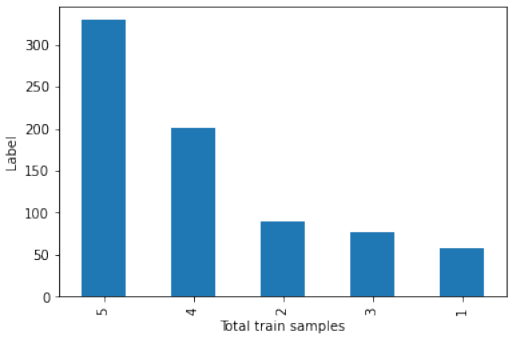
Hình 9: Sample from train dataset

The competition also provided a test dataset, which the team filtered similarly and obtained 242 review sentences. The class distribution for the test dataset is similar to the training dataset

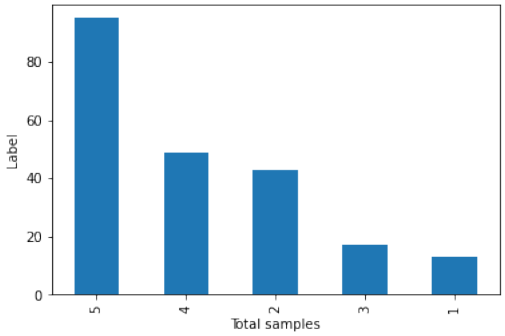
Nằm ở vị trí có mật độ giao thông lúc nào cũng khá cao và đông đúc, nhưng dịch vụ ở khách sạn rất tuyệt vời. Đồ ăn rất ngon dù giá không hề rẻ	5
Rất phù hợp với khách du lịch gia đình, nghỉ dưỡng. - Đồ ăn rất ngon, hàng cao cấp cũng có. cua huỳnh đế sống (tùy hôm), ghẹ bư... - Vị trí tốt, ngay mặt đường bờ biển, nội ngoại thất đẹp - Nhân viên xinh xắn, phục vụ nhiệt tình - Giá cả không phải bình dân, phù hợp gia đình du lịch nghỉ dưỡng	5
Địa điểm mát mẻ. Sạch, hàng hóa nhiều. Ăn uống xem phim giải trí đầy đủ... dịch vụ sửa chữa máy tính nhà cửa điện nước .... nấm sạo	5
Khách sạn đẹp, nhân viên nhiệt tình dễ thương, nước ép rất ngon, có nước, hoa quả, bánh và café tại sảnh.	5
Vị trí đẹp, lễ tân lịch sự mến khách	5
không có điểm nào để phê bình cả	3
Giá phòng và thuê xe máy: Giá phòng 950k/1đêm/6nguoi, Xe máy 150k/1xe/24h Chỗ bọn mình chọn ở là Hostel Carrisbay Quy Nhơn, 08 Ngô Mây. Còn thuê xe máy thì bọn mình tin tưởng lựa chọn dịch vụ cho thuê xe máy quy nhơn của anh Hoài bảo.Mình khuyên các bạn nên thuê chứ đừng bắt taxi đi đây đó vì giá thuê xe nó rẻ hơn nhiều so với taxi. Tụi mình 6 người nên ở phòng có giường tầng, có 4 giường cho 8 người lận, cũng khá thú vị, có ban công rộng rãi ngắm ra biển nữa. Trước khách sạn có quán nước khá ngon và rẻ. (Em pha chế cũng ngon không kém v). Khách sạn ngay trung tâm thành phố, gần biển, quảng trường trung tâm và gần luôn con đường ăn nhậu xuyên màn đêm nữa :))). Ngoài việc không có thang máy ra thì mọi thứ rất ok. Tóm lại mình vừa có chuyến du lịch rất tuyệt vời khi đến Quy Nhơn du lịch	4
Chuyến đi thật dễ chịu và thư giãn. Tổng chi phí chuyến đi khứ hồi cho 2 người là khoảng 2 đô la. 60.	4
Mình vừa có chuyến du lịch 4N3 tại thành phố Quy Nhơn. Mình đang ở homestay Hai chị em nhà, mọi thứ đều rất tuyệt vời, con người ở đây vô cùng dễ mến, thân thiện và dễ mến. Ngày đầu tiên mình di chuyển chủ yếu bằng taxi và cảm thấy khá tốn kém, hôm sau mình muốn thuê xe máy để đi lại, sau một thời gian tìm kiếm trên fb và gg thì mình nhận ra có quá nhiều chỗ cho thuê xe. Không biết gọi điện ở đâu, tối hỏi chị cùng xóm trọ thì chị ấy nói tối gọi điện cho Hoài Bão. Mình đọc review thấy dịch vụ này rất tốt nên cũng tin tưởng và gọi điện. Ấn tượng đầu tiên là bạn giao xe rất nhanh, không biết của hàng bạn có gần nhà không nhưng mình vừa gọi chưa đầy 10 phút là bạn đã mang xe đến. Lúc nhận xe nói chuyện rất cởi mở và nhiệt tình, làm hợp đồng cho thuê xe máy chuyên nghiệp, rồi tự giới thiệu cho mình nhiều quán ăn ngon và nhiều địa điểm hot ở đây. Nhắc tới nếu tới cần bất kỳ thông tin nào, hãy gọi cho cô ấy ngay lập tức và vì vậy tôi sẽ làm phiền cô ấy trong 3 ngày còn lại ở đây. Xin chân thành cảm ơn quý doanh nghiệp đang rất quan tâm, sẽ sớm thành công.	5
Gần bến xe, có rạp chiếu phim, siêu thị, cửa hàng thời trang	5
Địa điểm ngay khu trung tâm. Dễ di chuyển.	5
Đây là khách sạn đẹp, chất lượng 5 sao chuẩn quốc tế. Đội ngũ nhân viên nhiệt tình chuyên nghiệp, không gian sạch thoáng. Nhiều dịch vụ hỗ trợ chất lượng cao. Tuy nhiên không gian rộng và chưa có dịch vụ xe điện để phục vụ khách hàng đa dạng hơn	5

