

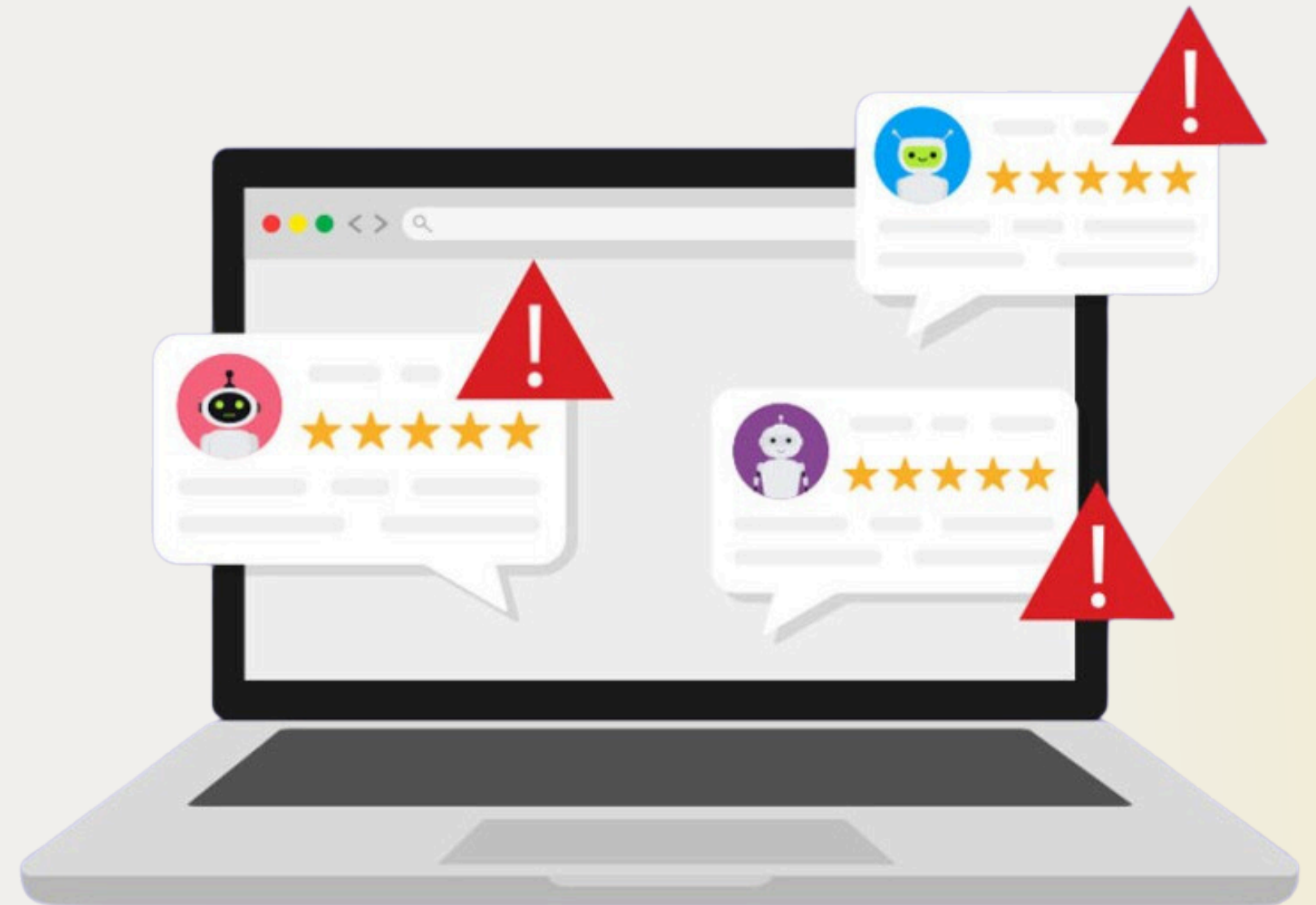
Spam review detection on e-commerce platforms in **Vietnam: Experiments** with **PhoBERT** and **Autoencoder**

Table of Content

Spam review detection on e-commerce platforms in Vietnam: Experiments with PhoBERT and Autoencoder

- Introduction
- Related work
- Methodology
- Evaluation
- Future work
- Conclusion

Introduction




In the digital era, e-commerce platforms have become an indispensable part of the retail sector, especially in rapidly developing countries like Vietnam.



Nowadays, product reviews play a key role in shaping customers' purchasing decisions

However, with such rapid development, the number of product reviews has increased significantly along with many product reviews that do not bring any value to customers but are only for the purpose of receiving rewards.

SPAM REVIEW DETECTION ON
E-COMMERCE PLATFORMS
IN VIETNAM



ctuphx_fa6

★★★★★


2023-07-16 12:09 | Phân loại hàng: What Hồng,L(53-58kg)

Màu sắc: đúng

Đúng với mô tả: đúng

Chất liệu: đẹp

đẹp nên mua. sẽ ủng hộ lần sau. học sinh giỏi lớp đi đầm sen nước nè các bạn hợp với em nha anh em sẽ cố gắng 3 trên bàn tiên của em là ngày gì mà đội để làm rõ nguyên



mttrinhc


★★★★★

2023-11-03 16:28 | Phân loại hàng: Lê CAM,M(48-52kg)

Đúng với mô tả: ok

Chất liệu: kb

Màu sắc: trắng



h3301a8enj

★★★★★

2023-06-12 09:14 | Phân loại hàng: Gấu trắng,M(4

Chất liệu: vải

Đúng với mô tả: bo ky ra ma

Màu sắc: hồng trắng

Một khi nổi nhớ em biến thành dải ngân hà.
Thì anh biết chạy đi đâu.
Dù có đến cuối chân trời kia cùng ngã cùng và bao nhiêu kỷ niệm đã hóa đá.
Đừng trêu đùa mãi anh như con rối trong nhà.
Vì anh biết tất tất tất cả.
Sao anh hệt như cỏ cây em thích vun trồng.
Đến khi đổi thay mặc đấ

Sp ok so với giá tiền . Giao hàng nhanh.
Bởi vì ngày thường họ đã chịu đủ áp bức từ năm tu sĩ Đại Thừa này và gia đình của họ, bây giờ họ lại thấy một cao thủ như vậy xuất hiện.



