Vietnamese Short Text Classification via Distributed Computation

Hiep Xuan Huynh[1]*, Linh Xuan Dang[2], Nghia Duong-Trung[3], Cang Thuong Phan[4]
Can Tho University, Can Tho city, Vietnam[1,2,4]
Technische Universität Berlin, Berlin, Germany[3]

***Abstract*—Social networking has been growing rapidly in Vietnam. The sharing information is diverse and circulates in many forms. It requires user-friendly solutions such as topic sorting and perspectives analysis in analyzing community trends, advertisements or anticipating and monitoring the spread of bad news. Unfortunately, Vietnamese is highly different from other languages and little research has been conducted in the literature on messages classification. The implementation of machine learning models on Vietnamese has not been thoroughly investigated and these models' performance is unknown when applying in a different language. Vietnamese text is a serialization of syllables, hence, word boundary identification is not trivial. This research portrays our endeavor to construct an effective distributed framework for addressing the task of classification of short Vietnamese texts on social networks using the idea of probability categorization. The authors argue that addressing the task sharps the successful combination of machine learning, natural language processing, and ambient intelligence. The proposed framework is effective and enables fast calculation, suitable for implementation in Apache Spark, meeting the demand for dealing with large amounts of textual data on the current social networks. Our data has been collected from several online text sources of 12412 short messages classified into five different topics. The evaluation shows that our approach has achieved an average of 82.73% classification accuracy. Thoughtfully learning the literature, we could state that this is the first attempt to classify short Vietnamese messages under a distributed computation framework.**

***Keywords*—Short text classification; Naïve Bayes; Apache Spark; Vietnamese; distributed computation**

I. Introduction

Social networking, also called virtual social network is a service that connects members on the Internet with many different purposes regardless of space and time. The online social network is a representative of Web 2.0 to simulate real social relationships using Web technology to connect members and allow them to create and share information with each other through mechanisms such as making friends, chatting, liking, sharing, tagging photos, commenting, or subscribing to a blog, a channel. In Vietnam in particular and the world in general, Facebook is the most popular social network. A person who joins a social network will be able to post or comment on another person's post. A person's post can be thoughts of self-expression, feeling of subjects, commenting on a problem or simply announcing an event. These messages, if interested by many people, will be shared by people and continue to be commented and become hot topics. Innovative social networks completely change how netizens link together and become an inevitable part of every day for hundreds of millions of members around the world. These services have provided platform for the participants to find friends and partners: based on groups (such as school names or city names), based on personal information (such as e-mail addresses, phone numbers), based on personal interests (sports, movies, books, or music), areas of interest (business, sales). According to [1], up to four Southeast Asian countries are in the top ten countries with the most Facebook users. In particular, Vietnam ranks 7th with 64 million users, accounting for 3% of the total global Facebook accounts. Thailand is right behind Vietnam, at No. 8 with 57 million users. Indonesia and the Philippines rank No. 4 and No. 6 with 126 million and 69 million accounts respectively. Capturing the popularity of social networks in Vietnam, many business components including companies, small and medium enterprises and especially online business individuals have found many measures to take advantage of its promotion to meet its business purpose.

Social networks' success has drawn the mindfulness of research on natural language processing. Among the significant exploitation, text classification has been described as one of the most important tasks, becoming a major discipline of the information systems [2]. The message on a social network can be a message between two people, a status line, or a comment of a certain status line. The message needs to have certain content. A person who wants to post an article on a personal page usually knows the topic he is talking about. Social media posts often have content that is not too long and only shows emoticons on a topic. A user can post one or more posts at the same time. These posts are generated continuously and without quantity limits. The spread of these messages is also very fast and wide [3]. The theme of messages on social networks is almost the same as the topics of common types of documents such as Science, Business, Law, Health, Sports, Technology. The problem is that collecting sample data takes a lot of time and effort. But in the classification problem, the sample dataset plays a very important role. In addition, there is an ambiguity about the topics of social media posts, because the messages do not always present a clear purpose for their users. An article can sometimes belong to two or three different topics when placed in another context. The messages themselves are short documents, contain very little information. Most of the messages on social network have images or videos attached, which limit the content analysis. The documents accompanying with these videos or images are sometimes just an unknown introduction, the core content of the article is in the videos and images. The amount of articles that are too large requires building on a large data processing platform, towards real-time processing to meet the needs of huge data analysis. In

*Corresponding authors.

Vietnam, there are not many research groups in the field of social network analysis. There are not many studies on the classification of published messages especially for Vietnamese so that it is difficult to compare results and evaluation.

Consequently, the need to classify the text-based messages electronically available has significantly grown. One can argue that automated text classification is one of the most crucial tasks in social media analytics [4]. Many research papers have been conducted to solve such problems as topic modeling [5], [6], [7], [8], geolocation [9], [10], and document classification [11]. Although many Vietnamese documents are electrically available, no one has been recently conducted on short Vietnamese messages classification. This research portrays our endeavor to construct an effective distributed framework for addressing the task of classification of short Vietnamese texts on social networks. We have collected a large amount of Vietnamese textual sources of 12412 messages and investigated message representation, word tokenization, and learning method that are suitable for the requirements of acceptable classification accuracy and distributed calculation.

The rest of this research paper is organized as follows. First, we discuss previous research on short Vietnamese text classification in Section (II). Next, we summarize fundamental materials and methodology in Section (III). The proposed framework for addressing the probabilistic classification of short Vietnamese messages is presented in Section (IV). In Section (V), the experiments are thoughtfully discussed. And finally, conclusion is stated in Section (VI).

II. Related Works

Text classification is a classic problem in data mining and machine learning [12]. The goal of the main classification problem is to find the appropriate topic in a set of predefined topics. Streams of text classification research has been done in several top-tier conference, journal and workshop. Criteria to select the appropriate topic for documents based on the similarity between them with the text in the training material semantically. Automatic sorting of text into a topic makes it easier to organize, store and query documents later. Besides, text classification is also used to assist in the process of searching, extracting information [13], [14].