Hình 10: Sample from test dataset

3.1.3 Class Distribution



Hình 11: Class distribution graph of the train dataset



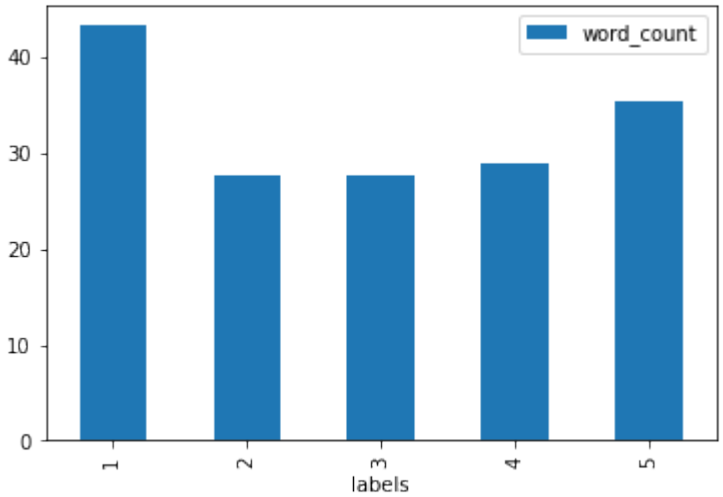
Hình 12: Class distribution graph of the test dataset

The class distribution for the train and test dataste are similar with an imbalanced distribution, only deviate in the number of 2 valued reviews in the test datatset.

3.2 Observations on the Dataset

The team carried out further EDA on this dataset, focusing on the distribution of sentence length and feature engineering.

- Observation on the similarity of the class distribution between test-train datasets: This observation gives us more confidence that our model will perform well on the test dataset since the distribution of the sentiment values roughly similar to the training dataset.
- Observation on the low number of samples in the dataset <1k: Another important observation is that the dataset is relatively small with less than 800 samples in total. This may affect the performance of the model, especially when we want to predict the sentiment value for a new, unseen review. It also poses a challenge in training a model with high accuracy due to the lack of data.
- Observation about the distribution of Sentence Length: The team analyzed the distribution of sentence length in the dataset and found that the dataset consists of long and complex reviews. The average sentence length is greater than 30 words across all 5 sentiment classes, with many outliers throughout the range of 50 to 400 sentences. There is one outlier with a maximum sentence length of 444.



Hình 13: Graph of word count distribution by class

```
1 near_avg[:5]#top 5 cau trung binh
```

	labels	Review	word_count
22	5	Nhân viên phục vụ rất nhiệt tình, giá cả cũng ...	32
48	5	Khách sạn mới, sạch sẽ, ở vị trí thuận lợi. Nh...	32
51	4	Từ nhân viên đến ban quản lý ai đều rất thân t...	34
57	5	JW Marriott đa phần có dịch vụ tốt. Không phải...	31
81	5	Dịch vụ ok.ql thân thiện , Nv phục vụ chuẩn k ...	32

Hình 14: Sample of sentences with the average word count in the dataset

```
1 near_avg[:5]#top 5 cau trung binh
```

	labels	Review	word_count
22	5	Nhân viên phục vụ rất nhiệt tình, giá cả cũng ...	32
48	5	Khách sạn mới, sạch sẽ, ở vị trí thuận lợi. Nh...	32
51	4	Từ nhân viên đến ban quản lý ai đều rất thân t...	34
57	5	JW Marriott đa phần có dịch vụ tốt. Không phải...	31
81	5	Dịch vụ ok.ql thân thiện , Nv phục vụ chuẩn k ...	32

Hình 15: Sentences with the max and min word count value in the dataset

```
1 max_review
```