Our ultimate goal is to ensure the rights of consumers. When users enter the product link, they will receive recommendations on whether to buy or not. However, in this presentation we focus on the smaller problem of detecting spam reviews: Experiments with PhoBERT and Autoencoder

Related work

Related work

Paper	Author	Method	Acuraccy
Ensemble machine learning model for classification of spam product reviews	M. Fayaz, A. Khan, J. U. Rahman, A. Alharbi, M. I. Uddin, and B. Alouffi	KNN, RF, and MLP	88.13%
Learning Document Representation for Deceptive Opinion Spam Detection	L. Li, W. Ren, B. Qin, and T. Liu	CNN	79.5%
Spam review detection using self attention based CNN and bi-directional LSTM	P. Bhuvaneshwari, A. N. Rao, and Y. H. Robinson	CNN and BiLSTM	87.3%
Towards accurate deceptive opinion spam detection based on word order-preserving CNN	S. Zhao, Z. Xu, L. Liu, and M. Guo, J. Yun	CNN	70.02%

Related work

PhoBert

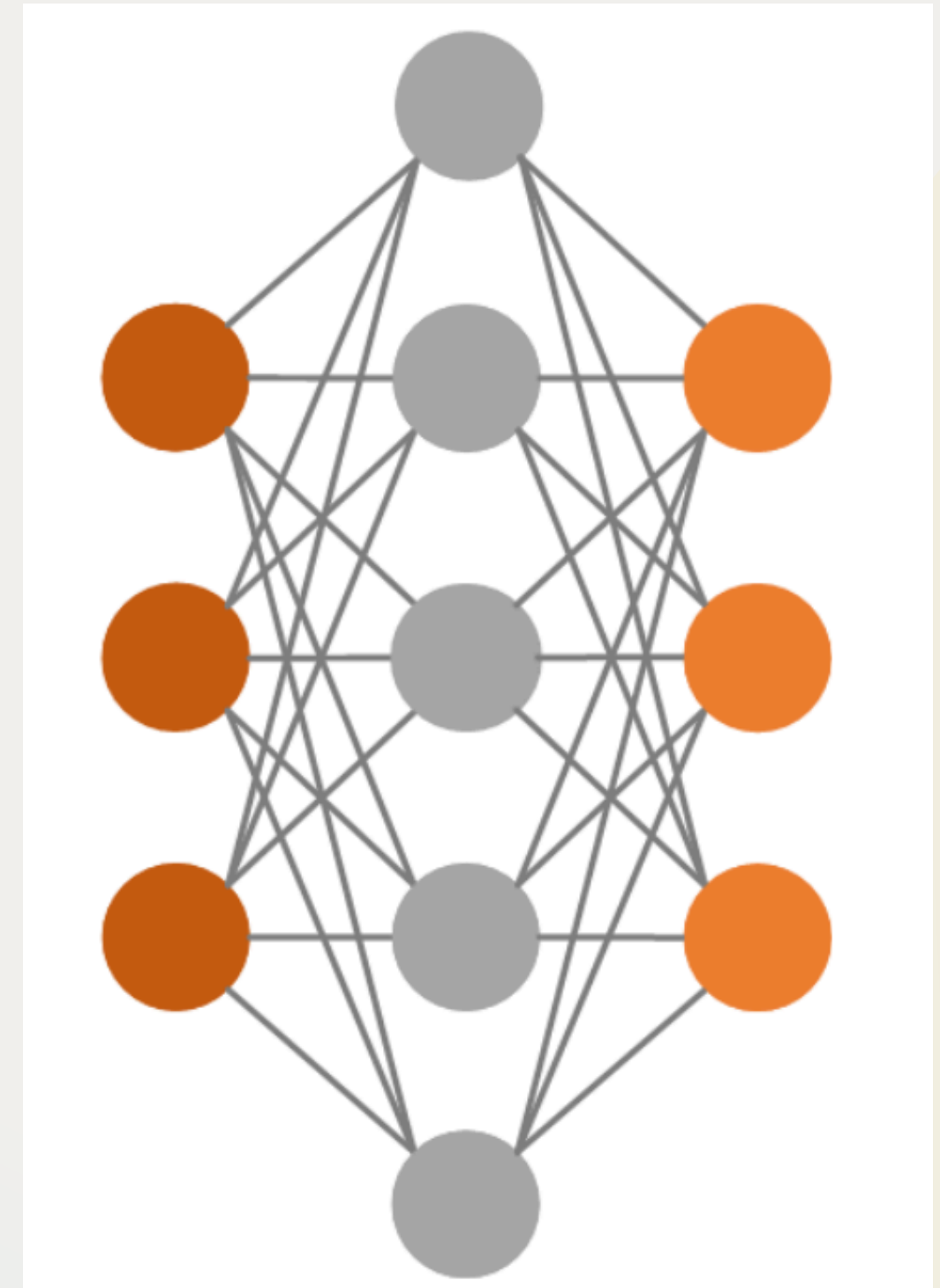
PhoBERT is a Bidirectional Encoder Representations from Transformers that offers fault tolerance, parallel processing, and self-learning capabilities. It is used for natural language processing tasks and has demonstrated comparable or superior performance, making it a robust candidate for spam review detection and maintaining the integrity of user-generated content in the digital landscape.