Over the years, the research field of ambient intelligence has witnessed remarkable achievements. Scholars have raised the potential integration between natural language processing and ambient intelligence that can further generate cutting-edge research leading to substantial technological advances. A vibrant part of many information repositories is textual sources across various media, usually in the nonstructural form. One can conclude that the capacity to process and make use of nonstructural information can boost ambient intelligence systems to a much higher level of quality. Applying machine learning within the task of text classification involves transforming a variety of textual sources into structured knowledge. The authors argue that the combination of machine learning, natural language processing, and ambient intelligence opens up exciting research challenges [15].

A text classification task contains four different research disciplines: feature extraction, dimensionality reduction, classification algorithms, and evaluation. Texts are unstructured data, and we need to transform them in a feature space. Several common feature extraction techniques from basic to more advanced are TFIDF, Word2Vec [16], and Glove [17]. The most crucial step of completing a text classification task is to choose the best classifier, starting from non-parametric techniques, e.g., k-nearest neighbor, to simply logistic regression, to tree-based classifiers, to deep learning-based models. Recently, we have witnessed the success of deep learning approaches over previous classification models due to its excellent performance and capacity to address non-linear relationships within text data. However, regarding the feasibility characteristics, Naïve Bayes algorithm, which is a very computationally inexpensive and low amount of memory consumption, is one of the most generative model. These implementations have one characteristic in common: the classifier is trained on a simple machine, ranging from a regular personal computer to a dedicated server. The extend of training phase in case of handing a large amount of data can be broadcast to several machines in a cluster, which yields the idea of distribution computation [18], [19], [20], [21]. In the implementation, we deploy our distributed framework upon Apache Spark.

An early effort to address the task of Vietnamese text classification was conducted more than a decade ago [22]. In that paper, the authors solved the problem of automatically categorizing given textual sources into predefined categories. A comparison between statistical N-Gram language modeling and bag of words approaches has been investigated on their collected dataset. Although they achieved a good accuracy score, the implemented models were not efficient in term of computation time, e.g. only three documents/second comparing to 776 documents/second in our distributed framework. The task of automatic text categorization has been studied by comparing the performance of several term weighting schemes rather than analyzing the actual classification task [23]. Nevertheless, these approaches have investigated short messages in very different classification tasks. For the problem of classifying Vietnamese text, many research projects have been published but their work were done in an isolated environment [24], [25], [26]. Thoughtfully learning the literature, we could state that this is the first contribution to classify short Vietnamese messages under a distributed computation.

III. Materials and Methods

A. Problem Definition

Given a set of n input texts denoted as $D = \{d_1, d_2, \ldots, d_n\}$. By applying some processing techniques we will classify them into a set of m classes denoted as $C = \{c_1, c_2, \ldots, c_m\}$. An example of text classification is the arrangement of news in newspapers into corresponding categories such as Sports, Entertainment, and Society. This can be done manually by the editors but this method faces some of the following difficulties: (i) It takes a lot of time and effort. (ii) Manual classification is sometimes inaccurate because the decision depends on the understanding and motivation of the implementer. (iii) For some professional fields, experts (medical, legal, economic) are needed. The decision of several experts may be contradictable. And (iv) When the number of documents is relatively high, an expert might find it difficult to implement.

TABLE I. VIETNAMESE TEXT AFTER TOKENIZATION PROCESS BY VNTOKENIZER SOFTWARE. THE UNDERSCORE CHARACTER IS ADDED BETWEEN INDIVIDUAL WORDS TO FORM AN APPROPRIATE MEANINGFUL TOKEN. THE TRANSLATED PARTS ARE FOR EXPLANATION IN THIS PAPER ONLY

Message id	Message content
[00164]	Đúng là làm Manuel_Neuer không dễ Bài_học đau_lòng cho Iker_Casillas. *Translated:* It is true that Manuel Neuer is not easy to do. The painful lesson for Iker Casillas.
[00202]	Trong hàng triệu triệu cổ_động_viên hướng về đội_tuyển quốc_gia Việt_Nam ngày hôm_nay những_ai có_mặt tại sân_vận_động chứng_kiến trực_tiếp trận đấu có_lẽ là những người may_mắn nhất nhưng có_lẽ cũng là những người vất_vả nhất Thời_tiết ở thành_phố Thường_Châu những ngày này rất khắc_nghiệt với nhiệt_độ âm và tuyết đã rơi trắng đường Nhưng chẳng điều gì ngăn được bước chân của các cđv Việt_Nam. *Translated:* Millions of fans are heading to Vietnam national team today. Those presented at the stadium witnessing the match directly were probably the luckiest people, but perhaps also the hardest. The weather in Changzhou City these days is very harsh. The temperature drops to negative degrees Celsius and white snow has fallen. But nothing stops the footsteps of Vietnamese fans.
[03189]	Hè sắp đến rồi CÙNG GIẢI NHIỆT MÙA HÈ THÔI Hay lên kế_hoạch cho những chuyến vi_vu xả_hơi để tận_hưởng thắng_cảnh tại nước_ngoài hay các biển đảo mới được biết đến như Đảo_Nam_Du hoặc tận_hưởng ngắn ngày cho một chuyến du_lịch về vùng sông_nước hoà_mình với thiên_nhiên chưa Hãy cùng Du_Lịch_Việt chuẩn_bị cho một chuyến du_lịch cùng mùa. *Translated:* The summer is coming and LET'S ENJOY THE SUMMER SEASON. Have you planned for a relaxing trip to enjoy the sights in foreign countries or new islands known as Nam Du or enjoying a short trip to rivers, mix with nature yet? Let's travel with Vietnam Travel to prepare for a trip with the season.
[04037]	Chiều Về lại thị_trấn Dương_Đông Kết_thúc chuyến tham_quan. *Translated:* Back to Duong Dong town in the afternoon. End of the tour.
[06840]	Từ hôm_nay đến hết đi Easy_Taxi tại thủ_đô Hà_Nội và thành_phố Hồ_Chí_Minh để có ngay voucher mua_sắm trên ứng_dụng Lazada trị_giá Easy_Taxi là một lựa_chọn gọi taxi ở các quán bar khách_sạn, siêu_thị nhà_hàng hay bất_cứ nơi đâu trở_nên tuyệt_vời nhanh_chóng và an_toàn hơn bao_giờ hết Tải ngay ứng_dụng Easy_Taxi tại đây IOS bitly easytaxiios ANDROID bitly easytaxiandroid WINDOW PHONE bitly easytaxiwp Voucher sẽ được gửi qua email và có giá_trị_sử_dụng đến hết Dành cho đơn hàng trên và chỉ áp_dụng khi mua qua ứng_dụng Lazada trên điện_thoại. *Translated:* From today to the end, go with Easy Taxi in Hanoi and Ho Chi Minh City to receive your voucher on Lazada application right away. Easy Taxi is a great option for making taxi calls at hotel bars, restaurants or anywhere, easily and safely than ever. Download now Easy Taxi application at bitly_easytaxiiosm, ANDROID bitly_easytaxiandroid and WINDOW PHONE bitly_easytaxiwp. The voucher will be sent via email and will be valid for use on the above orders and only applicable when purchased via Lazada on the phone.
[07040]	Cấp_báo Cô Ba_Sài_Gòn đã có_mặt tại Lazada xinh quá Ad muốn ngất xĩu rồi_đây. *Translated:* The newspaper Ba Ba Saigon was present at beautiful Lazada so Ad wanted to faint.
[09677]	Dùng_Android mà bỏ_qua các mẹo này là hối_hận đó nhé. *Translated:* Using Android without skipping these tips is regretful.
[12364]	Nỗi lo rụng tóc khi hoá_trị ung_thư sẽ không còn nữa. *Translated:* The worry of hair loss when cancer chemotherapy is no longer available.

