Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
n-Gram based language processing using Twitter dataset to identify COVID-19 patients

Nidal Nasser a,*, Lutful Karim b, Ahmed El Ouadrhiri c, Asmaa Ali d, Nargis Khan b

a Alfaisal University, Riyadh, Saudi Arabia
b Seneca College, Toronto, Ontario, Canada
c University of Houston, Texas, USA
d Queen’s University, Kingston, Ontario, Canada

ARTICLE INFO

Keywords:
Language categorization
Character n-gram
Word n-gram
TFIDF
LSVM
Natural language processing

ABSTRACT

Due to the rapid growth of electronic documents, e.g., tweets, blogs, Facebook posts, snaps in different languages that use the same writing script, language categorization, and processing have great importance. For instance, to identify COVID-19 positive patients or people’s emotions on COVID-19 pandemic from tweets written in 35 different languages faster and accurate, language categorization and processing of tweets is significantly essential. Among many language categorization and processing techniques, character and word n-gram based techniques are very popular and simple but very efficient for categorizing and processing both short and large documents. One of the fundamental problems of language processing is the efficient use of memory space in implementing a technique so that a vast collection of documents can be easily categorized and processed. In this paper, we introduce a framework that categorizes the language of tweets using n-gram based language categorization technique and further processes the tweets using the machine-learning approach, Linear Support Vector Machine (LSVM), that may be able to identify COVID-19 positive patients. We evaluate and compare the performance of the proposed framework in terms of language categorization accuracy, precision, recall, and F-measure over n-gram length. The proposed framework is scalable as many other applications that involve extracting features and classifying languages collected from social media, and different types of networks may use this framework. This proposed framework, also being a part of health monitoring and improvement, tends to achieve the goal of having a sustainable society.

1. Introduction

Twitter is a very popular micro-blogging and social networking application having approximately 350 million active users and 500 million tweets (or short messages) per day (Twitter, 2021). People tweet using 35 different languages to share emotions, events, and news. During the global COVID-19 pandemic that affected the people of the whole world, general people, health officials, governments used tweeters to share their emotions, general and personal information, health tips, government measures, and activities. Moreover, people often tweet if they are tested COVID-positive through Twitter. Hence, Twitter is a useful tool to identify COVID-positive patients that helps to monitor their health and responds appropriately for providing their treatment and preventing community spread. However, the number of tweets per second written in 35 different languages is huge.

Moreover, most existing researches (Ahmed, Ahmad, Rodrigues, Jeon, & Din, 2020; Alsaeedy & Chong, 2020; Chamola, Hassija, Gupta, & Guizani, 2020; Jelodar, Wang, Orji, & Huang, 2021; Li et al., 2020; Lyu et al., 2021; Rustam et al., 2020; Shorfuzzaman, Shamim Hossain, & Alhamid, 2021) on Twitter datasets only work on English datasets and do not consider other language datasets. These papers identify emotions, sentiments, or related information from social media using datasets in English. Even the recent works on COVID-19 that used the Twitter dataset considered datasets in English. However, people around the world use 35 languages to tweet. Hence, language categorization and processing are essential to identify the COVID-19 related posts in different languages worldwide. More specifically, identifying COVID-positive patients of different languages analyzing tweeter

* Corresponding author.
E-mail addresses: nnasser@alfaisal.edu (N. Nasser), lutful.karim@senecacollege.ca (L. Karim), ahmedelouadrhiri@gmail.com (A. El Ouadrhiri), ali@cs.queensu.ca (A. Ali), nargis.khan@senecacollege.ca (N. Khan).

https://doi.org/10.1016/j.scs.2021.103048
Received 4 October 2020; Received in revised form 28 April 2021; Accepted 23 May 2021
Available online 25 May 2021
2210-6707/© 2021 Elsevier Ltd. All rights reserved.
datasets can help governmental and health organizations monitor their health and achieve a sustainable society through health monitoring and improvements.