Related work

AutoEncoder

Autoencoders are effective tools for detecting spam reviews due to their ability to detect patterns and anomalies within data without requiring labeled training data. They enhance unsupervised learning techniques and complement supervised approaches, as demonstrated by their integration with PhoBERT in spam review detection research.



Methodology

Data Collection

https://shopee.vn/api/v2/item/get_ratings

Training set: 6000 reviews

Test set: about 1500 reviews

```
In [2]: 1 collect_reviews_product("reviews_45s.csv", max_products=500, min_len_cmt=4, types=[4, 5])
```

Đã lấy về 431 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 601.89 mili giây

Đã lấy về 441 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 533.01 mili giây

Đã lấy về 451 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 454.47 mili giây

Đã lấy về 461 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 602.05 mili giây

Đã lấy về 471 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 486.09 mili giây

Đã lấy về 481 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 479.59 mili giây

Đã lấy về 491 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 554.66 mili giây

Đã lấy về 501 sản phẩm trên tổng số tối đa 500 sản phẩm. Mất 633.49 mili giây

Đã thu thập và ghi 1 đánh giá của sản phẩm 19995035473 tại shop 678798230. Còn 500 sản phẩm nữa. Mất 602.23 mili giây

Đã thu thập và ghi 3260 đánh giá của sản phẩm 3261115659 tại shop 769085. Còn 499 sản phẩm nữa. Mất 214958.12 mili giây

Đã thu thập và ghi 15 đánh giá của sản phẩm 20883341400 tại shop 533312313. Còn 498 sản phẩm nữa. Mất 1249.10 mili giây

Đã thu thập và ghi 7 đánh giá của sản phẩm 18583831164 tại shop 178569040. Còn 497 sản phẩm nữa. Mất 1092.03 mili giây

Đã thu thập và ghi 58 đánh giá của sản phẩm 22967626489 tại shop 985237442. Còn 496 sản phẩm nữa. Mất 3639.67 mili giây

Đã thu thập và ghi 445 đánh giá của sản phẩm 21617915490 tại shop 788908334. Còn 495 sản phẩm nữa. Mất 24120.85 mili giây

Preprocessing

● Stopword Removal

Removing stopwords eliminates non-informative words from the text, such as "ấy", "ờ" aiding in focusing on more meaningful keywords during processing and model training.

● Emoji Removal

Removing emojis using regex simplifies the text, eliminating non-linguistic symbols, facilitating text processing and analysis

● Lowercasing

The text is converted to lowercase to standardize the representation of words and eliminate the distinction between uppercase and lowercase letters.

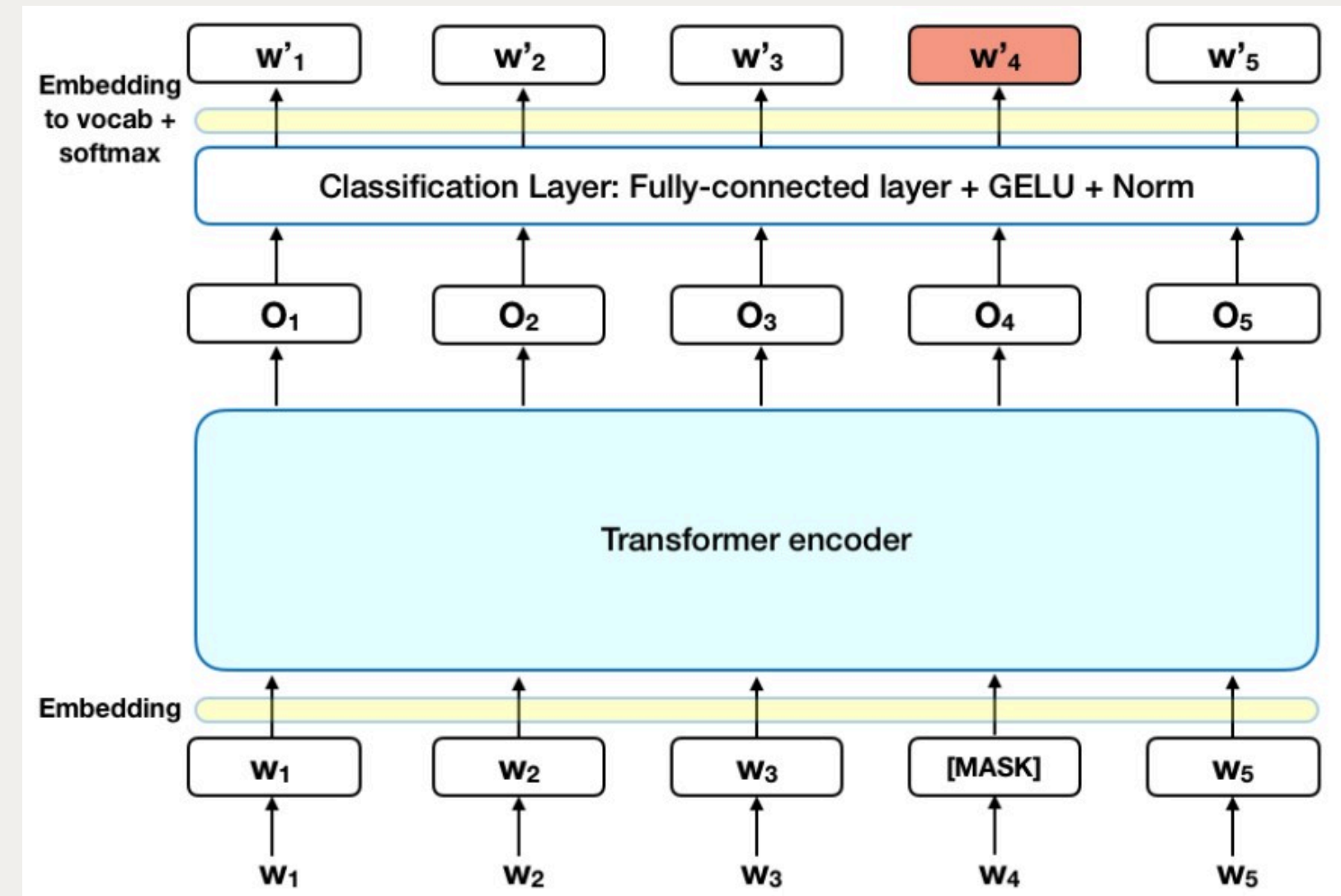
● Tokenization

Representing text as a sequence of "tokens" for easier processing and effectiveness in natural language processing tasks. (Using **VnCoreNLP**)

