MaSSP

MaSSP
DATA SCIENCE

Vietnamese Fake News Detection

Based on Tokenization from Pre-trained Models and Word Embeddings BiLSTM Model

Hai Le, Hieu Dinh, Hiep Nguyen, Dung Dao

August 5, 2023

Contents

1. Introduction
2. Methodology
3. Results
4. Discussion
5. Future works
6. Demo

Problem Statement

By mitigating the impact of fake news, this research aims to safeguard the integrity of information dissemination and promote a more informed and discerning online community in Vietnam.

Research Questions

- Can a deep learning model be trained to effectively distinguish between factual and fabricated news stories in Vietnamese with high accuracy?
- What neural network architectures and NLP techniques work best for the Vietnamese language and this task?
- How does the model's performance compare to human evaluation and baselines?

Research Goals

The objectives aim to develop an accurate, interpretable, and robust Vietnamese fake news detector using deep learning that can generalize well and be deployed in practical applications. Both model performance and practical utility are key goals of this research.

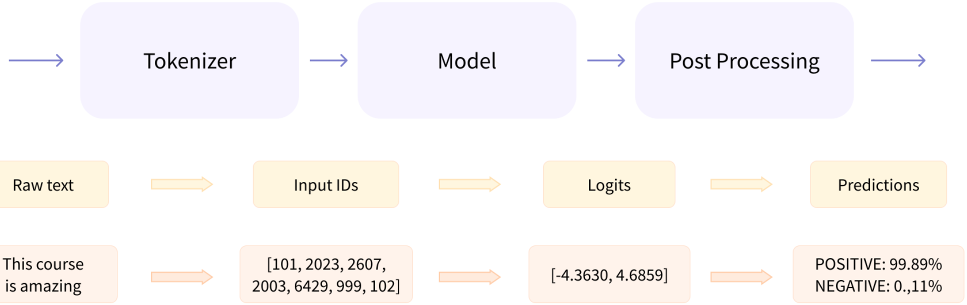
Datasets

ReINTEL 2020 dataset:

- comprises a total of 9713 items, including both news articles and social media posts collected from various sources, primarily from social media platforms (SNSs) and Vietnamese newspapers.
- covers a wide range of domains, such as entertainment, sports, finance, healthcare, and the Covid-19 pandemic.

Preprocessing - Tokenization

General understanding



Tokenizer → Model → Post Processing

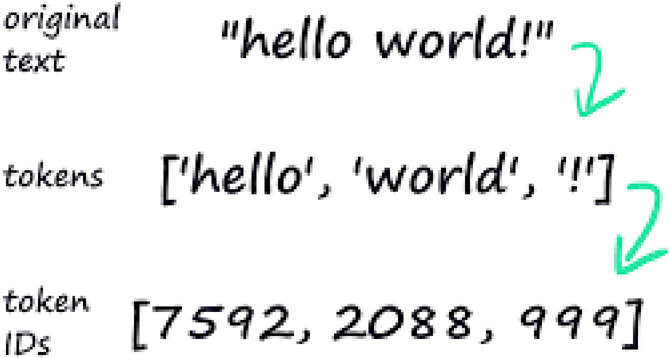
Raw text → Input IDs → Logits → Predictions

This course is amazing → [101, 2023, 2607, 2003, 6429, 999, 102] → [-4.3630, 4.6859] → POSITIVE: 99.89%
NEGATIVE: 0.,11%

Source: Hugging Face - NLP Course

Preprocessing - Tokenization

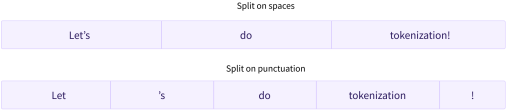
General understanding



Source: Towards Data Science

Preprocessing - Tokenization

Raw text → Tokens



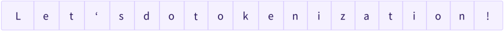
Split on spaces

Let’s	do	tokenization!

Split on punctuation

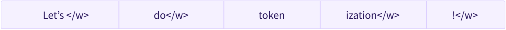
Let	’s	do	tokenization	!

Figure: Word-based



L	e	t	‘	s	d	o	t	o	k	e	n	i	z	a	t	i	o	n	!