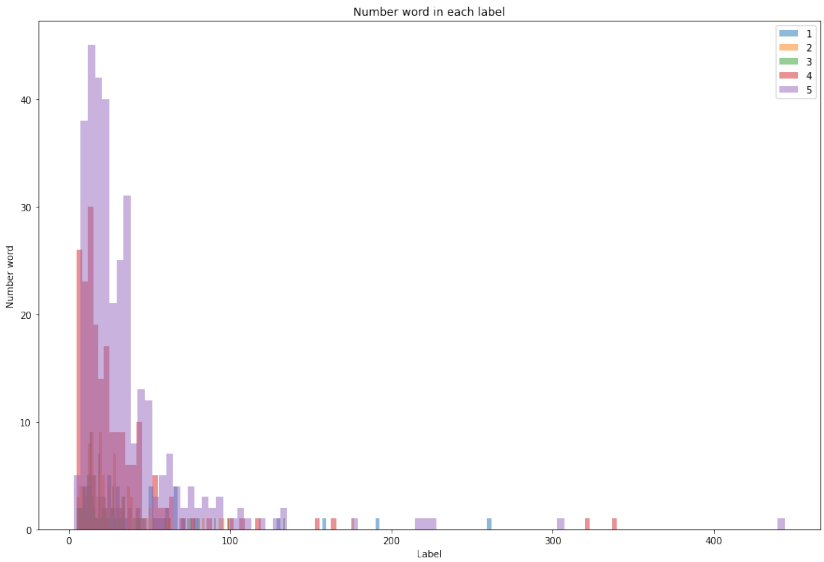
	labels	Review	word_count
355	5	Theo mình thấy là giá khá cao so với các chỗ k...	444

```
1 min_review
```

	labels	Review	word_count
631	5	Phòng sạch sẽ	3

Hình 16: A histogram of word count distribution of the dataset

- Feature Engineering The team conducted feature engineering on the dataset and found no clear pattern in the distribution of sentence length and sentiment.



Hình 17:

3.3 EDA conclusion

After conducting exploratory data analysis on the lodging service review dataset, our team identified two key observations.

- First, the dataset contains long and complex reviews with an average length of over 30 words and many outliers. This suggests that experimenting with different POS tag-based word tokenization methods may be useful for improving the accuracy of our sentiment analysis model.
- Second, the dataset is imbalanced and relatively small, with fewer than 800 samples. To address this issue, we plan to experiment with data augmentation techniques using POS tagging for guided masking language models such as PhoBERT.

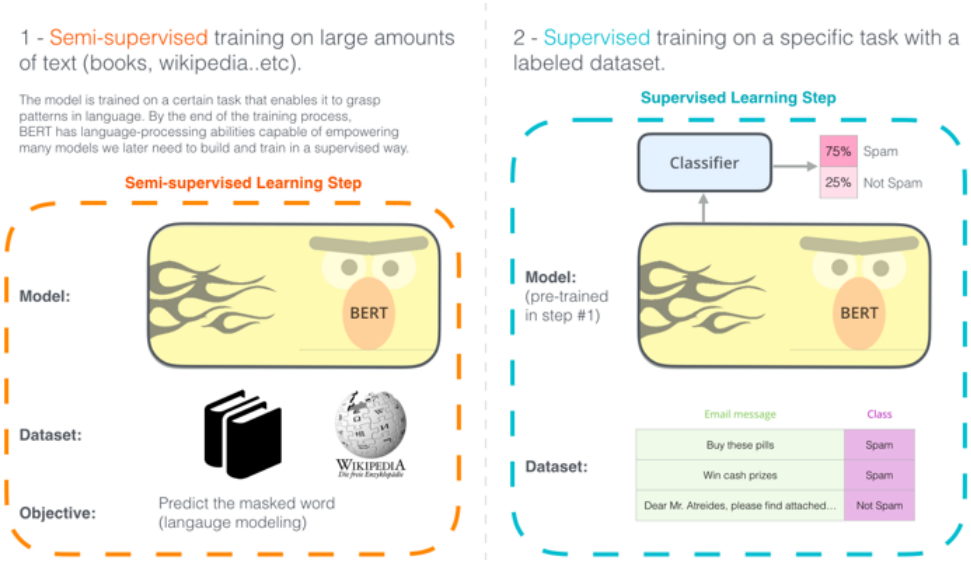
4 Data Augmentation

To address the challenge our dataset has posed as highlighted previously, we explore data augmentation techniques to increase the size of the dataset by using POS tagging for guided masking language model with PhoBERT. This technique involves randomly masking words in the sentences and replacing them with similar words based on their POS tags. By doing so, we can generate new synthetic data that is similar to the original data and can help improve the performance of the model.

4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based neural language model developed by Google AI Language in 2018. BERT has achieved state-of-the-art results on a variety of natural language processing tasks, such as text classification, named entity recognition, and question answering, by leveraging large-scale unsupervised pre-training.