Word Embedding using PhoBERT

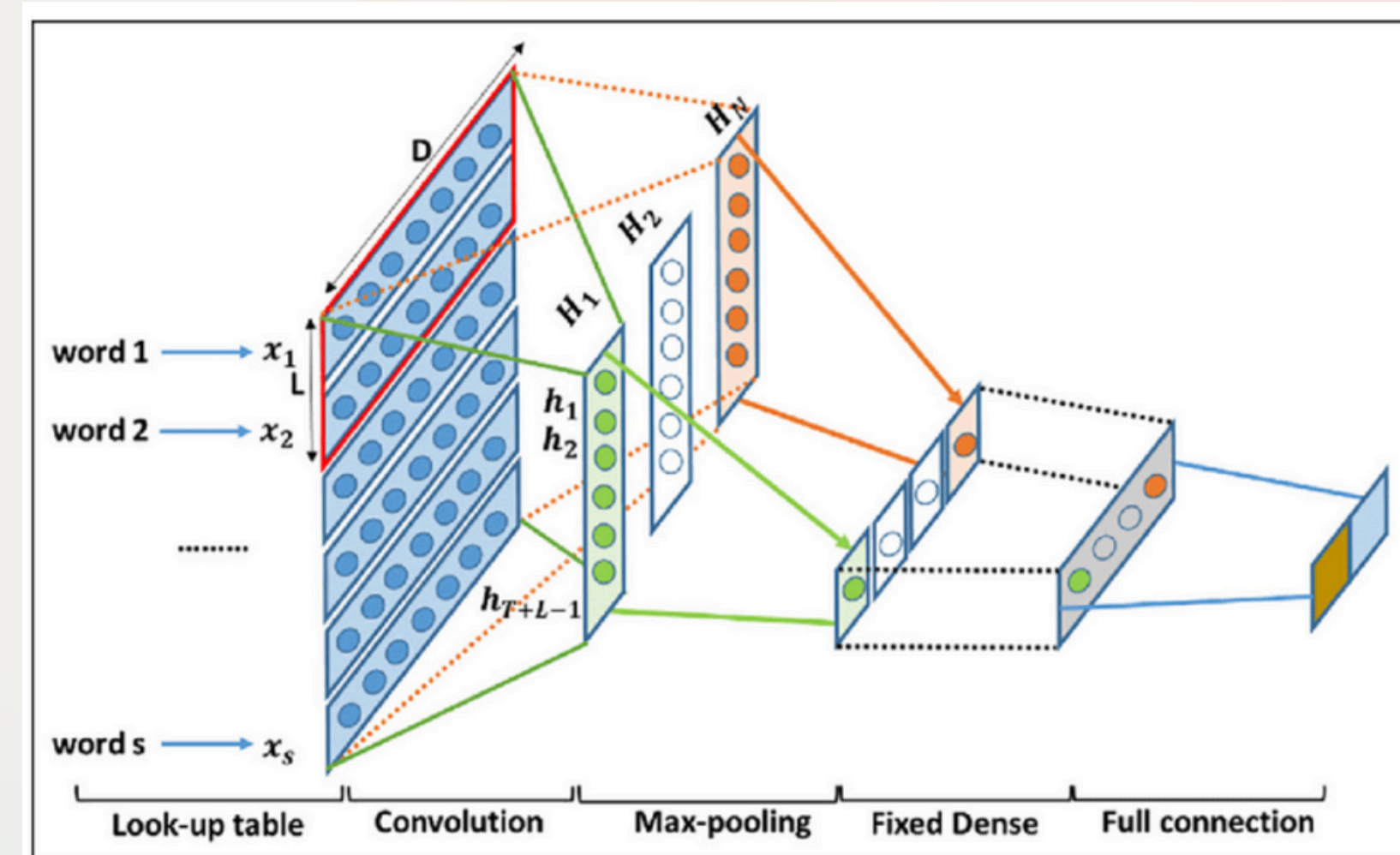
How VinAI trains PhoBERT to handle previous concerns:

- Used a large-scale corpus of 20GB Vietnamese texts
- Performed Vietnamese word segmentation before pre-training
- Pre-training corpus of 145M word-segmented sentences (3B word tokens)
- PhoBERT pre-training procedure is based on RoBERTa (Liu et. al., 2019) which optimizes BERT for more robust performance
- Two versions: PhoBERT-base (150M parameters) & PhoBERT-large (350M parameters)