Various text categorization techniques have been developed to classify different types of documents. Character n-gram is one of many text categorization techniques, which is reliable, efficient, faster, and can tolerate textual errors. Character n-gram is a set of n-characters substrings of a string (Cavnar & Trenkle, 1994; Mansur, UzZaman, & Khan, 2021). For instance, n-grams for the string BEST is represented as follows.

\[
\begin{align*}
   n &= 2, \text{ Bi-gram: BE, ES, ST} \\
   n &= 3, \text{ Tri-gram: BES, EST}
\end{align*}
\]

N-gram based technique is best suited for text classification, especially for language categorization. \( n = 1 \)-grams represent the alphabet of a language and is very effective in categorization. Most writing style supports more than one language. For instance, all languages of the former Soviet Union use Cyrillic script. Hence, language classification is essential. One existing approach of language classification is to keep and check the lexicon of words. A language is classified as \( L \) when the lexicon of the testing document match most sample words of \( L \). However, it is very challenging to build lexicons, especially for inflected languages. In such a case, n-gram based language classification provides a better solution. This technique requires minimal storage and computation, works well for both long and short documents, and does not require content or semantic analysis of language documents. Moreover, the n-gram technique performs word-stemming automatically, whether other approaches require language understanding. For instance, “move”, “movement”, “moving” bears the same meaning in terms of language classification, and most of the n-grams generated for each word are common using the character n-gram technique (Mayfield & McNamara, 2003). Cavnar and Trenkle (1994) originally proposed and implemented the n-gram based text categorization technique. This technique is known as the Cavnar method, which uses Term Frequency (TF) for language categorization and hence, is known as the TF method. This technique is still prevalent for language categorization due to its ease of implementation and processing requirements. However, their implementation requires a large storage space that slows down the categorization of today’s huge collection of documents. Another popular approach to categorize languages is the traditional Term Frequency Inverse Document Frequency (TFIDF) method (Elberrichi & Alohar, 2007).

On the other hand, the word n-gram is the sequence of \( n \) words taken from any text document. For instance, the word n-grams for the sentence “coronavirus is contagious” is represented as follows.

\[
\begin{align*}
   n &= 1, \text{ unigrams: coronavirus, is, contagious} \\
   n &= 2, \text{ bigrams: coronavirus is, is contagious} \\
   n &= 3, \text{ trigrams: coronavirus is contagious}
\end{align*}
\]

Though character n-gram is significant for categorizing languages, word n-grams is more effective for feature/sentiment analysis. For instance, “COVID positive” is a bigram in tweets that could identify the person as COVID positive. In conjunction with Linear Support Vector Machine (LSVM) (Hassan, Gomaa, Khoriba, & Haggag, 2020), a very well-known machine learning approach, in this paper, we introduce a model that integrates character and word n-gram and can efficiently identify COVID positive patients. The contributions of this paper are as follows.

- Introduce a framework that integrates language categorization with feature extraction of multilingual Twitter datasets to identify COVID-19 positive patients from tweets.
- Introduce an approach for language categorization using the character n-gram based approach.
- Introduce an efficient approach for extracting features and building a classifier using word n-gram in conjunction with LSVM.
- Introduce a framework for health monitoring and improvement to achieve a sustainable environment and society.

Though the paper introduces language categorization and feature extraction approaches and integrates them in a framework, we implement the language categorization approach using the Cavnar TF method but efficiently using data structures in C++ and compare it with the TFIDF approach. The proposed framework

The proposed framework is scalable as it can also be used in many other applications that involve extracting features and classifying languages collected from social media and different types of networks.

The remaining part of this paper is organized as follows. Section 2 presents some existing research on natural language processing (e.g., language categorization and processing). Section 3 presents the working principles of the proposed framework, such as n-gram based text categorization and processing techniques and the TFIDF method. Section 4 presents the implementations of the Cavnar TF Method and TFIDF approaches in C++ using efficient data structures and some important observations while testing the experiments. In Section 5, the results of the experiments are presented with some analysis. Finally, Section 6 concludes this paper with some limitations and some future research directions.

