January, 2023

Analyze Vietnamese sentiment by different word embeddings then use SVM model

Nguyen Dinh Nguyen Bac, Nguyen Tan Dat, Truong Thanh Huong
Industrial University of Ho Chi Minh City

Abstract: Vietnamese sentiment analysis on e-commerce platforms, social networks... poses a significant challenge. Especially in the complex context of classifying English texts and the difficulty increases when exposed to Vietnamese, a language rich in nuances and diverse opinions. This article explores the effectiveness of various word embedding methods, including Keras, Fastext (Facebook AI Research), and PhoBert, with the goal of identifying the most suitable approach for Vietnamese sentiment analysis. The central question revolves around finding a word embedding method that can robustly address machine learning challenges in the Vietnamese language. Leveraging customer review data categorized into positive and negative sentiments within a Vietnamese dataset, the study employs the SVM machine learning model for training. Evaluation metrics such as Accuracy, Confusion Matrix, and F1 score are applied to assess the performance of the different methods. The ultimate aim is to contribute to the advancement of Natural Language Processing (NLP) in Vietnam and inspire innovative ideas for future research in this domain.

Keyswords: SVM ,Sentiment classification, Text classification

1 Introduction

In recent years, sentiment analysis, a vital component of natural language processing, has seen a surge in significance. Its broad applications include grasping public opinion, interpreting market trends, and understanding social media dynamics. This study specifically targets sentiment analysis for the Vietnamese language, facing challenges arising from its intricate structure.

By leveraging machine learning techniques, particularly Support Vector Machines (SVM), our goal is to assess the suitability of these models for Vietnamese and propose innovative solutions. Enhancements in accuracy and efficiency in sentiment analysis are crucial for decoding sentiments in textual data, aiding in gauging public perception, understanding consumer preferences, and tracking societal trends.

Beyond the technical aspect, the development of robust sentiment analysis models tailored to the Vietnamese populace holds the potential to revolutionize communication strategies and decision-making processes in both business and social sectors. As these models evolve, they contribute not only to a deeper understanding of language dynamics but also to the enhancement of practical applications that shape the interactions between individuals and technology.

2 Related Work

Sentiment analysis, also known as opinion mining, has garnered significant attention in recent years due to the explosion of user-generated content on the internet. Various approaches have been explored to extract sentiment information from textual data, with a growing emphasis on leveraging word embeddings for more accurate and context-aware sentiment classification.

The advent of word embedding techniques, such as Word2Vec, GloVe, and FastText, marked a significant shift in sentiment analysis research. These methods capture semantic relationships between words and their contextual information, enabling more effective representation of textual data. [5] Demonstrating its ability to capture syntactic and semantic similarities in large datasets.

Adapting a pre-trained language model to a specific task using task-specific data, optimizing hyperparameters, and evaluating performance. It leverages pre-training knowledge for task-specific improvement[6].

[7]Showed that pre-training on a massive amount of data followed by task-specific fine-tuning outperforms traditional methods on various natural language understanding tasks, including sentiment analysis.

For our research, we employed word embedding techniques from three pre-trained models: Keras, FastText, and PhoBert. Subsequently, we integrated them with the Support Vector Machine (SVM) machine learning algorithm. This allowed us to compare and provide an overall perspective.

3 Proposed Model

3.1 Super Vector Machine Model

Support Vector Machine (SVM): SVM is another state of art algorithm which is mostly used for categorization. SVM is based on the concept of calculating margins. It is used to separate groups of data by drawing a line in between. The margins are selected such that there is a minimum difference between margin and labeled classes resulting in reducing classification *error*[1]. Figure 1 shows the architecture of the support vector machine classifier.