BERT's architecture is based on the Transformer model, which uses a self-attention mechanism to learn contextual relationships between words in a sentence. The Transformer model consists of an encoder and a decoder, but in the case of BERT, only the encoder is used. The encoder consists of a stack of identical layers, each with a multi-head self-attention mechanism and a feedforward neural network. BERT also includes an embedding layer, which maps each word in the input sequence to a high-dimensional vector representation.



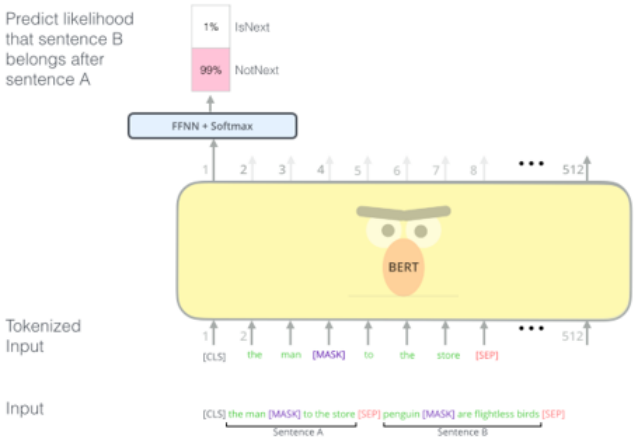
Hình 18: An infographic overview of the BERT pre-train model

BERT is trained using two unsupervised learning tasks: masked language modeling (MLM) and next sentence prediction (NSP). The masked language model (MLM) task is a pre-training task in which a percentage of the tokens in a sentence are randomly masked and the model is trained to predict the masked tokens. Specifically, during pre-training, 15% of the input tokens are randomly masked, and the model is trained to predict the original tokens given the context of the sentence. This task is used to train the model to understand the relationships between words and to learn contextual representations that can be used for downstream natural language processing tasks. NSP is a task where the model is given two sentences and it must predict whether the second sentence follows the first sentence in the original text. BERT uses a transformer encoder architecture, which enables it to capture bidirectional context in a sentence.

4.2 Masked Language Model

Language Modeling is the task of predicting the next word given a sequence of words. In masked language modeling instead of predicting every next token, a percentage of input tokens is masked at random and only those masked tokens are predicted.

The masked words are not always replaced with the masked token – [MASK] because then the masked tokens would never be seen before fine-tuning. Therefore, 15% of the tokens are chosen at random, 80% of the time tokens are actually replaced with the token [MASK], with 10% of the time tokens are either replaced with a random token or left unchanged.



Hình 19: BERT model on the MLM task

4.3 phoBERT

PhoBERT is a variant of BERT that is specifically designed for Vietnamese text. PhoBERT is pre-trained on a large corpus of Vietnamese text using the MLM task, similar to the original BERT model. In addition, PhoBERT uses a technique called subword tokenization to break down Vietnamese words into smaller subword units, which helps the model to better handle the complex morphology of the Vietnamese language. PhoBERT also includes additional pre-training tasks, such as masked sentence prediction and next sentence prediction, which help to improve the model's performance on downstream tasks.

PhoBERT was trained using a large corpus of Vietnamese text, which consisted of approximately 24GB of data. The model was trained using a masked language modeling objective, which involved randomly masking out tokens from the input sequence and training the model to predict the missing tokens. In addition to masked language modeling, phoBERT was also trained using a next sentence prediction task, which involved predicting whether two input sentences were consecutive in the original text.

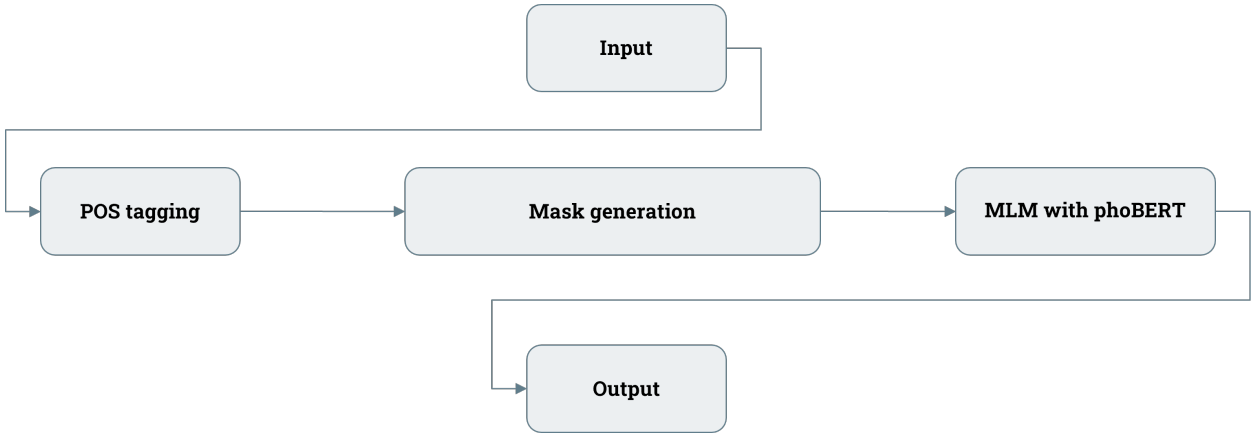
PhoBERT is available in multiple versions, including base and large versions, which differ in terms of the number of parameters in the model. The base version has 110 million parameters, while the large version has 375 million parameters. Both versions are capable of performing a wide range of NLP tasks, including text classification, named entity recognition, and question answering, among others.