2. Literature review

Extensive researches have been conducted on language and text processing using Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DP) approaches (Alsaedy & Chong, 2020; Cavnar & Trenkle, 1994; Chamola et al., 2020; Elberrichi & Alohar, 2007; Hassan et al., 2020; Iftim, Bakir, & Weikum, 2008; Jelodar et al., 2021; Li et al., 2020; Lu & Li, 2021; Mansur et al., 2021; Mayfield & McNamara, 2003; Rustam et al., 2020). However, only a few researches have been done on social media datasets for sentiment analysis, specifically social media datasets on the global pandemic, COVID-19. For instance, the work done in Rustam et al. (2020) predicts the number of infected cases, deaths, and recoveries using machine learning approaches, Linear Regression (LR), Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Machine (SVM), and Exponential Smoothing (ES) based on the dataset available by John Hopkins University and find the ES performs the best. The work done in Li et al. (2020) uses datasets collected from Weibo and NLP mechanisms to classify situational information into seven different types, such as help seek, emotional support, donations, cautions. This type of information helps governments, health personal, and individuals to respond appropriately. In Chamola et al. (2020), the authors present a comprehensive review on the importance of the Internet of Things (IoT), Unmanned Aerial Vehicles (UAVs), blockchain, Artificial Intelligence (AI), and 5 G to mitigate the impact of COVID-19 in the world economy, mental health and other areas. In Lyu, Chen, Wang, and Luo (2021), authors analyze 1.7 million tweets to identify two groups of users who use the non-controversial term “coronavirus” and controversial terms “Chinese virus” or “Wuhan virus”, and predict the number of users who are more likely to use controversial terms using machine learning techniques. In Jelodar et al. (2021), the NLP method based on topic modeling is used for automated extraction of COVID-19-related discussions from social media to uncover various issues related to COVID-19 from public opinions. They also investigate how to use the DL-based Long Short-Term Memory (LSTM) approach for sentiment classification of COVID-19 comments and find that this approach produces better results than other well-known ML approaches. The work done in Alsaedy and Chong (2020) introduce a new strategy to identify areas with high human density and mobility, which are at risk for spreading COVID-19 using existing cellular network functionalities — handover and cell (re) selection. The work done in Ahmed et al. (2020) introduces deep learning to impose social distancing to stop spreading COVID-19 and Shorfuzzaman et al. (2021) introduces video-surveillance-based smart cities to tackle COVID-19. However, most of these approaches considered datasets in English and did not consider datasets in other languages. To identify COVID related posts written in different languages, language
classifications are essential and then can extract features from a document written in a specific language, which is the main focus of this paper. Moreover, introducing health monitoring and improving people of different languages in the world through language categorization expect to build a sustainable society.

3. The proposed framework and approaches

This section presents the working principle of the approaches used in the proposed framework. Fig. 1 illustrates the sequence of operations or data processing to respond to any specific events, e.g., COVID-19, where we find language categorization is the first step. Fig. 2 demonstrates the flowchart of the language categorization and processing.

3.1. Character n-gram based method

N-gram based language categorization technique uses pre-existing language category samples. The process starts by generating n-gram frequency profiles for category samples and a test document. Fig. 3 presents the flowchart of the n-gram based language categorization technique (Cavnar & Trenkle, 1994), which is also described as follows.

3.1.1. Generate profile

Firstly, a scanner is used to preprocess language documents that keep necessary tokens and remove unnecessary information, e.g., punctuations, numbers, tags. Words are also transformed into lowercase. Then, n-grams are generated for n = 1–5 and stored. Frequency profiles of category and testing documents are generated by sorting n-grams in descending order of their number of occurrences. N-grams are also ranked based on the number of occurrences. An n-gram with the highest frequency has the highest rank that is represented by the lowest number (e.g., 0, 1, 2, and so on). N-gram frequency distribution over ranks results in Zipf’s graph (Ifrim et al., 2008), and it is imperative to identify the n-grams cutoff range in Zipf’s curve that results in the best categorization. Cavnar and Trenkle (1994) mention that the top 300 n-grams are highly correlated to language, and n = 1-grams have the highest ranks that represent the alphabet of a language. Hence, 1-grams are very effective for language classification. They used the top 300 n-grams for language classifications. However, they did not mention how to rank n-grams with the same frequencies. In this paper, we use the same rank for n-grams that have the same frequency.