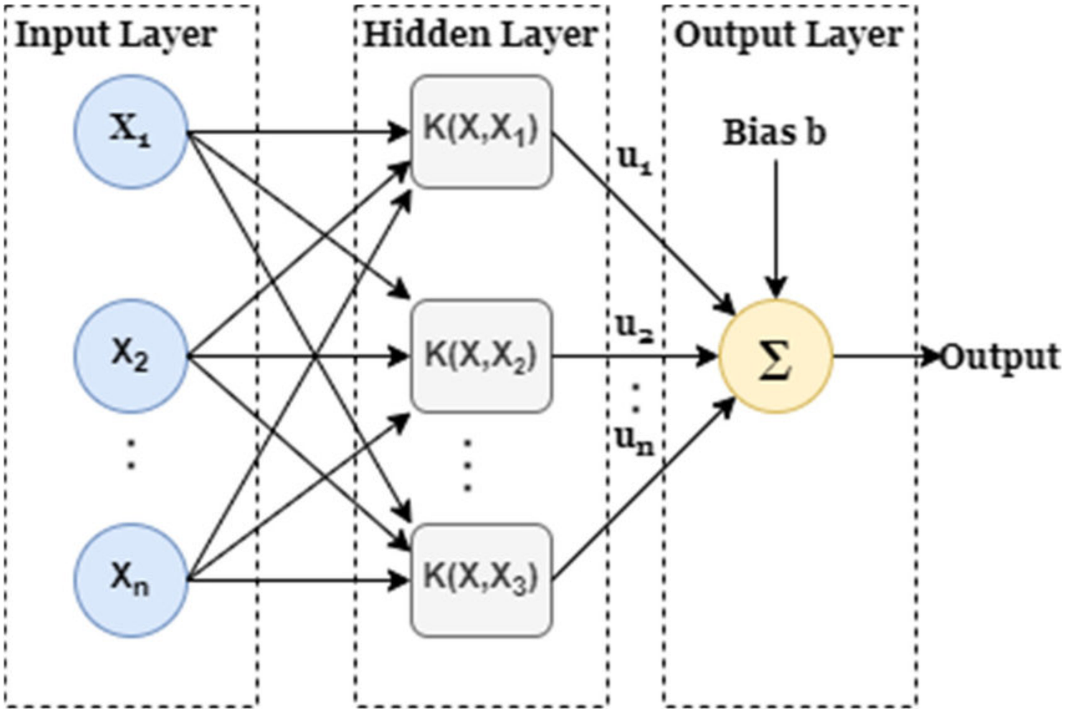

Figure 1: *SVMArchitecture*[2]

The reason we choose this algorithm is because:

- This is an algorithm that works effectively with high dimensional spaces.

- The algorithm consumes little memory because it only uses points in the support set for prediction in the decision function.

- We can create multiple decision functions from different kernel functions. Even using the right kernel can help improve the algorithm significantly.

3.2 Keras Word Embedding

In natural language processing (NLP), Keras provides an efficient way to represent words for use in deep learning models with the Embedding layer. This layer essentially translates words, which are discrete symbols, into dense vectors. These vectors capture semantic relationships between words and are valuable for tasks like sentiment analysis, topic modeling, and machine translation.

Concept

- Words are converted into one-hot encoded vectors, where each word corresponds to a vector with zeros except for a single "1" at the index representing that word. This method is inefficient and captures no semantic relationships.
- Word embeddings address this by mapping words to dense vectors of fixed size (e.g., 50, 100, 300 dimensions). These vectors encode semantic similarities and differences between words.

Keras Embedding Layer

- Takes integer-encoded tokens as input, where each token represents a unique word in the vocabulary.
- Maintains a weight matrix (lookup table) where each row represents the embedding vector for a word.
- During training, the model adjusts the weights in this matrix to learn "better" representations of the words.

Benefits

- Reduced data sparsity: Dense vectors are more efficient than one-hot encoding, especially for large vocabularies.
- Capture semantic relationships: Embeddings encode similarities between words, enabling the model to better understand context and meaning.
- Improves model performance: Using word embeddings can significantly boost the accuracy of NLP tasks compared to one-hot encoding.

Approaches to Word Embedding in Keras

- Train your own embeddings: The Embedding layer learns an embedding matrix from scratch based on your training data. This can be useful for smaller datasets or specific domains.
- Use pre-trained embeddings: Popular pre-trained embeddings like GloVe and Word2Vec offer pre-trained vectors for thousands of words. These can be directly loaded into the Embedding layer, saving training time and potentially improving performance.

3.3 FastText Word Embedding

FastText is a popular word embedding technique used to represent words as numerical vectors. These vectors capture the semantic relationship between words, meaning words with similar meanings have similar vectors. This allows us to use these vectors for various NLP tasks like: Question answering, Text classification, Named entity recognition, Machine translation.

FastText Word Embedding

- Unlike other word embedding techniques like Word2Vec, which represent words as entire units, FastText breaks down words into subword units called character n-grams. These n-grams are small snippets of characters, like bi-grams (two-letter combinations) or tri-grams (three-letter combinations).
- Each n-gram is then assigned a vector, and the final vector representation of a word is the sum of the vectors of all its n-grams. This approach has several advantages:
  - Handles out-of-vocabulary words: Even if a word hasn't been seen before, FastText can still generate a vector for it based on its n-grams, which might have been seen in other words. This makes it more robust to unseen data.
  - Captures morphological relationships: Words with similar prefixes or suffixes often have related meanings. By using n-grams, FastText can capture these morphological relationships and represent them in the word vectors.

  - More efficient: Breaking down words into n-grams allows FastText to learn smaller and more efficient vector representations compared to techniques that treat entire words as units.

Benefits

- More accurate representation of rare and unseen words.
- Improved performance on various NLP tasks like text classification, sentiment analysis, and question answering.
- Smaller model size compared to some other techniques.
- Faster training due to efficient subword-based approach.

3.4 PhoBert Word Embedding

PhoBert is a pre-trained language model specifically designed for Vietnamese. It's based on the popular BERT architecture but adapted to handle the intricacies of the Vietnamese language, including complex word segmentation and tonal variations. Two versions are available: PhoBert-base and PhoBert-large, catering to different computational resources and performance needs.

PhoBert Word Embedding

- Like other language models, PhoBert learns semantic relationships between words during its training process. This knowledge is then encoded in dense, fixed-size vectors for each word, known as word embeddings.
- Compared to traditional one-hot encoding, PhoBert embeddings capture subtle semantic nuances, word similarities, and contextual understanding.