B. Text Pre-processing

Data pre-processing is the first important step of any data mining process. It makes data in its original form easier to observe and explore. For the problem of text classification, due to specific characteristics, each language has its own characteristics. The preprocessing process will help improve sorting efficiency and reduce the complexity of the training algorithm. Depending on the purpose of the classifier, we will have different preprocessing methods, such as

- Convert text to lowercase and correct spelling errors.
- Remove punctuation marks (if no sentence separation is performed).
- Remove special characters ([], [.], [,], [:], [”], [”], [;], [/], [[]], [J̃], [J́], [!], [@], [#], [$], [%], [Ĵ], [&], [*], [(], [)]), digits.
- Separate of words by single word method (English) or compound words (Vietnamese).
- Remove the stopwords, e.g. the words that appear most in the text that are not meaningful when participating in text classification.
- Standardize the words, switch back from the original (usually applicable to English).
- Convert text into vectors as input for classification learning machine.

C. Text Transformation and Presentation

One of the first tasks in dealing with text classification is to choose an appropriate text representation model. A raw document (string form) needs to be transferred to another model to facilitate representation and calculation. Depending on the different classification algorithms, we have our own representation model. The vector space model is one of the simplest and most commonly used models in this task. A text source is represented in the form, with an n-dimensional vector to measure the value of the text element. A document is expressed as a collection of tokens and/or words, each token is considered an attribute or characteristic and the text corresponds to an attribute vector. After identifying the properties, we need to calculate the attribute value (or weighted keyword) for each text.

We discuss term frequency-inverse document frequency

(TFIDF) [27], one of the most fundamental techniques for retrieving relevant documents from a text source or from a collection of text sources. Although TFIDF is fundamental, it statically proves the effectiveness in text mining [28], [29]. Having gathered all the tokens from the tokenization step, all given messages are converted from bag-of-words representations of token counts into sparse vectors with TFIDF weights. TFIDF is an acronym of term frequency-inverse document frequency, and this score often used in text processing and information retrieval. The idea of TFIDF weight is to calculate a score that expresses the relative importance of words in the documents. The score is statistically measured by evaluating the significance a token gains in a document and in a collection. The importance of a word is proportionally judged by counting the number of times it exists in a document while compensating its appearance in the corpus. In this way, we discard grammar structure, words' order, and part-of-speech. It is intuitive that the frequency with which a token appears in a message could indicate the extent that the message pertains to that token. The TFIDF weight reflects how significant a token gets to a message. The more appearance a token exists in many messages, the more penalty it gets punished. The best characteristics of the tokens to the message is measured by the highest score of TFIDF.

TFIDF weight is expressed as

$$TFIDF = TF * IDF \ , \tag{1}$$

where *TF* is how many times a word appears in a document, and *IDF* is the logarithm score of the number documents in the whole corpus divided by how many documents that the specific word appears. More precisely, the *TF* is calculated as follows:

$$\text{TF}(w, d) = \frac{n^d(w)}{|d|} \ , \tag{2}$$

where the number of times a word w appears in a document d and the total number of words in d are $n^d(w)$ and $|d|$, respectively.

While the TF is calculated on a per-document basis, the IDF is computed on the basis of the entire corpus. Thus, the IDF is calculated as follows.

$$\text{IDF}(w) = \log \frac{|C|}{n^C(w)} \ , \tag{3}$$

where $|C|$ represents the number of documents in the corpus and $n^C(t)$ represents the number of documents that contains the word w.