3.1.2. Measuring and finding minimum distances

After generating frequency profiles of category and testing documents, the rank distances between each of the language sample documents and testing document T is computed as follows. Fig. 4 illustrates the process of measuring rank distances between T and sample documents.

The out of the place of an n-gram “A” of T is the absolute difference of the rank of “A” in T and a category document (that contains “A”). For instance, in Fig. 4, the n-gram “ING” is 0 rank out of place because it has the same rank in the frequency profile of T and category document. However, as the rank of “THE” is 5 in the category profile and 2 in T, “THE” is 3 ranks out of place. If n-gram “A” in T does not exist in the category profile, the rank out of place for “A” in the category profile is set to a maximum value that has not been mentioned in Cavnar and Trenkle (1994). In our paper, we set this value to the number of n-grams in T that is used for categorization.

Moreover, we set the maximum rank distance to approximately the mean of the number of n-grams of T. Then, the total distance between the frequency profile of T and each language category is computed by adding all “rank out of place” values. T is in the language category with which it has the minimum distance.

3.2. TFIDF method

n-Grams of each document are represented by vectors. To balance the n-gram profiles, we use “TFIDF” measurements (Elberrichi & Alohar, 2007; Hassan et al., 2020). To understand this model, we use the following notations.

\[ W_{ij} = tf_{ij} \times idf_j \]  \hspace{1cm} (1)

Where in Eq. (1)

- \( W_{ij} \) = Weight of n-gram j in document i
- \( tf_{ij} \) = Frequency of n-gram j in document i
- \( idf_j \) = Number of Language documents that contain n-gram j
- N = Total number of Language documents in the collections
- n = number of characters in the substring

Fig. 2. Proposed Framework for Language Categorization and COVID-Response.

Fig. 1. Sequence of Language Processing.
In Eq. (2), \( d \) is a language document to be categorized, and \( c \) is a category document. \( w_{jd} \) and \( w_{jc} \) represent the weight of \( n \)-gram \( j \) in \( d \) and \( c \), respectively. Document \( d \) is in a category with which \( d \) has the highest “Cosine” value (between 0 and 1).

3.3. Word n-gram based approach

Character n-gram seems efficient in language categorization whether word n-gram seems more effective for extracting emotion, features related to COVID-response such as tested COVID-positive, tested COVID-negative, and so on. However, before extracting features, the tweets have to preprocess only to keep keywords relevant to the proposed solution. This can be done by cleaning all non-letter characters, emoticons, username, tweet time, hyperlinks, punctuations (comma, period). Then other preprocessing approaches like stop word removal, stemming, and tokenization are performed. Stop word represents words such as a, an, the. The stemming is the process of removing suffixes from words, which have the same meaning. For example, “processed”, “process”, “processing” bear the same meaning, and removal of ed, ing would reduce the number of words using stemming. Once features are extracted from the tweets, the proposed framework uses a classifier for training and classification. This paper uses LSVM in this regard (Hassan et al., 2020).

4. Experiments

The following subsections present the datasets used and experiments detail observations while testing the experiments.

4.1. Datasets

In our experiment, we use a multi-language approach, i.e., we use Tweeter datasets in 7 languages: German, Dutch, Portuguese, Danish, Italian, Spanish, and French, collected from A large scale Covid-19 Twitter chatter dataset for open scientific research – an International collaboration (2021); Covid-19 Twitter Chatter Dataset for Scientific use (2021). All these languages use ASCII character representations. The size of these datasets is in the range of 2–4 KB. As a part of feature selections, we remove tags, punctuations, numbers, and other irrelevant information and only keep relevant information. Moreover, training and testing documents are normalized into lowercase for accuracy and consistency. Before preprocessing, we create a larger training document by appending 100–150 files in each language. After preprocessing, the size of the training sets and testing documents in each language are in the range of 120–200 KB and 500 Bytes – 4 KB, respectively. The testing documents are divided into three classes based on their size: (1) smaller than 1 KB (2) 1–2 KB and (3) larger than 2 KB. We limit the test to ±100, ±200, ±300, and ±400 most frequent n-grams in each of these classes. Let \( X < = 100 \) and \( Y > 100 \) represent the total number of n-grams up to rank \( r \) and \( r + 1 \) respectively where \( X = ±100 \) if \( 100-X < Y-100 \) otherwise, \( Y = ±100 \).