Basic Model : CNN

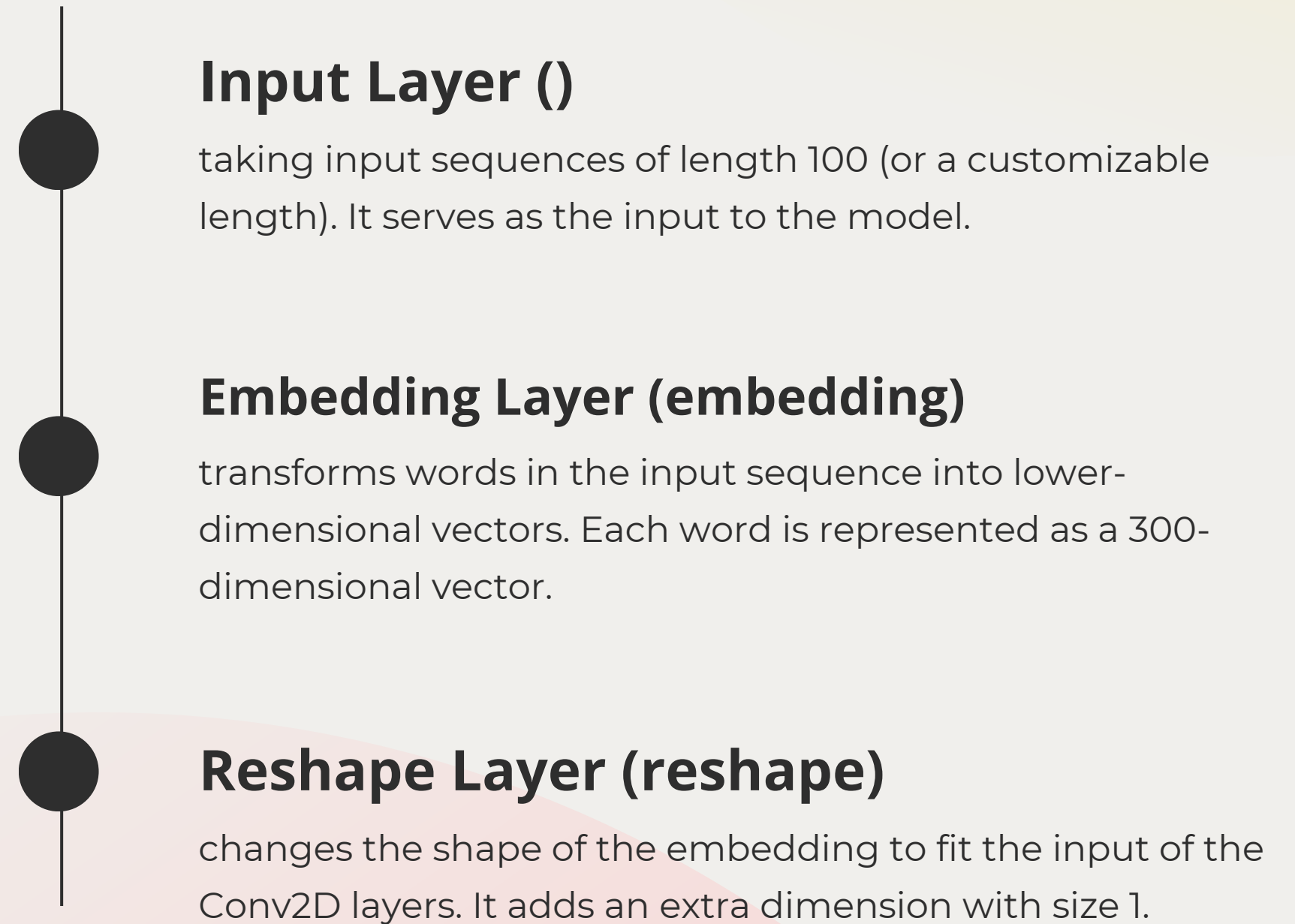
We apply the traditional convolutional neural network model to represent sentences. To make a composition for all of sentences, we use average operation to capture all of the sentence features on the pooling layer. This is a basic model, which is modified below to suit the deceptive review spam detection task.



Basic Model : Architecture of CNN

The CNN model has a total of 11 layers. Below is the main structure of Model CNN:

Total Trainable Parameters:
5,127,890.



Basic Model : Architecture of CNN

Conv2D Layers

Perform convolution on the input data to find spatial features. Each layer has a kernel of a different size.

MaxPooling2D Layers

Retain the maximum value from the result of the convolution, reducing the size of the output tensor. Max pooling after each convolution helps capture important information.

Concatenate Layer (concatenate)

Concatenates the outputs of the MaxPooling2D layers into a single tensor along the third axis (axis=1).

Flatten Layer (flatten)

Flattens the tensor from the concatenate layer into a vector to prepare for fully connected layers.

Dropout Layer (dropout)

Prevent overfitting by randomly turning off some neurons during training.

Dense Layer (dense)

The final fully connected layer with softmax activation, suitable for binary classification. It outputs predicted probabilities for each class.

PhoBERT

Transformers is an architecture that has been proposed in recent years and is currently in widespread use. The appearance of BERT helps many downstream tasks in NLP attain high-performance results while training on a small dataset. BERT and its variances become the baseline approaches in many NLP tasks, which is called BERTology

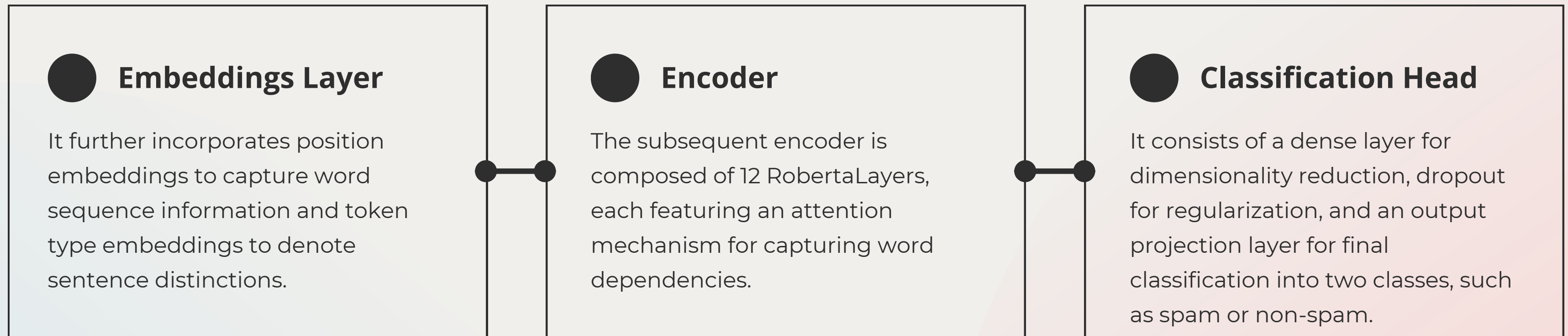
BERT (Bidirectional Encoder Representations from Transformers)