TABLE II. STATISTICS OF DATA COLLECTION

Topic	Site	Likes / Followers	Raw messages	Filtered messages
Sport	banthethaovtv	135000 / 184000	2511	2386
News	thoisuvtv	616000 / 715000	2989	2278
Traveling	dulichviet	144000 / 144000	3344	2986
Sales	lazadavietnam	22 million /22 million	3039	2343
Technology	thegioididong	3 million / 3 million	3088	2419
Total			**14971**	**12412**

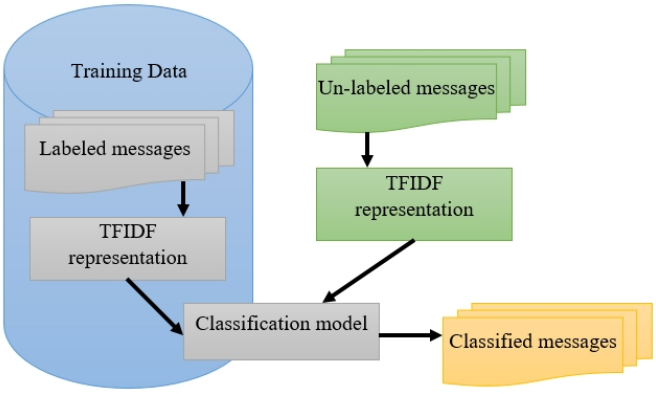

Figure 1. The Overall Architecture of a Textual Classification Model.

D. Naïve Bayes Classifier

Naïve Bayes is a popular machine learning model thanks to its great performance [30]. It merely meant as a machine learning approach that we utilize in the work. Readers may refer to mathematics or probabilities machine learning textbook [31] for advanced information.

Given an observation, model's parameters and a label represented by a vector $\mathbf{u}$, a set of parameter ω and a target $t = c$ respectively, the generative model to classify $\mathbf{u}$ is defined as follows:

$$P(t = c|\mathbf{u}, \omega) = \frac{P(t = c|\omega)P(\mathbf{u}|t = c, \omega)}{\sum_{c'} P(t = c'|\omega)P(\mathbf{u}|t = c', \omega)} \tag{4}$$

where $P(\mathbf{u}|t = c, \omega)$, $P(t = c|\mathbf{u})$, and $P(t = c)$ are the class-conditional density, the class posterior, and the class prior respectively. Proportionally, Equation (4) can be computed as in the following equation:

$$P(t = c|\mathbf{u}, \omega) \propto P(t = c|\omega)P(\mathbf{u}|t = c, \omega). \tag{5}$$

Moreover, the class-conditional density $P(\mathbf{u}|t = c, \omega)$ in Equation (4) is calculated as follows:

$$P(\mathbf{u}|t = c, \omega) = \prod_{i=1}^{D} P(u_i|t = c, \omega_{ic}) \tag{6}$$

which we yield a Naïve Bayes classifier.

IV. OUR PROPOSED FRAMEWORK FOR ADDRESSING PROBABILISTIC CLASSIFICATION OF SHORT VIETNAMESE MESSAGES

A. Design Concept

In this research, we introduce a framework to explore and label topics for short Vietnamese messages according to the traditional text classification procedure which is presented in Fig. (1). Nevertheless, we employ the message classification via a distributed framework Apache Spark [32], [33]. The complete design of our proposed framework is presented in

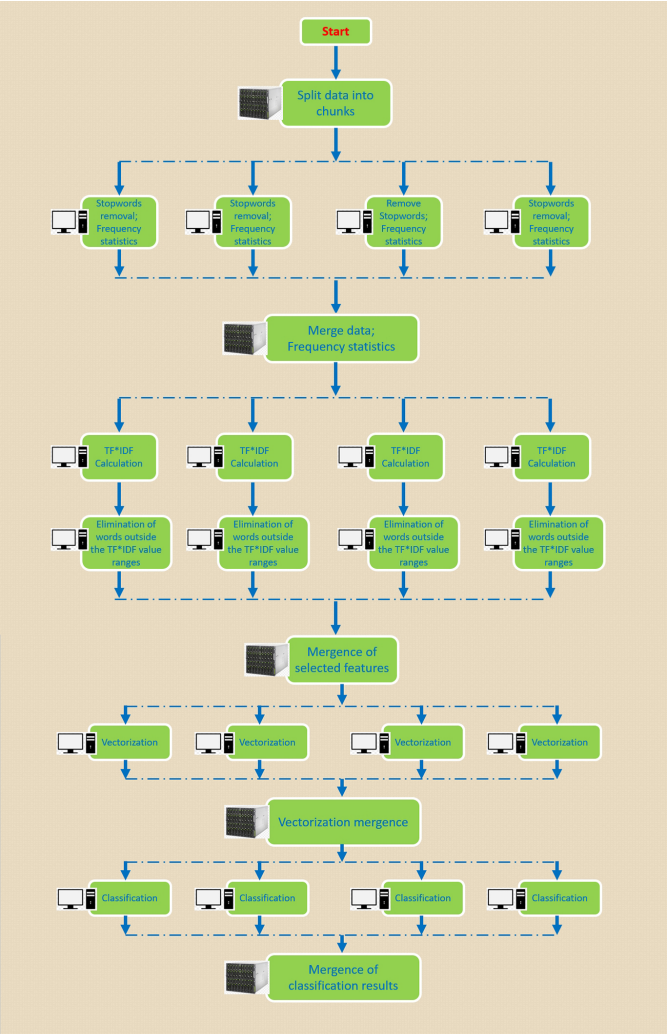

Figure 2. The Overall Architecture of our Proposed Distributed Framework.

Fig. (2). Consequently, heavy load, e.g. data pre-processing, vectorized representation, and classification, of a traditional machine learning task is effectively done in parallel.

B. Text Classification Pipeline

Depending on each specific case, the text classification problem will have different processes. Here are some basic steps: (1) Pre-processing data is a step to clean the data before starting to process in the next steps. It includes some concepts of natural language processing such as removing redundant characters, deleting stop words that don't make much sense, removing words that appear in most texts, spell checking. (2) Separation is an extremely important step, especially for Vietnamese. There are many ways to separate different words, we will learn more in the next section. (3) Document representation is the pipeline of transforming the input text data set into attributes compatible with the classification model in the next steps, facilitating easier problem-solving. (4) Characteristic extraction is the step to find the core characteristics from the original dataset or in other words, to choose a typical characteristic that is representative of the dataset as the basis for the algorithm. (5) Model training is the step in which we use machine learning algorithms to find the best model. Finally, classification (6) is the use of the trained model in the above step to conduct classification for the dataset in practice.