4.2. Implementations

To build the language classifier, we develop two sets of code: one for training the system with sample documents of each language, namely “Offline Processing” and another for testing language documents against the trained systems, namely ‘Online Processing’. The online processing consumes huge data bandwidth and incurs latency in transmission, whether the offline data processing or training reduces bandwidth requirements and data latency. Hence, the proposed framework does not consume too much bandwidth as it performs offline processing for training massive data. We present the detailed “offline” and “online” implementations in the following subsections.

4.2.1. Offline processing

In offline processing, we build the frequency profile of category samples. In this paper, we use Twitter 2 language datasets. The size of each language dataset is in the range of 3–5 KB. To make a larger dataset, we append about 100 datasets of each language into one training file and then preprocess the larger dataset using a scanner to remove all irrelevant information. We produce n-gram profiles for each of these language datasets using dynamic data structures “map” and “vector” in C++ since the number of n-grams in the dataset of each language and the number of language datasets that contain a specific n-gram cannot be determined a priori. When the datasets of each language are read, we store each n-gram in a “multi-map” so that n-grams are
sorted based on their frequency. Then n-grams of each language that are stored in a multi-map are added to a map that results in a single inverted index table for all language datasets. Each n-gram of the “map” is also connected to a vector in which each node contains a Language ID, n-gram frequency, and rank. We store the language ID and corresponding language name in another vector so that we can display the language name of the testing document after categorization. After processing the datasets of all training language documents, three output files are generated offline. The file “ngramProfile.txt” contains all n-grams that are stored in the map along with the number of languages in which they occur. Map store n-grams in the alphabetical order. The format of n-grams in “ngramProfile.txt” is as follows:

- `<total number of n-grams>`
- `<n-gram1> <language frequency1>`
- `<n-gram2> <language frequency2>`

Another file “languageofngrams.txt” stores the language ID, frequency and rank of each n-gram that are stored in the vector associated with the map in the following format:

- `<total number of entries>`
- `<langID1> <ngramFreq1> <rank1>`
- `<langID2> <ngramFreq2> <rank2>`

Total number of entries = \( \sum_{i=1}^{n} \text{nGramInLang}_i \), \( n \) = Number of Languages. The “listoflanguages.txt” file consist of language information according to the following format:

- `<TotalNumberOfLanguages>`
- `<langID1> <languageName1>`
- `<langID2> <languageName2>`

### 4.2.2. Online processing

All information that is stored in the files created in offline processing will also be stored into three vectors for the ease of online processing. A vector V1 stores each n-gram with its starting position in the “languageofngrams.txt” file. We simply add all number of languages that contain each n-gram (i.e., language frequency) to calculate the current starting position for each new n-gram in V1. If we require the language frequency of an n-gram e.g. “A” later, it can be calculated by subtracting the starting position of “A” from the starting position of the next consecutive n-gram. For instance, if the starting position and language frequency of “A” is 5 and 2, and the starting position of a newly added n-gram is \( 7 = 5 + 2 \). We find the language frequency of “A” by subtracting 5 from \( 7 = 7 - 5 = 2 \), if it is required later. The total number of n-grams is stored in a static variable. Another vector V2 stores the information of “languageofngrams.txt” file (i.e., language ID, n-gram Frequency, rank). Another vector V3 of type “structure” stores language ID and language name (from “listoflanguages.txt” file) along with variables for storing \( x, y, \) and \( xy \) weights (for TFIDF model) and distances (for Cavnar method). Similarly, the testing document is processed to generate its n-gram frequency profile. For each of the n-grams in the testing document, the corresponding n-gram in category profiles is searched using a binary search technique. If we find the n-gram, we measure distances using the Cavnar distance measure technique, based on rank, and vector space model, based on frequency, and compare the categorization performance.