Model is structured around a transformer-based architecture, a neural network design that has proven highly effective in natural language processing tasks

Consists of an encoder stack comprising multiple layers, typically utilizing the Transformer's attention mechanism. The key innovation lies in bidirectional attention, allowing BERT to consider both preceding and subsequent context for each word in a given sequence simultaneously

PhoBERT Architecture

The model's architecture enables it to capture complex contextual relationships within the input data, contributing to its success in various NLP applications.



AutoEncoder

An autoencoder is an unsupervised neural network that learns to map the input to itself. Therefore, the dimensionality of inputs is same as the dimensionality of outputs. Considering the labeled data problem in review spam detection domain, autoencoder can be used as an anomaly detection system. The hidden layer representations of real reviews and spam reviews are significantly different which is used to separate them into subspace

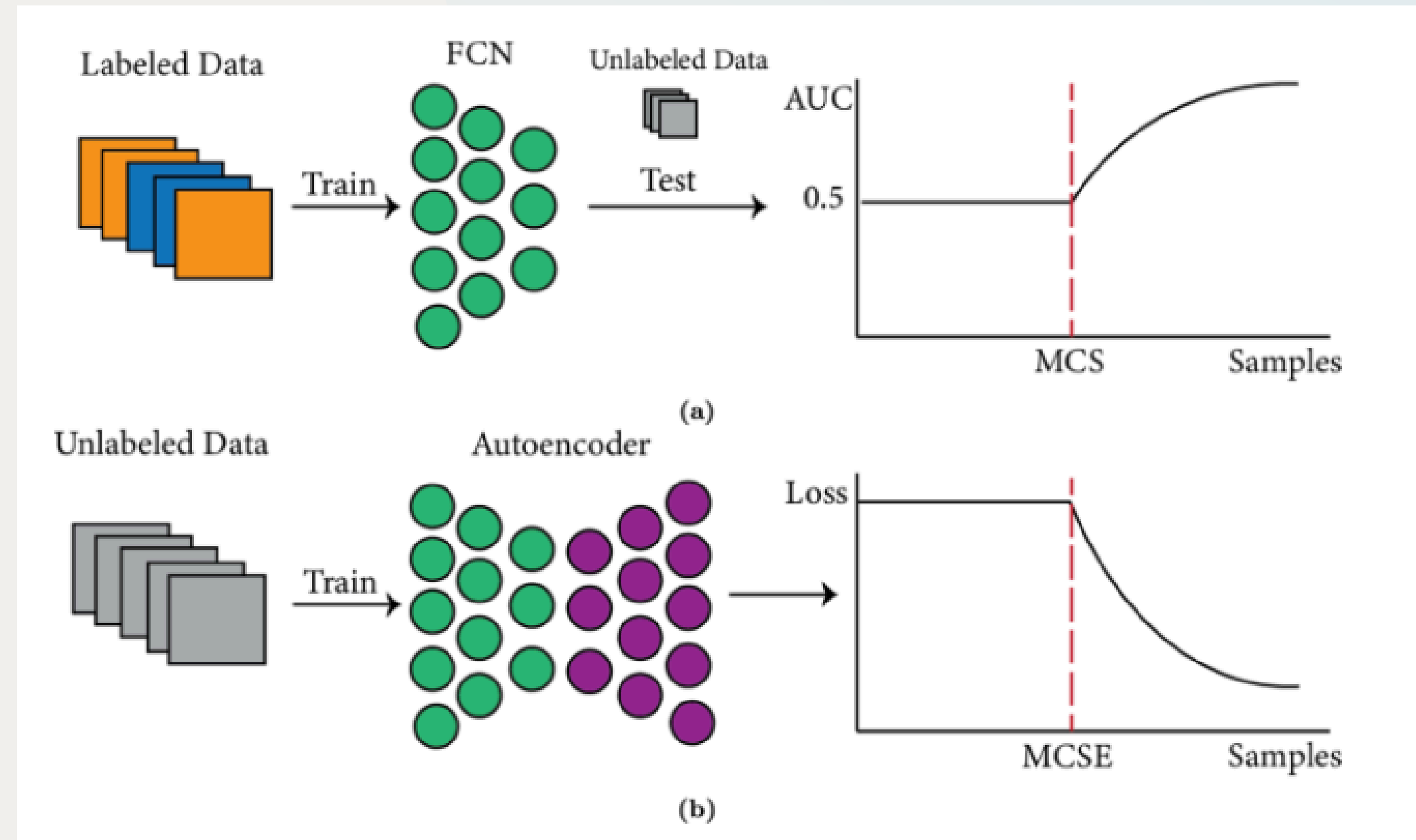


FIGURE. 01: PROPOSED SPAM REVIEW DETECTION MODEL

Architecture of AE

The model is made up of an encoder and decoder, which transform input data into a compressed latent space and then reconstruct it. The Leaky ReLU activation function fosters non-linearity in the network, while the sigmoid activation ensures the reconstruction adheres to the input's original range. The architecture is versatile and can be applied to various tasks, with effectiveness depending on encoding size and input data.

```
class Autoencoder(nn.Module):
    def __init__(self, input_size, encoding_size):
        super(Autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(input_size, encoding_size * 8),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 8, encoding_size * 4),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 4, encoding_size * 2),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 2, encoding_size),
            nn.LeakyReLU()
        )
        self.decoder = nn.Sequential(
            nn.Linear(encoding_size, encoding_size * 2),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 2, encoding_size * 4),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 4, encoding_size * 8),
            nn.LeakyReLU(),
            nn.Linear(encoding_size * 8, input_size),
            nn.Sigmoid()
        )

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Encoder

The encoder consists of a sequence of fully connected (linear) layers, each followed by a Leaky ReLU activation function.