C. Distributed Computing Framework

Many people can agree that one of the most successful cluster computing platform is Apache Spark due to its great ability to compute fast and can be generally utilized in many research and business domains [33]. Hinging on the efficiency of supporting a broad variety of computations' types, Apache Spark can handle stream processing and queries by the extension of the well-known MapReduce model [34]. Moreover, a physical execution engine called the DAG scheduler gains great achievement in processing batch and streaming data. From the very first idea of design, Apache Spark executes computation directly in machine's memory which in turn boosting the computing speed significantly. Apache Spark provides multi-purposed APIs that support many modern programming languages, e.g. in Python, Scala, and Java. Spark Core is the main architecture of Spark consisting of components for fault recovery, memory management, optimization, task scheduling, and storage interaction. Apache Spark's main programming abstraction is resilient distributed datasets, or called RDDs in short, is a distributed collection of elements defined by Spark's main architecture. During computation, RDDs are distributed around a cluster of machines and can be performed in parallel effectively and transparently. A wide variety of machine learning functionalities are integrated into Spark's MLlib library [35]. Apache Spark can be deployed in a standalone machine or associated with Mesos [36]. The overall architecture of our distributed framework is illustrated in Fig. (2).

V. Experiments

A. Data Collection

We utilize a commercial tool called Facebook Fplus [37] developed by a domestic company FPLUS24H. Corresponding to each topic, we choose Facebook pages based on the number of likes and followers in the belief that these pages will focus on writing articles related to their main subject. For each topic, we select the Facebook pages with the most number of likes and followers compared with other similar pages. Our statistics of data collection is presented in Table (II). The process of filtering messages by topic is done through the following steps. First, we filter empty messages, or messages holding too little content leading to unclear meaning and unknown topics. Second, we remove messages embedding videos, images with accompanying texts that do not show the right content. Next, we filter wrong spelling messages, e.g. without Vietnamese accents. Fourth, we remove messages that the topics do not match with the contents. And finally, we split messages into separate files for easy storage and processing in Apache Spark.

B. Vietnamese Text Tokenization

For the tokenization task, we utilize vnTokenizer [38] in our research, see Table (I). The combination of tokenization accuracy among software is out of the main concern of this research paper. We utilize a list of 1942 Vietnamese stopwords [39] in our data processing.

C. Evaluation Metrics

Suppose we are solving a binary classification task with a labeled dataset $\mathcal{D} = \{\mathbf{x}_i, t_i\}$. Given a threshold parameter ϕ that guilds our decision rule $g(\mathbf{x})$ We also define m_+ the total of condition positives, m_- the total of condition negatives, $\hat{m}_+$ the total predicted condition positives, $\hat{m}_-$ the total predicted condition negatives, and m the total population.

We can compute the sensitivity, also known as true positive rate (TPR), probability of detection, or recall by using:

$$\text{TPR} = \frac{\text{TP}}{m_+} \approx P(\hat{t} = 1 | t = 1) \ . \tag{7}$$

Similarly, we can compute the fall-out, also known as false positive rate or probability of false alarm by using:

$$\text{FPR} = \frac{\text{FP}}{m_-} \approx P(\hat{t} = 1 | t = 0) \ . \tag{8}$$

The true negative rate (TNR) or specificity is defined as follows:

$$\text{TNR} = \frac{\text{TN}}{m_-} \approx P(\hat{t} = 0 | t = 0) \ . \tag{9}$$

The false negative rate (FNR) or miss rate is calculated as follows:

$$\text{FNR} = \frac{\text{FN}}{m_+} \approx P(\hat{t} = 0 | t = 1) \ . \tag{10}$$

If we work with a dataset for binary text classification when the number of negatives is very large or a dataset for multi-class text prediction when class imbalance exists, considering TPR, FPR, TNR and FNR themselves is not very informative. Before going further, we define positive predictive value (PPV) or precision as follows:

$$\text{PPV} = \frac{\text{TP}}{\hat{m}_+} \approx P(t = 1 | \hat{t} = 1) \ . \tag{11}$$

By combining Equation (7 and 11), we can compute F1-score as follows:

$$\text{F1-score} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \ , \tag{12}$$

which is widely used in information retrieval systems.

D. Implementation

The fundamental goal of machine learning models is to make accurate predictions on unseen observations. In order to estimate the strength of a particular learning model, practitioners usually split data into several proportions which serves for specific purposes in the machine learning pipeline. More specifically, the data is split into a training set containing samples to train the model and a test set consisting of instances to pretend an unbiased evaluation of the investigated learning model. We set up five different splitting schemes by tuning various split ratios. For each topic, we eventually examine how the proposed framework performs on these five dataset splitting schemes.

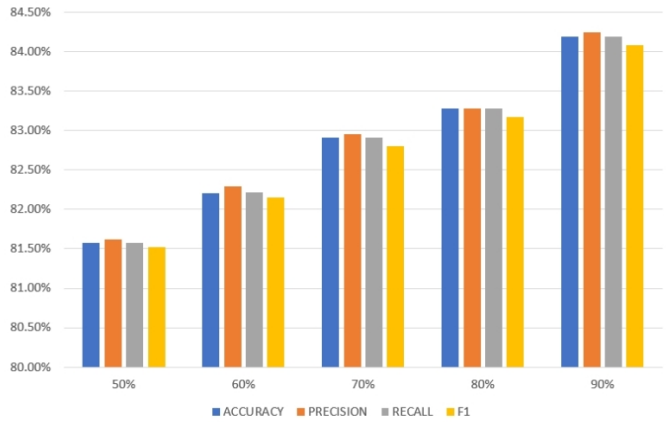

Figure 3. The Framework Performance on all Five Dataset Splitting Schemes

- Splitting scheme a. The percentage of the training and test parts is 50% and 50% respectively. We denote it as $50|50$ and 50% hereafter.
- Splitting scheme b. The percentage of the training and test parts is 60% and 40% respectively. We denote it as $60|40$ and 60% hereafter.
- Splitting scheme c. The percentage of the training and test parts is 70% and 30% respectively. We denote it as $70|30$ and 70% hereafter.
- Splitting scheme d. The percentage of the training and test parts is 80% and 20% respectively. We denote it as $80|20$ and 80% hereafter.
- Splitting scheme e. The percentage of the training and test parts is 90% and 10% respectively. We denote it as $90|10$ and 90% hereafter.