### 4.2.3. Implementation differences

Our approach differs from the Cavnar method (Cavnar & Trenkle, 1994) in terms of the following key implementation aspects.

- In offline processing, we store training sets of all languages together in a single map whose values are linked to a vector. Hence, our implementation is efficient in terms of space requirements and search an n-gram since searching on a map is very fast.
- Cavnar method uses separate arrays for each language to store the frequency profiles of the training and testing documents considering the small-sized training and testing documents, which is not efficient.
- Since frequency profiles of the training set of each language are stored into different arrays, an n-gram appearing in different languages are stored in several arrays (not memory efficient). In our implementation, the map eliminates storing duplicate n-grams and hence, requires less memory as compared to the technique used in Cavnar and Trenkle (1994). For small datasets, memory is not so crucial. However, for large datasets, our implementation should outperform in terms of processing speed and memory requirements.

### 4.3. Observations

Some observations while testing the experiments with a few small-sized categories and testing documents are presented as follows.

Cavnar measure is dependent on the ordering of n-grams according to ranks and also the existence of the testing n-grams in a category document. For instance, if most n-grams of a testing document \( Z \) exist in a category document \( C_1 \) but a few n-grams of \( Z \) (with higher ranks) do not exist in \( C_1 \), the distance of \( C_1 \) will be larger than the category document \( C_2 \), which has more n-grams with higher ranks than \( C_1 \) that exist in \( Z \). In this case, the size of a category and testing document also has a significant impact on the categorization. During our initial test, we identify an interesting scenario of the TFIDF approach. We mainly train the system with 7 language documents of size in the range of 150–200 KB. Using \( \pm 100 \) n-gram measure, TFIDF could not categorize one Dutch testing document. After careful observation, we identify that \( \pm 100 = 45 \) n-grams that is, \( \pm 100 \) n-grams measure could look for only 45 n-grams of the testing document in the training set. Moreover, the document frequency of each of these n-grams was equal to the total number of language documents in the training set. Hence, the Inverse Document Frequency (IDF) of all n-grams was equal to 0, which results in zero weights and an undefined 0/0 similarity value. This is a limitation of the TFIDF method.

#### 4.3.1. Word n-gram and TFIDF Approaches

Once we can categorize languages, we only work on the tweets datasets in English and implement the framework to extract features to identify people’s responses on COVID-19, such as COVID positive or negative as part of the proposed framework. We use the machine learning approach LSVM as it is a very well-known and efficient approach as a classifier and compare the performance of LSVM using the Cavnar method and TFIDF weighting method.

![Fig. 5. Ratio of Incorrect to Total Classifications using ± n-gram Measure and Document Length < = 1KB.](image-url)
5. Results and analysis

Figs. 5–7 present language categorization results when the maximum rank distance for the Cavnar TF method is set to the \( n \)-gram length (e.g., ±100, ±200, 200, and so on). We use both ±200 and 200 just to identify whether ± is a good categorization measure rather than using a fixed number (e.g., 200) of \( n \)-grams. Some of the results show substantial improvements in the vector space model (TFIDF) using a fixed number of \( n \)-grams. Hence, we would use a fixed number of \( n \)-grams in our further experiments. Figs. 8 and 9 demonstrates results using 100, 200, 300, and 400 \( n \)-gram lengths. Moreover, this \( n \)-gram length is used as the maximum rank distance that is required to measure distances in the Cavnar TF method, when an \( n \)-gram of a testing document is not found in the training set. Figs. 10 and 11 present categorization results when the maximum \( n \)-gram distance is set to a fixed value (e.g., 250 in our experiment) around the mean of all \( n \)-gram lengths.

Figs. 12 and 13 show the accuracy of language categorization (in percentage) for smaller articles (<1 KB). Similarly, we can show the accuracy of categorization for larger articles. Other performance metrics to evaluate the approaches are as follows.

**Precession** - is the ratio of true positive to total positive identified (including true positive and false positive, where false positive is supposed to be negative)

**Recall** – is the ratio of true positive to actual positive. Actual positive includes true positive and false negative, where false negative is supposed to be true positive.