Figure: Character-based



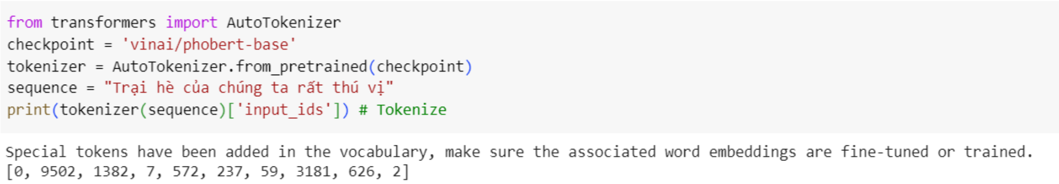
Let’s </w>	do</w>	token	ization</w>	!</w>

Figure: Subword

Source: Hugging Face - NLP course

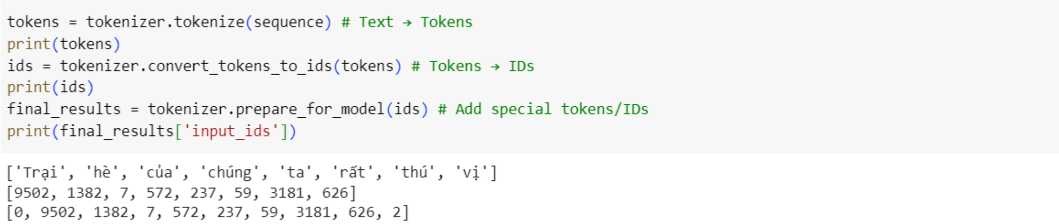
Preprocessing - Tokenization

Tokens → Perspective IDs



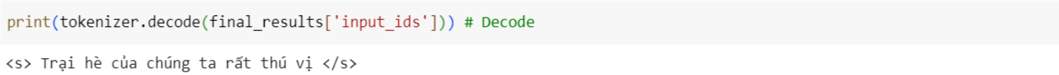
```
from transformers import AutoTokenizer
checkpoint = 'vinai/phobert-base'
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
sequence = "Trại hè của chúng ta rất thú vị"
print(tokenizer(sequence)['input_ids']) # Tokenize
```

```
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
[0, 9502, 1382, 7, 572, 237, 59, 3181, 626, 2]
```



```
tokens = tokenizer.tokenize(sequence) # Text → Tokens
print(tokens)
ids = tokenizer.convert_tokens_to_ids(tokens) # Tokens → IDs
print(ids)
final_results = tokenizer.prepare_for_model(ids) # Add special tokens/IDs
print(final_results['input_ids'])
```

```
['Trại', 'hè', 'của', 'chúng', 'ta', 'rất', 'thú', 'vị']
[9502, 1382, 7, 572, 237, 59, 3181, 626]
[0, 9502, 1382, 7, 572, 237, 59, 3181, 626, 2]
```



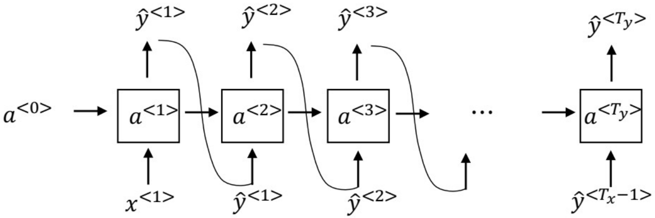
```
print(tokenizer.decode(final_results['input_ids'])) # Decode
```

```
<s> Trại hè của chúng ta rất thú vị </s>
```

Model Selection

Overview of RNN

Recurrent Neural Network: a type of artificial neural network which is used for sequential data or time series data.



$$a^{\langle t\rangle} = \tanh\left(W_{aa}a^{\langle t-1\rangle} + W_{ax}x^{\langle t\rangle} + b_a\right)$$

$$y^{\langle t\rangle} = \sigma\left(W_{ya}a^{\langle t\rangle} + b_y\right)$$

Figure: Recurrent Neural Network model

Model Selection

Vanishing gradient descent

The vanishing gradient problem is caused by the derivative of the activation function used to create the neural network

Backpropagation:

$$\mathcal{L}^{\langle t \rangle}\left(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}\right) = -y^{\langle t \rangle} \log \hat{y}^{\langle t \rangle} - \left(1 - y^{\langle t \rangle}\right) \log\left(1 - \hat{y}^{\langle t \rangle}\right)$$

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{\langle t \rangle}\left(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}\right)$$

$$\frac{d\mathcal{L}(\hat{y}, y)}{dW_{aa}} = \sum_{t=1}^{T_y} \left(\frac{d\mathcal{L}}{d\hat{y}^{(t)}} \frac{d\hat{y}^{\langle t \rangle}}{da^{(t)}} \frac{da^{\langle t \rangle}}{dW_{aa}} \right) \tag{1}$$