All experiments have been conducted on a normal laptop including distributed computing infrastructure and virtual machines. The environment specifications are CPU Intel Core i7 MQ, 8GB of RAM, graphics card NVIDIA GT 740M, Apache Spark 2.2.0, IDE IntelliJ IDEA 2017 ver 2.6, Scala programming language.

E. Experimental Results

All experiments have been conducted five times to assure the performance stability of the system. The authors then reported average scores and their standard deviation, e.g., a measure of the amount of variation or dispersion of our five-times computation. A low standard deviation indicates that the scores tend to be close to the mean of 5 times, while a high standard deviation indicates that the scores are spread out over a broader range. The experimental process is split into two scenarios. First, we investigate the performance on each topic separately. We present the experimental results on five topics (see Table (III) for Sports, see Table (IV) for News, see Table (V) for Traveling, see Table (VI) for Sales, see Table (VII) for

Technology). Second, we investigate the performance on the complete dataset, see Table (VIII) and Fig. (3).

TABLE III. THE CLASSIFICATION PERFORMANCE ON THE SPORTS TOPIC

Sports			
Splitting scheme	# Test instances	# Accurate prediction	Percentage
50|50	1183.2 ± 9.7570	963.8 ± 10.2323	0.8145 ± 0.0051
60|40	916.2 ± 18.3084	762.8 ± 28.0303	0.8323 ± 0.0157
70|30	705 ± 27.3404	591.4 ± 18.4607	0.8391 ± 0.0168
80|20	465.6 ± 12.7590	389.4 ± 12.9537	0.8363 ± 0.0139
90|10	223 ± 48.0468	201.4 ± 16.1338	0.8281 ± 0.0313

TABLE IV. THE CLASSIFICATION PERFORMANCE ON THE NEWS TOPIC

News			
Splitting scheme	# Test instances	# Accurate prediction	Percentage
50|50	1169 ± 25.4263	992.8 ± 23.7002	0.8492 ± 0.0072
60|40	946.8 ± 16.6943	809.6 ± 10.5023	0.8552 ± 0.0118
70|30	711.4 ± 35.9277	614.6 ± 27.8980	0.8641 ± 0.0076
80|20	474.4 ± 34.5079	410.2 ± 32.5079	0.8646 ± 0.0233
90|10	256.2 ± 10.0598	221.2 ± 6.6858	0.8637 ± 0.0156

TABLE V. THE CLASSIFICATION PERFORMANCE ON THE TRAVELING TOPIC.

Traveling			
Splitting scheme	# Test instances	# Accurate prediction	Percentage
50|50	1497.8 ± 17.0792	1355.2 ± 17.5698	0.9047 ± 0.0047
60|40	1192 ± 41.2492	1094.6 ± 34.1291	0.9183 ± 0.0058
70|30	896 ± 22.0340	816.4 ± 20.6712	0.9111 ± 0.0083
80|20	593 ± 16.9558	538.4 ± 16.4103	0.9079 ± 0.0147
90|10	304.8 ± 13.9713	282.4 ± 15.0266	0.9263 ± 0.0125

TABLE VI. THE CLASSIFICATION PERFORMANCE ON THE SALES TOPIC

Sales			
Splitting scheme	# Test instances	# Accurate prediction	Percentage
50|50	1150 ± 17.2481	817.8 ± 29.1753	0.7109 ± 0.0167
60|40	905.4 ± 20.3297	649.6 ± 23.7339	0.7174 ± 0.0182
70|30	668 ± 27.0277	474.2 ± 16.4225	0.7102 ± 0.0198
80|20	460.2 ± 21.2179	338.8 ± 11.3666	0.7366 ± 0.0194
90|10	226 ± 12.2882	166.8 ± 10.3778	0.7380 ± 0.0220

TABLE VII. THE CLASSIFICATION PERFORMANCE ON THE TECHNOLOGY TOPIC

Technology			
Splitting scheme	# Test instances	# Accurate prediction	Percentage
50|50	1209.6 ± 25.0059	902.6 ± 27.4918	0.7461 ± 0.0166
60|40	969 ± 25.7099	732.4 ± 21.8929	0.7558 ± 0.0074
70|30	731.6 ± 28.6408	557.8 ± 33.4917	0.7619 ± 0.0182
80|20	484 ± 21.7715	374.2 ± 6.9426	0.7740 ± 0.0269
90|10	247.2 ± 15.7066	191.2 ± 14.5670	0.7730 ± 0.0175

F. Remark and Discussion

The average execution time of the system is calculated through two phases: Phase I: Filter characters, separate words, vectorize messages, and perform in vector space model for the dataset. The average execution time is 20 minutes 55 seconds. Phase II: Divide the data into two parts, train the machine learning model, and predict the test set, calculate the accuracy of each topic, and analyze the system's efficiency. The average execution time is 16 seconds, which proves the feasibility of Naïve Bayes classifier. It works well with text data and is fast in comparison to other classification algorithms. The advantages of Naïve Bayes the biased assumption about the shape of the data distribution. The model limits the prediction capacity to data scarcity and frequency of words in the whole text source.

With the highest accuracy of about 83.18% with a minimal value of standard deviation 0.93%, the experimental results have proved the feasibility and computation stability of our proposed system. The topic with the highest predictive rate is Traveling with 92.63%. Particularly, the Sales topic has the lowest rate, with 73.80%. It can be explained for this reason because the Lazada fan page specializes in selling goods online. The categories of goods are very diverse. The standard deviation of all experiments is quite low, which indicates the performance stability of the proposed system. Furthermore, the execution time of the system is also relatively short, especially the classification process. When the data set is large enough, the learning process only occurs once, and the classification process must be repeated over time.