```
self.encoder = nn.Sequential(  
    nn.Linear(input_size, encoding_size * 8),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 8, encoding_size * 4),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 4, encoding_size * 2),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 2, encoding_size),  
    nn.LeakyReLU()  
)
```

The first linear layer

Takes an input of size `input_size` and outputs a representation of size `encoding_size * 8`.

Subsequent layers

Gradually reduce the dimensionality, mapping the input to lower-dimensional representations.

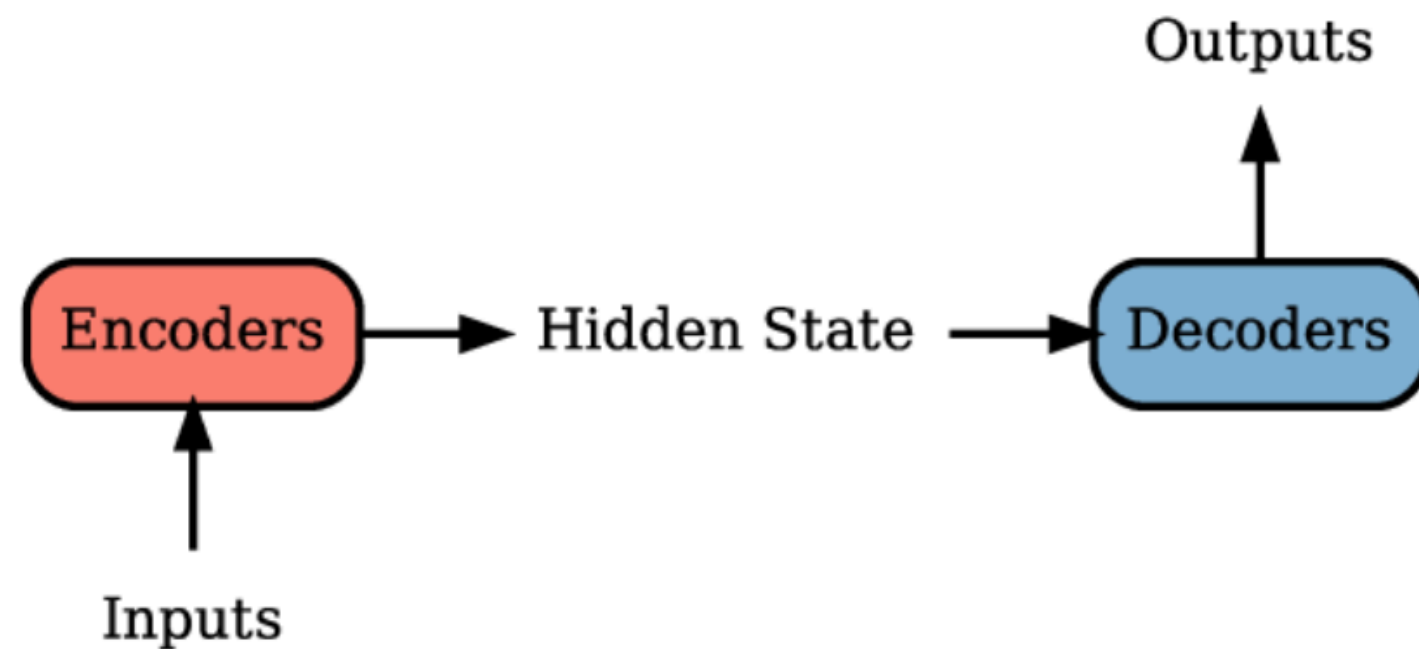
The final linear layer

Compresses the input into a representation of size `encoding_size`.

Decoder

The decoder mirrors the structure of the encoder but in the reverse order.