Model Selection
Overview of LSTM

Long Short-Term Memory: address problem related to vanishing gradient descent by integrating gating functions into their state dynamics

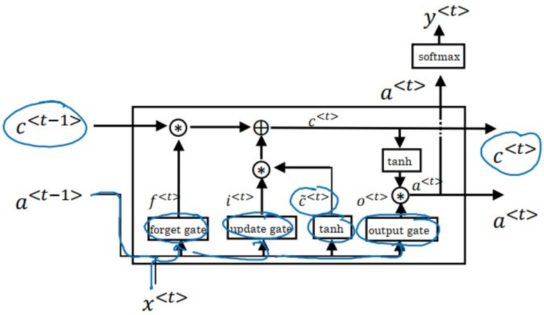


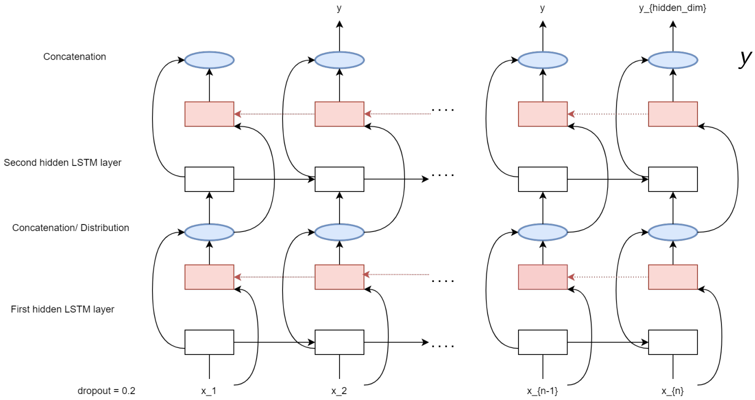
Figure: Long short-term memory model

$$\tilde{c}^{<t>} = \tanh\left(W_c\left[a^{<t-1>}, x^{<t>}\right] + b_c\right)$$
$$\Gamma_u = \sigma\left(W_u\left[a^{<t-1>}, x^{<t>}\right] + b_u\right)$$
$$\Gamma_f = \sigma\left(W_f\left[a^{<t-1>}, x^{<t>}\right] + b_f\right)$$
$$\Gamma_o = \sigma\left(W_o\left[a^{<t-1>}, x^{<t>}\right] + b_o\right)$$
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$
$$a^{<t>} = \Gamma_o * c^{<t>}$$

Model Selection

2 layers Bidirectional LSTM

In the bi-directional LSTM, both forward path and backward path are computed independently and their outputs are concatenated.



2 layers, dropout ratio $= 0.2$

$$y^{\langle t\rangle} = \sigma\left(W_{ya}[a_f^{\langle \vec{t}\rangle}, a_b^{\langle \vec{t}\rangle}] + b_y\right)$$

Figure: Diagram of a two-layer bi-directional LSTM network

Model Selection

The proposed model

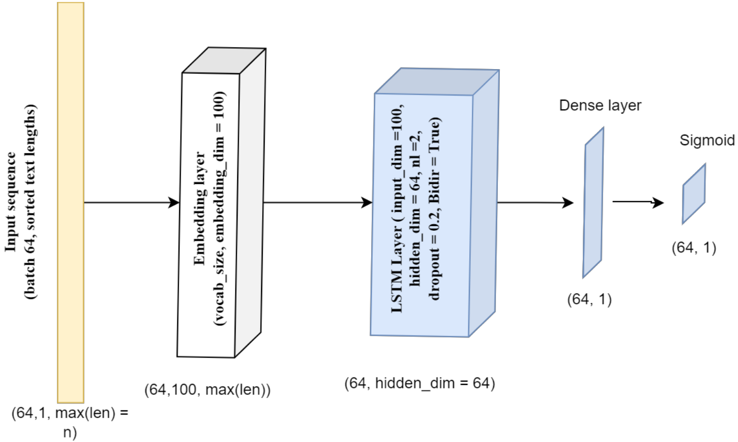

Input sequence
(batch 64, sorted text lengths)
(64,1, max(len) = n)
Embedding layer
(vocab_size, embedding_dim = 100)
(64,100, max(len))
LSTM Layer (input_dim =100, hidden_dim = 64, nl =2, dropout = 0.2, Bidir = True)
(64, hidden_dim = 64)
Dense layer
(64, 1)
Sigmoid
(64, 1)