**Accuracy** – is the ratio of true positive and negative to total tweets identified as different types.

\[
F\text{- measure} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}
\] (3)

We estimate these performance metrics based on our experimental results presented above and find results as presented below in Table 1.

5.1. Discussion and analysis

The actual numbers of \( n \)-grams in the testing documents that are used for matching with those of the training documents are in the range of 50–250 while using ±200 \( n \)-gram measure. For instance, ±200 \( n \)-grams measures use 60–70 \( n \)-grams of Spanish articles (of size <1KB), and this results in the most incorrect categorization for Danish articles using the TFIDF method. It is also observed that IDFs of the most top-ranked \( n \)-grams are 0, and hence, only a few non-zero IDFs influence the categorization. If the \( n \)-grams with non-zero IDFs have a slightly more frequency in a language document other than the actual category, a
mismatch occurs. Cavnar method is advantageous in such a scenario because \( n \)-grams in the actual category are expected to have higher ranks (lower number) for the top-ranked \( n \)-grams (even if only the top 60 \( n \)-grams are considered) in the testing document, which results in a minimum distance.

In all cases, we find the Cavnar method (Cavnar & Trenkle, 1994) outperforms the TFIDF. A fixed number of \( n \)-grams provide a better result than \( \pm \)\( n \)-grams measure. Moreover, the plus \((+)^{n}\)\( n \)-gram has a better categorization result than the minus \((-)^{n}\)\( n \)-gram in the TFIDF method. This is because \((+)^{n}\) has more \( n \)-grams than \((-)^{n}\) measure that is required to distinguish languages based on weights when top-ranked \( n \)-grams appear in several languages. From the results presented in Figs. 8 and 10, we observe that the Cavnar method performs better when the maximum rank distance is set to a fixed number around the mean rather than assigning it to the \( n \)-gram length.

We observe that, for most cases, the percentage of correct results is low for smaller testing documents, whether it is high for larger testing documents. In other words, larger testing documents and \( n \)-grams length provide better categorization results as compared to smaller testing documents and \( n \)-gram length. However, some results have anomalies, where increasing \( n \)-gram length generates more inaccurate results. One possible reason is that a few language documents might contain words/passage of other languages. If the maximum rank distance increases Cavnar method provides more accurate results than the TFIDF method. For instance, when \( n \)-gram length = 100, the Cavnar method provides more accurate results for the maximum rank distance = 250 (Fig. 10) than the maximum rank distance = 100 (Fig. 8). This is because the \( n \)-grams boundary in training set increases if the maximum rank distance increases in the Cavnar method.

6. Conclusion and future work

Character \( n \)-gram based language categorization technique proposed by Cavnar and Trenkle (1994) is very simple but efficient in terms of memory and processing requirements. Moreover, this technique tolerates textual errors and works well for both short and large documents. In addition to their proposed technique, we implement the traditional TFIDF method in this paper. Our implementation eliminates storing
duplicate n-grams of different language datasets and hence, is more efficient in terms of memory and speed as compared to existing methods. Initially, we test our experiment using n:n-grams and fixed n-gram measures and find a fixed n-gram length generate more accurate results. Later, we use a fixed n-gram length to test our experiments. For the Cavnar method, we set the maximum rank distance to the n-gram length and also a number $X$, around the mean of all n-gram lengths used in the experiments. Cavnar method generates more accurate results when the maximum rank distance was set to $X$. Except for a few anomalies, the Cavnar method has much better performance than the TFIDF method. We also observe that the classification accuracy is directly proportional to the size or n-gram length of testing documents from experiment results. In this paper, we use languages that are easy to categorize. Categorizing similar languages, e.g., Australian English, British English, Canadian English, etc., is a challenge using the character n-gram based technique. Moreover, the n-gram based technique cannot be used in a particular text categorization, which requires the understanding of contents, e.g., movie review, opinion mining, user classifications in social networks etc. As an extension of this paper work, we would work and compare with other text categorization techniques, e.g., fast logistic regression with variable length n-grams (Ifrim et al., 2008) that can classify different types of texts. We classified languages and only worked on the tweets written in English for identifying the responses as COVID positive or negative as part of this research. In the future, we plan to build a model that can identify responses in all 35 languages the Twitter use. Moreover, we plan to add a middle layer in the proposed framework that would enhance the training and learning process more effectively so that the outcome will have less error. We also plan to perform an emotional analysis of COVID-19 patients to come up with a solution to mitigate mental health crises during this unprecedented time. Hence, we would be able to build a sustainable society through health monitoring and improvements.