TFIDF is one of the most popular term-weighting schemes today as 83% of text-based recommender systems in digital libraries use it [40]. It is widely supported in many machine learning libraries and can be applied as on-the-shelf effectively. Therefore, traditional TFIDF is applied in this paper. The pitfall of this text presentation is that it ignores the semantics and syntactic of the text. The calculation time also depends on how many unique words in all text corpora. The tuning might be considered to improve the shortcomings of the IFIDF algorithm regarding the classification accuracy of machine learning models used, ignoring the calculation efficiency in the classification process. How to improve the accuracy together with efficiency is the direction for further research in the future.

To get a confirmation on how our proposed solution performs, we have conducted several experiments by replacing Naïve Bayes classifier by logistic regression, decision trees, and random forests. The operating configuration is similar, except for the models themselves. The experimental results reported in Tables IX, X, and XI have proved the advancement of our proposed solution.

VI. CONCLUSION

The problem of discovering and identifying themes for social network messages is an urgent problem in the context of the current social network explosion. The topics explored from these messages in combination with analyzing the perspective will contribute to predicting the spread of the messages. It helps develop solutions to monitor and prevent bad information, causing serious impacts, spreads on social networks. The paper has proposed and built a distributed framework for addressing probabilistic classification using Apache Spark to meet the need to handle large amounts of data. The Naïve Bayes classification method is suitable to build on a large data processing platform. Initial results for an accuracy score of about 83% and can be further improved when collecting more amount of dataset. we have also built a set of social network messages including five topics for the process of analyzing and researching social networks. The paper contributes to solving the problem of classifying the topic of short messages that are appearing on social networks in Vietnam.

REFERENCES

[1] "Tnw: The heart of tech." [Online]. Available: http://www.thenextweb.com/

TABLE VIII. THE CLASSIFICATION PERFORMANCE ON THE COMPLETE FIVE TOPICS

The complete experimental dataset						
Splitting scheme	# Test instances	# Accurate prediction	Accuracy	Precision	Recall	F1 score
50|50	6193.6 ± 78.99	5051.2 ± 101.22	0.8155 ± 0.0087	0.8158 ± 0.0089	0.8155 ± 0.0087	0.8144 ± 0.0088
60|40	5003.8 ± 52.51	4104.2 ± 65.00	0.8201 ± 0.0044	0.8206 ± 0.0043	0.8201 ± 0.0044	0.8190 ± 0.0045
70|30	3710.8 ± 75.98	3052.8 ± 69.92	0.8226 ± 0.0055	0.8236 ± 0.0055	0.8226 ± 0.0055	0.8217 ± 0.0054
80|20	2510.8 ± 44.37	2073.4 ± 29.66	0.8258 ± 0.0056	0.8269 ± 0.0059	0.8258 ± 0.0056	0.8249 ± 0.0059
90|10	1248.4 ± 29.59	1039.4 ± 19.70	0.8326 ± 0.0077	0.8344 ± 0.0088	0.8326 ± 0.0077	0.8318 ± 0.0093

TABLE IX. THE CLASSIFICATION PERFORMANCE ON THE COMPLETE FIVE TOPICS. LOGISTIC REGRESSION MODEL IS USED

Splitting scheme	Accuracy	Precision	Recall	F1 score
50|50	0.7450 ± 0.0045	0.7465 ± 0.0038	0.7450 ± 0.0045	0.7444 ± 0.0041
60|40	0.7549 ± 0.0064	0.7572 ± 0.0066	0.7549 ± 0.0064	0.7549 ± 0.0067
70|30	0.7659 ± 0.0106	0.7674 ± 0.0105	0.7659 ± 0.0106	0.7659 ± 0.0107
80|20	0.7710 ± 0.0088	0.7722 ± 0.0081	0.7710 ± 0.0088	0.7707 ± 0.0084
90|10	0.7716 ± 0.0123	0.7733 ± 0.0120	0.7716 ± 0.0123	0.7718 ± 0.0120

TABLE X. THE CLASSIFICATION PERFORMANCE ON THE COMPLETE FIVE TOPICS. DECISION TREES MODEL IS USED

Splitting scheme	Accuracy	Precision	Recall	F1 score
50|50	0.6816 ± 0.0068	0.7514 ± 0.0110	0.6816 ± 0.0068	0.6643 ± 0.0084
60|40	0.6820 ± 0.0035	0.7532 ± 0.0029	0.6820 ± 0.0035	0.6649 ± 0.0039
70|30	0.6851 ± 0.0166	0.7517 ± 0.0112	0.6851 ± 0.0166	0.6680 ± 0.0183
80|20	0.6924 ± 0.0071	0.7621 ± 0.0099	0.6924 ± 0.0071	0.6763 ± 0.0085
90|10	0.6956 ± 0.0146	0.7560 ± 0.0139	0.6956 ± 0.0146	0.6811 ± 0.0170

TABLE XI. THE CLASSIFICATION PERFORMANCE ON THE COMPLETE FIVE TOPICS. RANDOM FOREST MODEL IS USED

Splitting scheme	Accuracy	Precision	Recall	F1 score
50|50	0.6325 ± 0.0074	0.7968 ± 0.0054	0.6325 ± 0.0074	0.6071 ± 0.0058
60|40	0.6209 ± 0.0080	0.7950 ± 0.0025	0.6209 ± 0.0080	0.5949 ± 0.0084
70|30	0.6253 ± 0.0160	0.7983 ± 0.0035	0.6253 ± 0.0160	0.6003 ± 0.0171
80|20	0.6266 ± 0.0201	0.7974 ± 0.0048	0.6266 ± 0.0201	0.6007 ± 0.0206
90|10	0.6180 ± 0.0199	0.8001 ± 0.0016	0.6180 ± 0.0199	0.5933 ± 0.0227

[2] V. Basile, C. Bosco, E. Fersini, N. Debora, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti *et al.*, "Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter," in *13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2019, pp. 54–63.