One notable aspect of phoBERT is that it was trained using a combination of in-domain and out-of-domain data. This means that the model was trained not only on a large corpus of general Vietnamese text, but also on more specific types of text, such as news articles and social media posts. This approach has been shown to be highly effective in improving the performance of pre-trained language models on domain-specific tasks.

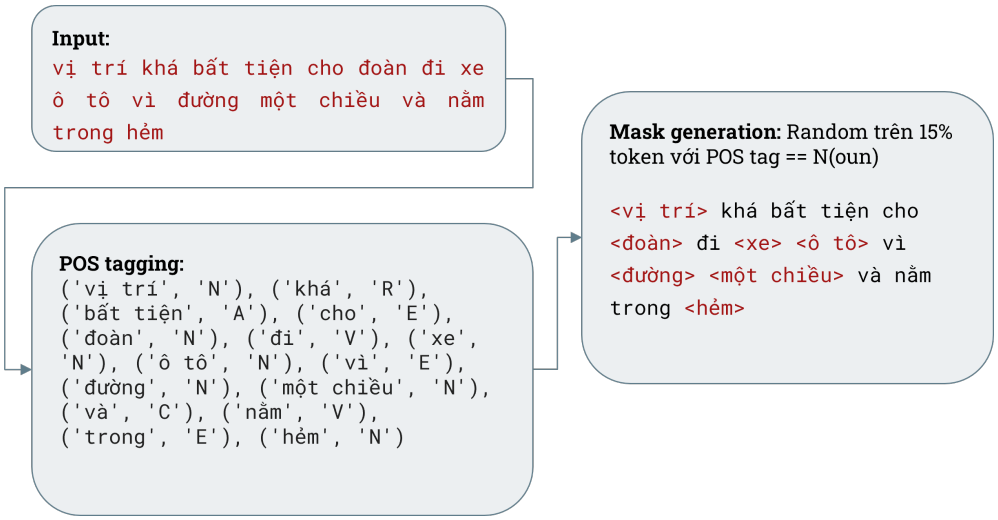
In terms of performance on the MLM task, PhoBERT has achieved state-of-the-art results on a number of benchmarks for Vietnamese text. PhoBERT has also been used as a base model for a number of other natural language processing tasks in Vietnamese, including named entity recognition, sentiment analysis, and machine translation, and has achieved state-of-the-art performance on many of these tasks as well.

4.4 Data augmentation pipeline

The pipeline consists of three steps: POS tagging, mask generation, and MLM with phoBERT.