```
self.decoder = nn.Sequential(  
    nn.Linear(encoding_size, encoding_size * 2),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 2, encoding_size * 4),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 4, encoding_size * 8),  
    nn.LeakyReLU(),  
    nn.Linear(encoding_size * 8, input_size),  
    nn.Sigmoid()  
)
```



Structure of the decoder

It takes the compressed representation from the encoder and reconstructs the original input size.

Leaky ReLU activation function

Each linear layer in the decoder is followed by a Leaky ReLU activation function, except for the last layer, which is followed by a Sigmoid activation.

The Sigmoid activation

The Sigmoid activation in the final layer ensures that the reconstructed output is in the range $[0, 1]$, suitable for tasks like image reconstruction or input normalization.

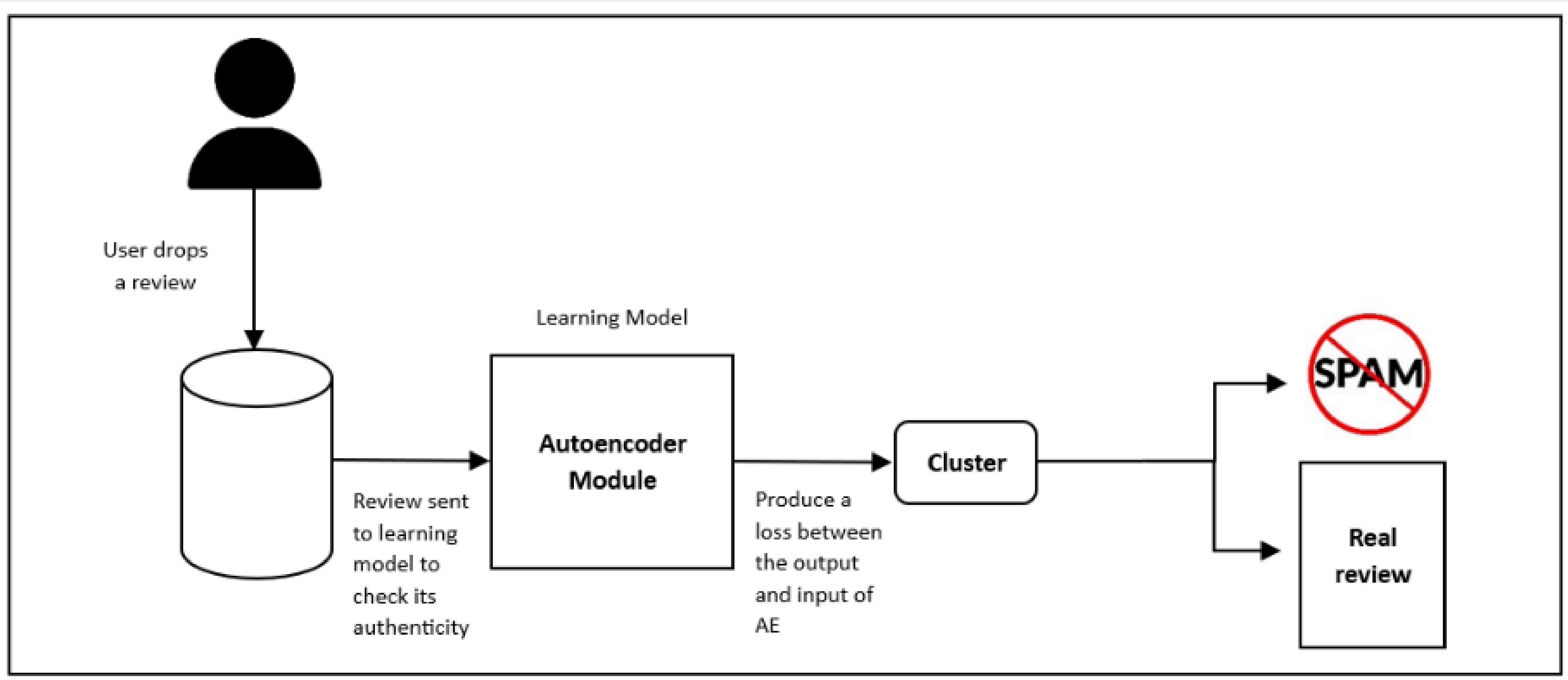
Activation Functions

Leaky ReLU is used as the activation function in the intermediate layers, allowing a small, non-zero gradient when the input is negative.

Sigmoid activation in the output layer ensures the reconstruction values are within the appropriate range.

The primary purpose of this autoencoder is to perform dimensionality reduction, capturing essential features of the input data in a lower-dimensional space. This compressed representation can be valuable for various tasks, such as denoising, anomaly detection, or feature learning

Overview of the proposed spam review detection model



Evaluation

Text CNN

Text CNN has accuracy is 83.55%. It also has F1 Score of 77.24%, indicating a balance between precision and recall across classes. Text CNN has a quite high true positive rate as shown in the confusion matrix, indicating areas of strength and potential improvement.

Metric	Text CNN
Accuracy	0.8355
F1 Score	0.7724
Confusion Matrix	<div>[[2704 160 [493 613]]</div>

PhoBERT Model

PhoBERT outperforms Text CNN in accuracy and F1 score, achieving 90.01% and 86.89% respectively. Both models exhibit high true positive counts, but PhoBERT's confusion matrix suggests a higher precision in positive case.

PhoBERT outperforms Text CNN indicating its effectiveness in making predictions for the classification task.

Metric	PhoBERT
Accuracy	0.9001
F1 Score	0.8689
Confusion Matrix	$\begin{bmatrix} 2758 & 109 \\ 288 & 819 \end{bmatrix}$

AutoEncoder

Metric	Test set 2% label 1	Test set 1% label 1	Test set 0.5% label 1
Accuracy	0.7689	0.7717	0.7723
Recall	0.5517	0.6071	0.5714
Confusion Matrix	<pre>[[2217 650 [26 32]]</pre>	<pre>[[2217 650 [11 17]]</pre>	<pre>[[2217 650 [6 8]]</pre>

The model's accuracy remained above 75%. Recall varied across test sets, with the highest recall achieved in the 1% spam test set. However, the confusion matrix showed a substantial number of false positives, indicating that many non-spam review were incorrectly classified as spam.

Splitting the test set into several percent labels did not significantly change accuracy and F1 score. The model is stable and able to handle variations in the test data and it will be very suitable for data with a small percentage of labels, but not very good.

Future Work

We have experimented with detecting review spam with TextCNN, PhoBERT and Autoencoder models, in which PhoBERT still gives superior results.

However, the results from the Autoencoder unsupervised approach are remarkable, achieving 76% performance. When review spam tactics change rapidly and become complex, there are reviews in the first half that are very good but the rest are spam, making the model difficult to grasp. We think that in the future work, maybe using the attention mechanism to focus more on the second half of the review will yield better results than the unsupervised approach.

Conclusion

In the current context of e-commerce development, developing a model to detect spam reviews is important.

The task of detecting Vietnamese spam reviews is a challenging task and there has not been much research on this issue. Our research also provides a data set of about 200,000 reviews collected from the e-commerce platform shopee that have been labeled. In addition, we have proposed a very promising unsupervised approach Autoencoder for further research.

We hope that this study has provided an empirical perspective on review spam detection and will stimulate research on this issue.