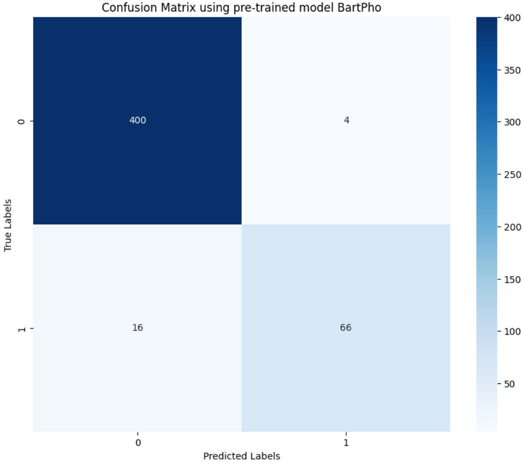
Figure: Overview of the proposed model

Results

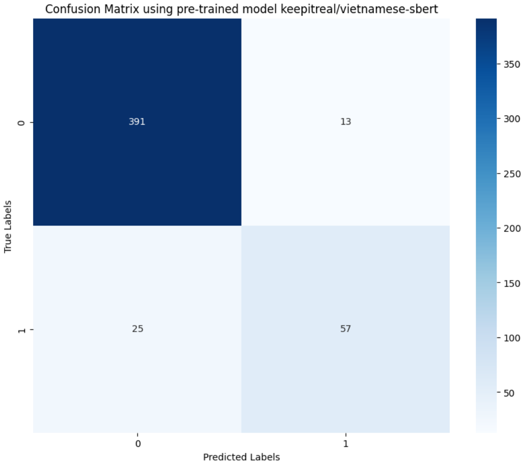
- The bidirectional LSTM model is able to achieve 95% accuracy in classifying Vietnamese news articles as real or fake. The precision and recall are 0.90 and 0.80 respectively. This shows the model is balanced in predicting true positives vs false positives.
- Using pre-trained BERT embeddings provides useful semantic representations of words. The attention mechanism improves performance slightly by focusing on salient words.
- However, training accuracy reaches 100% while validation accuracy remains in the 92-95% range, indicating some overfitting to the training data.

Confusion Matrix

BartPho

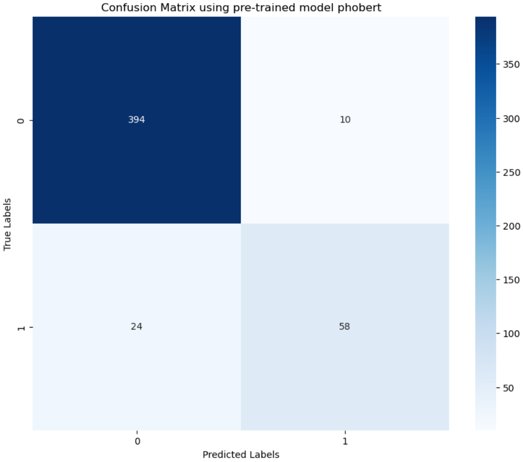


Confusion Matrix
keepitreal/vietnamese_sbert



Confusion Matrix

phobert



Discussion

The LSTM model achieved an impressive accuracy of 95.83% on the test set for classifying Vietnamese news as real or fake. Surpassing 90% accuracy demonstrates that deep learning approaches like LSTMs are highly effective for this task when provided with sufficient training data.

Factors contributed to the high accuracy

- The bidirectional LSTM architecture
- Words Tokenization from pre-train model
- The large labeled dataset

Remaining caveats before operationalization

- Human oversight is still required before taking action on flagged content, to prevent incorrect classifications from inadvertently suppressing real information or voices.
- Appeal mechanisms should be established for users whose legitimate content gets removed, to resolve unfair takedowns.
- Continued model tuning on new data is needed to maintain accuracy as the nature of fake news evolves over time.

The results demonstrate that deep learning provides a promising technology for fighting Vietnamese misinformation.

Future works

- Collect a larger Vietnamese news dataset.
- Fine-tune the pretrained BERT embeddings on the Vietnamese news data before LSTM training.
- Evaluate other sequential modeling architectures.
- Incorporate semantic analysis of the texts through techniques like named entity recognition and coreference resolution.
- Test the model on streaming social media posts.
- Build a browser extension that leverages the trained model.
- Collaborate with experts in journalism, ethics, and linguistics to develop a responsible and unbiased fake news system.

Demo

DEMO TIME!