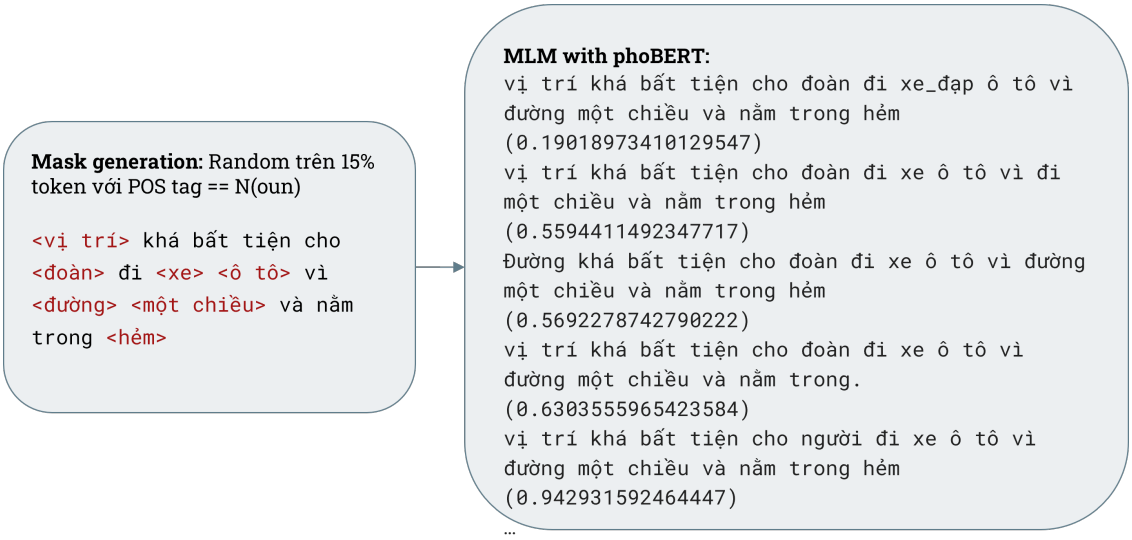


4.4.1 POS tagging and Mask generation



The first step in our data augmentation pipeline is POS tagging. This involves identifying the parts of speech (POS) of each word in the input sentence. Here we use VNCoreNLP as it is the best performer in our previous survey on POS tagging.
In the second step, mask generation is performed on the sentence. Masking involves replacing a selected word with a special [MASK] token, which tells the MLM model to predict what the missing word is. In this step, we randomly mask 15% of the tokens in the sentence that have a POS tag of "N". By masking a random 15% of tokens with Noun POS tags, we are more likely to generate variations of the original sentence that maintain the core meaning and syntactical structure of the sentence while introducing some variations. Additionally, because nouns tend to be more concrete and specific than other parts of speech, masking them can lead to more interesting and diverse augmentations.

4.4.2 MLM with phoBERT



In the third step, MLM is performed on the masked sentence using phoBERT. The MLM task involves training a language model to predict the masked tokens in a sentence. In phoBERT, the masked tokens are replaced with a special token, [MASK], and the model is trained to predict the original token. The output of this step is a list of generated sentences and a score for each sentence. This score is implemented by the Huggingface library with their BERT/phoBERT model from mlm-scoring package.

The pseudo-log-likelihood (PLL) score is a commonly used scoring metric in the masked language modeling task for evaluating the quality of generated text. It is a measure of how well a language model can predict the masked tokens in a given sentence.

Formally, the PLL score for a masked token is calculated as follows:

$$PLL(t_i) = \sum_{w_j \in V} log\ p(t_i = w_j | \mathbf{t}_{-i})$$

Let V be the set of masked tokens in the sentence, where t_i is the masked token, $\mathbf{t}-i$ is the sentence with the masked token removed, and $p(t_i = w_j|\mathbf{t}-i)$ is the probability of the masked token being replaced with token w_j according to the language model. Intuitively, the PLL score measures the log-probability of each possible replacement for the masked token, and sums them up to obtain a score for the entire sentence. Higher PLL scores indicate better quality generated text.
In the context of BERT and phoBERT, PLL scores are used as a proxy for sentence quality. During the MLM task, each masked token in a sentence is replaced with a candidate token, and the PLL score is calculated for each candidate sentence. The sentence with the highest PLL score is selected as the output of the MLM task.
Note that PLL scores are only an approximation of sentence quality, and should not be used as the sole metric for evaluating the performance of a language model. Other metrics, such as human evaluation and downstream task performance, should also be considered.

4.4.3 Pipeline output

MLM with phoBERT:

```
vị trí khá bất tiện cho đoàn đi xe_đạp
ô tô vì đường một chiều và nằm trong
hẻm
(0.19018973410129547)
vị trí khá bất tiện cho đoàn đi xe ô
tô vì đi một chiều và nằm trong hẻm
(0.5594411492347717)
Đường khá bất tiện cho đoàn đi xe ô tô
vì đường một chiều và nằm trong hẻm
(0.5692278742790222)
vị trí khá bất tiện cho đoàn đi xe ô
tô vì đường một chiều và nằm trong.
(0.6303555965423584)
vị trí khá bất tiện cho người đi xe ô
tô vì đường một chiều và nằm trong hẻm
(0.942931592464447)
...
```

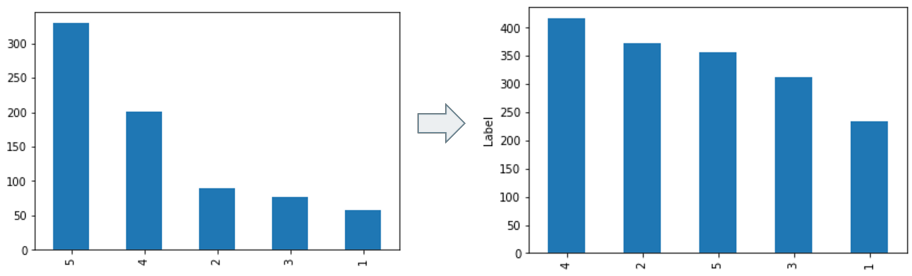
Output: Số câu tăng cường sẽ dựa theo tỉ lệ dữ liệu của class sentiment tương ứng:

- class 1, 2, 3: top 3 câu có score cao nhất
- class 4: top 1 câu
- class 5 không tăng cường dữ liệu

```
vị trí khá bất tiện cho người đi xe ô tô vì
đường một chiều và nằm trong hẻm
(0.942931592464447)
Đường khá bất tiện cho đoàn đi xe ô tô vì
đường một chiều và nằm trong hẻm
(0.5692278742790222)
vị trí khá bất tiện cho đoàn đi xe ô tô vì
đi một chiều và nằm trong hẻm
(0.5594411492347717)
```

After generating the list of sentences and their corresponding scores, they are sorted in descending order based on their score. Depending on the original class of the input sentence, a certain number of top scoring sentences are selected to be added to the dataset as augmented data. For example, if the original sentence is classified as a 1, 2, or 3, the three highest scoring sentences will be selected for augmentation. If the original sentence is classified as 4, only the highest scoring sentence will be selected, and no sentences will be augmented for sentences classified as 5.