Benefits

- Improved accuracy in NLP tasks: Utilizing PhoBert embeddings significantly boosts the performance of various NLP tasks such as:
  - Named entity recognition (NER): Identifying people, locations, and organizations in Vietnamese text.
  - Part-of-speech tagging (POS): Categorizing words based on their grammatical role in a sentence.
  - Dependency parsing: Analyzing the syntactic structure of sentences.
  - Sentiment analysis: Understanding the sentiment expressed in Vietnamese text.
- Efficient representation: Word embeddings are much more compact than one-hot encoding, allowing for efficient processing and smaller model sizes.
- Handling unseen words: PhoBert's subword-based approach can generate reasonable embeddings for words not encountered during training, improving generalization.

4 Experiment

We implement model machine learning using sklearn and train it on Kaggle with GPU P100. In this model we use default parameter. To evaluate model we use Accuracy, Confusion Matrix, F1-score as the evaluate.

4.1 Dataset and Proprocessing

We perform tokenization, remove stop word, replace teen code. Then we pad the sequence to fix length and convert them to the word embedding. After that we use $train_test_split$ in sklearn library to devide data. With the Vietnamese data set of 30000 lines divided into 21000 lines for training and 9000 lines for testing, with 50% negative and 50% positive shown in table below :

label	Train	Test
0	10442	4558
1	10558	4442

4.2 Results

We evaluate our proposed model based on Confusion matrix, accuracy and F1-score after using word embedding combined with SVM model:

Confusion matrix of Keras word embedding

	True diagnosis		
	Positive	Negative	Total
Positive	2535	2023	4558
Negative	1699	2743	4442
Total	4558	4442	9000

Confusion matrix of Fasttext word embedding

	True diagnosis		
	Positive	Negative	Total
Positive	3834	724	4558
Negative	684	3758	4442
Total	4558	4442	9000

Confusion matrix of PhoBert word embedding

	True diagnosis		
	Positive	Negative	Total
Positive	3926	632	4558
Negative	600	3842	4442
Total	4558	4442	9000

Accuracy is a metric used to evaluate the performance of a classification model. It represents the ratio of correctly predicted instances to the total instances in the dataset. The accuracy is typically expressed as a percentage.

$$Accuray = \frac{TP + TN}{TP + TN + FP + FN}$$

Word Embedding	Accuracy
Keras	58.6%
Fasttext	84.4%
PhoBert	86.3%

Table 1: Accuracy

The F1 score is a metric in machine learning that combines both precision and recall into a single value. It is particularly useful in situations where there is an imbalance between the classes in a dataset. The F1 score is the harmonic mean of precision and recall, and it provides a balance between these two metrics.

$$F1 = \frac{2 * (\textbf{precision} * \textbf{recall})}{\textbf{precision} + \textbf{recall}}$$

Word Embedding	F1-score
Keras	59.6%
Fasttext	84.2%
PhoBert	86.2%

Table 2: F1-score

5 Conclusion

The conclusion of this study represents a significant milestone in the application of Sentiment Analysis for the Vietnamese language, achieving a model accuracy of approximately 86.3%. The utilization of PhoBert Word Embedding has demonstrated notable performance and stability throughout the model training process.

Given the rapid technological advancements and the substantial increase in online interactions, Sentiment Analysis plays a crucial role in evaluating and understanding user opinions within textual content. Particularly when applied to the Vietnamese language, our model has showcased its ability to capture a diverse range of emotional nuances within text.

However, there remain challenges and opportunities to optimize the model, including improving accuracy, addressing specific language nuances, and expanding applications to various domains. This opens up potential avenues for future research in the development and optimization of Sentiment Analysis models for the Vietnamese language.

By employing an effective combination of word embedding techniques and machine learning approaches, this research aims to contribute to the understanding of emotion processing in natural language and provide practical tools for real-world applications in the Vietnamese language market.

References

- 1.Shai Shalev-Shwartz, Yoram Singer (March 2011) Pegasos: Primal estimated sub-gradient solver for SVM.

- 2. Vaibhav Kadam, Satish Kumar, Arunkumar Bongale, Seema Wazarkar, Pooja Kamat and Shruti Patil (May 2021) Enhancing Surface Fault Detection Using Machine Learning for 3D Printed Products.

- 3. Na'Im R Tyson (April 2018) Word Embedding Models & Support Vector Machines for Text Classification.

- 4. Shameer Bashira, Arvind Selwalb (2021) A comprehensive survey of Sentiment Analysis: Word Embeddings approach, research challenges and opportunities. ICICNIS 2021

- 5. Tomas Mikolov, Kai Chen, Jeffrey Dean, Greg Corrado (Sep 2013) Efficient Estimation of Word Representations inVector SpaceTomas.

- 6. Jeremy Howard, Sebastian Ruder (May 2018) Universal Language Model Fine-tuning for Text Classification

- 7. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov (2016) Enriching word vectors with subword information.

- 8. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (May 2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

- 9. Dat Quoc Nguyen, Anh Tuan Nguyen (Nov 2020) PhoBERT: Pre-trained language models for Vietnamese. EMNLP 2020, pages 1037–1042