[3] H. X. Huynh, B. U. Lai, N. Duong-Trung, H. T. Nguyen, and T.-C. Phan, "Modeling population dynamics for information dissemination through facebook," *Concurrency and Computation: Practice and Experience*, p. e6333, 2021.

[4] N. Duong-Trung, *Social Media Learning: Novel Text Analytics for Geolocation and Topic Modeling*. Cuvillier Verlag, 2017.

[5] T. T. Dao, T. D. Thanh, T. N. Hai, and V. H. Ngoc, "Building vietnamese topic modeling based on core terms and applying in text classification," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, 2015, pp. 1284–1288.

[6] N. Duong-Trung and L. Schmidt-Thieme, "On discovering the number of document topics via conceptual latent space," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2051–2054.

[7] N. Duong-Trung, N. Schilling, L. Drumond, and L. Schmidt-Thieme, "Matrix factorization for near real-time geolocation prediction in twitter stream." in *LWDA*, 2016, pp. 89–100.

[8] N. Duong-Trung, M.-H. Nguyen, and H. T. Nguyen, "Clustering stability via concept-based nonnegative matrix factorization," in *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, 2019, pp. 49–54.

[9] N. Duong-Trung, N. Schilling, and L. Schmidt-Thieme, "Near real-time geolocation prediction in twitter streams via matrix factorization based regression," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 1973–1976.

[10] N. Duong-Trung, N. Schilling, L. R. Drumond, and L. Schmidt-Thieme, "An effective approach for geolocation prediction in twitter streams using clustering based discretization," 2017.

[11] G.-S. Nguyen, X. Gao, and P. Andreae, "Vietnamese document representation and classification," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2009, pp. 577–586.

[12] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[13] M. M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36–54, 2018.

[14] V. B. Kobayashi, S. T. Mol, H. A. Berkers, G. Kismihok, and D. N. Den Hartog, "Text classification for organizational researchers: A tutorial," *Organizational research methods*, vol. 21, no. 3, pp. 766–799, 2018.

[15] S. Kumar and M. I. Nezhurina, "An ensemble classification approach for prediction of user's next location based on twitter data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 11, pp. 4503–4513, 2019.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[18] G. Aydin and I. R. Hallac, "Document classification using distributed machine learning," *arXiv preprint arXiv:1802.03597*, 2018.

[19] P. Semberecki and H. Maciejewski, "Distributed classification of text documents on apache spark platform," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2016, pp. 621–630.

[20] J. Xin, Z. Wang, C. Chen, L. Ding, G. Wang, and Y. Zhao, "Elm: distributed extreme learning machine with mapreduce," *World Wide Web*, vol. 17, no. 5, pp. 1189–1204, 2014.

[21] H. X. Huynh, V. T. Nguyen, N. Duong-Trung, V.-H. Pham, and C. T. Phan, "Distributed framework for automating opinion discretization from text corpora on facebook," *IEEE Access*, vol. 7, pp. 78 675–78 684, 2019.

[22] V. C. D. Hoang, D. Dinh, N. Le Nguyen, and H. Q. Ngo, "A comparative study on vietnamese text classification methods," in *2007 IEEE International Conference on Research, Innovation and Vision for the Future*. IEEE, 2007, pp. 267–273.

[23] V. T. Nguyen, N. T. Hai, N. H. Nghia, and T. D. Le, "A term weighting scheme approach for vietnamese text classification," in *International Conference on Future Data and Security Engineering*. Springer, 2015, pp. 46–53.

[24] H. T. Huynh, N. Duong-Trung, Q. D. Truong, and H. X. Huynh, "Vietnamese Text Classification with TextRank and Jaccard Similarity Coefficient," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 6, pp. 363–369, 2020.

[25] H. T. Huynh, N. Duong-Trung, X. S. Ha, N. Q. T. Tang, H. X. Huynh, and D. Q. Truong, "Automatic keywords-based classification of vietnamese texts," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2020, pp. 1–3.

[26] L.-D. Quach, N. Duong-Trung, A.-V. Vu, and C.-N. Nguyen, "Recommending the workflow of vietnamese sign language translation via a comparison of several classification algorithms," in *International Conference of the Pacific Association for Computational Linguistics*. Springer, 2019, pp. 134–141.

[27] C.-H. Chen, "Improved tfidf in big news retrieval: An empirical study," *Pattern Recognition Letters*, vol. 93, pp. 113–122, 2017.

[28] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, "Hot topic detection based on a refined tf-idf algorithm," *IEEE Access*, vol. 7, pp. 26 996–27 007, 2019.

[29] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec," *Information Sciences*, vol. 477, pp. 15–29, 2019.

[30] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "A brief survey of text mining: Classification, clustering and extraction techniques," *arXiv preprint arXiv:1707.02919*, 2017.

[31] W. M. Bolstad and J. M. Curran, *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.

[32] A. G. Shoro and T. R. Soomro, "Big data analysis: Apache spark perspective," *Global Journal of Computer Science and Technology*, 2015.

[33] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[34] I. A. T. Hashem, N. B. Anuar, A. Gani, I. Yaqoob, F. Xia, and S. U. Khan, "Mapreduce: Review and open challenges," *Scientometrics*, vol. 109, no. 1, pp. 389–422, 2016.

[35] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "Mllib: Machine learning in apache spark," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.

[36] R. Ignazio, *Mesos in action*. Simon and Schuster, 2016.

[37] "Facebook fplus - instagram zalo shopee - advertisement." [Online]. Available: https://plus24h.com/

[38] N. T. M. Huyen, A. Roussanaly, H. T. Vinh *et al.*, "A hybrid approach to word segmentation of vietnamese texts," in *International Conference on Language and Automata Theory and Applications*. Springer, 2008, pp. 240–249.

[39] Stopwords, "stopwords/vietnamese-stopwords," Apr 2017. [Online]. Available: https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords.txt

[40] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.