4.5 Augmentation result



The data augmentation process resulted in a significant increase in the size of the dataset, from 755 to 1691 sentences. The number of sentences for each class was also substantially increased, with the minimum number of samples per class raised no less than 100 for all targeted classes and even over 250 for class 2. Furthermore, samples from the augmented sentences generated by the pipeline below are varied, while still maintaining the original semantic meaning of the input sentence.

Review	labels
Gần biển,gần chỗ ăn uống.Nhân viên dễ thương	4
Gần biển, gần chỗ ăn uống. dễ thương	4
Vị trí đẹp, ngay đối diện với cổng DH Quy Nhơn. Hơi nhỏ nhưng với vị trí như vậy thì cũng là quá tốt	4
Vị trí đẹp, ngay đối diện với THDH Quy Nhơn. Hơi nhỏ nhưng với vị trí như vậy thì cũng là quá tốt	4
Tuy không tiện nghi nhưng giá cả bình dân nhất Quy Nhơn	4
Tuy không tiện nghi nhưng lại bình dân nhất Quy Nhơn	4
Giường hơi mềm, không phù hợp lắm với người bị đau lưng	3
Giường hơi mềm, không phù hợp lắm với người bị đau.	3
Có thể do dịch nên ít khách. Phòng có mùi mốc	2
Có thể do dịch nên ít khách. Phòng có mùi mốc	2
Cách âm kém, ko có hỗ trợ KH về các vật dụng mượn thêm vào buổi tối (như tô muỗng)	2
Cách âm kém, không có hỗ trợ KH về các vật dụng mượn thêm vào buổi tối (như tô muỗng)	2
30/4 vừa rồi đông khách, vợ mình đợi 30" chưa đi được thang máy.	3
30/4 vừa rồi đông khách, " mình đợi 30 " chưa đi được thang máy.	3
Có chỗ để xe máy ở tầng trệt, ở trong hẻm nhưng đường đủ rộng để xe 4 bánh vào tới nơi.	2
Có chỗ để xe máy ở tầng trệt, ở trong hẻm nhưng đường đủ rộng để xe 4 bánh vào tới nơi.	2
Phòng khá cũ. Mình ở 2 lần 1 lần ở phòng bình thường thì thấy ok. 1 lần ở studio view biển thất vọng thực sự. View biển đã bị chắn bởi công trình (n	1
Phòng khá cũ. Mình ở 2 lần 1 lần ở phòng bình thường thì thấy ok. 1 lần ở studio view biển thất vọng thực sự. View biển đã bị chắn bởi công trình (r	1
Khách sạn gần trung tâm đi lại khá tiện. Phòng ốc xịn xò sạch sẽ. Mình book được phòng view nhìn chếch ra biển khá chill	4
Khách sạn gần trung tâm đi lại khá tiện. Phòng ốc xịn xò sạch sẽ dễ book được phòng view nhìn chếch ra biển khá chill	4
Khe vịn mở cửa tủ quần áo hơi cạn nên khi mở hơi khó, dễ gãy móng tay, nên khoét sâu hơn hoặc có nắm cửa bên ngoài.	2
Khe vịn mở cửa tủ quần áo hơi cạn nên khi mở hơi khó, dễ gãy ngón tay, nên khoét sâu hơn hoặc có nắm cửa bên ngoài.	2
Mình book phòng view biển nhưng view không đẹp, có nhiều tòa nhà, công trình đang thi công che khuất tầm nhìn. Không được đẹp như hình mô tả.	2
Mình book phòng view biển nhưng view không đẹp, có nhiều tòa nhà, công trình đang thi công che khuất tầm nhìn. Không được đẹp như hình mô tả.	2
Quán cà phê và khách sạn, trong chính khách sạn, chúng tôi không chỉ ở trên bãi biển. Họ làm sạch bãi biển và làm sạch. Có một số giường tắm nắn	3
Quán cà phê và khách sạn, trong chính khách sạn, chúng tôi không chỉ ở trên bãi biển. Họ làm sạch bãi biển và làm sạch. Có một số giường tắm nắn	3
Cách âm kém, đêm hôm phòng bên cạnh cười nghe rõ. Không có nhân viên hỗ trợ xách đồ. View biển nhưng trước mặt có công trình xây dựng	2
Cách âm kém, đêm hôm người bên cạnh cười nghe rõ. Không có nhân viên hỗ trợ xách đồ. View biển nhưng trước mặt có công trình xây dựng	2
Ko có cách âm, nên cải thiện	4
Ko có tâm, nên cải thiện	4
Chủ khách sạn không cho khách nhận phòng ngay mà dắt đi xem tùm lum và chê phòng khách đã đặt để khách đổi phòng khác giá cao hơn! Phàn n	2
Chủ khách sạn không cho khách nhận phòng ngay mà dắt đi xem tùm lum và chê phòng khách đã đặt để khách đổi phòng khác giá cao hơn! Phàn n	2
Hồ bơi ngày mồng 5/12 nhân viên báo đang bảo trì, do đó không dùng được trong ngày lưu trú ở khách sạn.	2
Hồ bơi ngày mồng 5/12 nhân viên báo đang bảo trì, do đó không dùng được trong thời_gian lưu trú ở khách sạn.	2
Máy lạnh cũ nên không điều chỉnh được nhiệt độ, hơi nóng	2
Xe cũ nên không điều chỉnh được nhiệt độ, hơi nóng	2