Author disclosure

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgement

Authors would like to thank Alfaisal University for funding and supporting this work through the Internal Research Grant C20220.

References

A large scale Covid-19 Twitter chatter dataset for open scientific research – an International collaboration, accessed from https://zenodo.org/record/3819464.

Ahmed, I., Ahmad, M., Rodrigues, J. J. P. C., Jeon, G., & Din, S. (2020). A deep learning-based social distance monitoring framework for COVID-19. Sustainable Cities and Society, 65, Article 102571. https://doi.org/10.1016/j.scs.2020.102571. ISSN 2210-6707.

Alsaeeed, A. A. R., & Chong, E. K. P. (2020). Detecting regions at risk for spreading COVID-19 using existing cellular wireless network functionalities. IEEE Open Journal of Engineering in Medicine and Biology, 1, 187–189. https://doi.org/10.1109/OJEMB.2020.3002447

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. Proceedings of SDAR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 161–175.

Chamola, V., Hassija, V., Gupta, V., & Guzani, M. (2020). A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact. IEEE Access, 8, 90225–90265. https://doi.org/10.1109/ACCESS.2020.2992341

Covid-19 Twitter Chatter Dataset for Scientific use, accessed from www.panacealab.org/covid19https://github.com/theadnacealab/covid_19_twitter.

Elberrichi, Z., & Alohar, B. (2007). N-grams in texts categorization. Scientific Journal of King Faisal University, 8(2), 25–38.

Hassan, N., Gomaa, W., Khoriba, G., & Haggag, M. (2020). Credibility detection in twitter using word n-gram analysis and supervised machine learning techniques. International Journal of Intelligent Engineering and Systems, 13(February (1)), 291–300. Ifrim, G., Bakir, G., & Weikum, G. (2008). Fast logistic regression for text categorization with variable-length n-grams. In KDD ’08: Proceeding of the 14th ACM SIGKDD International conference on Knowledge discovery and data mining (pp. 354–362).

Jelodar, H., Wang, Y., Orji, R., & Huang, H. Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. IEEE Journal of Biomedical and Health Informatics, doi: https://doi.org/10.1109/JBHI.2020.3001216.

Lü, L., et al. (2020). Characterizing the propagation of situational information in social media during COVID-19 epidemic: A case study on Weibo. IEEE Transactions on Computational Social Systems, 7(April (2)), 556–562. https://doi.org/10.1109/TCSS.2020.2980007

Lyu, H., Chen, L., Wang, Y., & Luo, J. Sense and sensibility: Characterizing social media users regarding the use of controversial terms for COVID-19. IEEE Transactions on Big Data, doi: https://doi.org/10.1109/TBDATA.2020.2996401.

Mansur, M., UzZaman, N., & Khan, M. Analysis of N-gram based text categorization for Bangla in a newspaper corpus. Dhaka, Bangladesh: Center for Research on Bangla Language Processing, BRAC University.

Mayfield, J., & McNamee, P. (2003). Single n-gram stemming. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Rustam, F., et al. (2020). COVID-19 future forecasting using supervised machine learning models. IEEE Access, 8, 101489–101499. https://doi.org/10.1109/ACCESS.2020.2997311

Shorofuzzaman, M., Shamim Hossain, M., & Alhamid, M. F. (2021). Towards the sustainable development of smart cities through mass video surveillance: A response to the COVID-19 pandemic. Sustainable Cities and Society, 64, Article 102582. https://doi.org/10.1016/j.scs.2020.102582. ISSN 2210-6707.

Twitter, access from https://about.twitter.com.