5 Experimentatal Result

In this chapter, we present the results of our experiments. For evaluating the performance of these models, we used two standard metrics for text classification tasks: accuracy and F1 score. The accuracy measures the percentage of correctly classified instances, while the F1 score takes into account both precision and recall, and is particularly useful when the classes are imbalanced. To report our results, we adopted a convention of denoting the accuracy and F1 score as a pair of values, separated by a dash.

5.1 Sentiment analysis result: Count vectorization, TF-IDF; Random Forest, SVM, and Naive Bayes

We can see that since the comments usually consist of 2-3 sentences, using TF-IDF in combination with stopword removal helps the machine learning models to understand and make better predictions by reducing the content in comments that are very lengthy. On the other hand, count vectorization is based on the frequency of occurrence of words, so models like Naive Bayes or Random Forest, which have a probabilistic (random) nature, will provide higher accuracy. In contrast, SVM performs better with TF-IDF.

Model	Random Forest	SVM	Naive Bayes
Count vectorize	0.52-0.44	0.50-0.36	0.57-0.45
TF-IDF	0.51- 0.42	0.56-0.44	0.47-0.20

5.2 Sentiment analysis result: POS tagging tokenization methods: maximum matching, underthesea, and VNCoreNLP libraries

Model	Random Forest	SVM	Naive Bayes
Maximum Matching	0.51-0.44	0.52-0.40	0.54-0.42
underthesea	0.52-0.44	0.50-0.36	0.57-0.45
VnCoreNLP	0.64-0.59	0.61-0.57	0.59-0.54

We can see that the best machine learning model is Naive Bayes for all three tokenization methods. Additionally, VNCoreNLP and underthesea have better results because they are trained on more diverse tokenization datasets compared to the group's dataset. Furthermore, these libraries use multiple tokenization methods, so their results will be better than those using only maximum matching, which is the tokenization method used by the group.
It is interesting to note that while Naive Bayes performed the best across all tokenization methods, the differences in performance between the libraries were not significant. This suggests that the choice of tokenization library may not have a large impact on the

overall performance of the sentiment analysis task. However, using a more diverse tokenization dataset and multiple tokenization methods may improve the accuracy of the model.

5.3 Sentiment analysis result: original and augmented dataset

We can see that the models' performance improved in terms of both accuracy and F1 score by 2-4% after data augmentation, and this also helped to significantly improve the F1 score of the models by addressing the problem of imbalanced data. Furthermore, data augmentation enabled the models to learn additional important features that influence the classification decision for the machine learning models.

Model	Random Forest	SVM	Naive Bayes
Non augment	0.640-0.590	0.61-0.57	0.59-0.54
Augmented data	0.649-0.616	0.63-0.59	0.61-0.55

References

[0] A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models - https://arxiv.org/abs/2010.15036

[1] A Review of Machine Learning Algorithms for Text Classification - https://link.springer.com/chapter/10.1007/978-981-16-9229-1$_1$4

[2] Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques - https://www.hindawi.com/journals/complexity/2021/255

[3] PhoBERT: Pre-trained language models for Vietnamese - https://arxiv.org/abs/2003.00744

[4] A Fast and Accurate Vietnamese Word Segmenter - https://arxiv.org/abs/1709.06307

[5] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding - https://arxiv.org/abs/1810.04805

[6] VnCoreNLP: A Vietnamese Natural Language Processing Toolkit - https://arxiv.org/abs/1801.01331

[7] Masked Language Model Scoring - https://arxiv.org/abs/1910.14659

[8] Pre-Trained Models: Past, Present and Future - https://arxiv.org/abs/2106.07139

[9] Fine-grained Sentiment Classification using BERT - https://arxiv.org/abs/1910.03474

[10] Sentiment Analysis Implementing BERT-based Pre-trained Language Model for Vietnamese - https://ieeexplore.ieee.org/document/93

[11] The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) - https://jalammar.github.io/illustrated-bert/

[12] A history of progress on text representation in NLP (Part 3 - Transformer models) - https://luungoc2005.github.io/blog/2020-06-11-brief-history-of-nlp-p3