Multimedia Tools and Applications (2021) 80:13475–13490
https://doi.org/10.1007/s11042-020-10314-9

Generic framework for multilingual short text categorization using convolutional neural network

Liriam Enamoto[1] · Li Weigang[1] · Geraldo P. Rocha Filho[1]

Received: 21 November 2019 / Revised: 26 August 2020 / Accepted: 22 December 2020 /
Published online: 15 January 2021

Abstract
Online social media is a powerful source of information that can influence users' decisions. Due to the huge volume of data generated by such media, many researches have been done to automate text categorization. However, finding useful information to satisfy user's needs is not an easy task. There are many challenges to overcome especially in short text categorization that in addition to being a time-consuming and costly process, short messages have misspellings, typos, irony words and lack of context. To solve these challenges, this article proposes GM-ShorT, a Generic framework for Multilingual Short Text Categorization based on Convolutional Neural Network (CNN). For this, GM-ShorT collects online social media data. Such data were used as input to CNN that is combined with a word embedding mechanism to categorize short text messages. We explored several architectures for CNN and show that GM-ShorT can be used in multilingual Short text categorization with an accuracy of 13.58% higher when compared to other classical approaches.

Keywords Convolutional neural network · Text categorization · Multilingual text · Social media

1 Introduction

In recent years, a massive amount of data have been produced in online social media and motivated by this continuously growing volume of text data, many researches have been done to automate text categorization [2, 6, 7]. There are still challenges to overcome in this field such as useful information extraction from noisy data, misspellings and typos, abbreviation, irony words, negation and lack of context [15, 20, 23]. Regardless of these

✉ Liriam Enamoto
liriam.enamoto@gmail.com

Li Weigang
weigang@unb.br

Geraldo P. Rocha Filho
geraldof@unb.br

[1] Department of Computer Science, University of Brasília, Brasília, DF, Brazil

challenges, online social media is a powerful source of information that can influence users' decisions and can help people daily.

People use social media to communicate quickly, as well as mobilize and organize populations to achieve various objectives, such as get relevant and timely information [24], influence people's opinions about a specific topic and get support and advice from online users by sharing daily problems. The most common use of social media is to update users with real-time information which might not be available through official channels in a timely way [16]. For instance, to share real-time information about possible tsunami after earthquake [18], hospitals where the H1N1 influenza vaccine is available [12]. On the other hand, during emergencies such as natural disasters and epidemics, the amount of information posted on social media exceeds the capacity of the public to consume it. Furthermore, crisis and emergency events that occur in one country might cross borders and continents affecting people who use different languages. Therefore, the use of social media as communication is one of the fundamental tools for emergency management [20].

The text categorization activity allied with deep learning is one of the most promising pathways to provide solutions related to emergency management via social media. Text categorization is the activity of labeling natural language text with thematic categories from a predefined set and can be applied in document indexing, document filtering and any application requiring document organization [19]. Machine learning algorithms such as Support Vector Machine (SVM), Naïve Bayes and Neural Network have made a great advance in extracting and classifying text.

In the literature, several studies address deep learning algorithms in text categorization, such as Convolutional Neural Network (CNN). However, some studies create a large annotated corpus which is a time consuming and costly process [29]. The CNN models used in these studies are complex and hard to effectively conduct hyperparameters fine-tuning and therefore, they are not easily adaptable to different domains [28]. Also, most researches apply machine learning methods on English text categorization and few studies on multilingual text categorization have been done [9, 22] due to the complexity of the issue.

This article aims to explore this gap presenting a **G**eneric framework for **M**ultilingual **Shor**t **T**ext Categorization (GM-ShorT) with the following contributions:

1. Using GM-ShorT, the CNN model can be trained with a small dataset;
2. A decision module composed by a shallow CNN model with 2 layers that can be easily adapted to new contexts;
3. The mechanism of GM-ShorT multilingual text categorization covers alphabetical (i.e., English and Portuguese) and non-alphabetical (i.e., Japanese) languages.

The remainder of this article is structured as follows. Section 2 describes the related works, discussing the challenges and gaps for this research. Section 3 describes the development of the generic framework for multilingual short text categorization. Section 4 shows how the proposed framework was validated. Finally, Section 5 presents the conclusion and indicates future works.

2 Related works

This section presents related works that use CNN to solve the short text categorization problem, as well as some challenges in this area. Originally used for computer vision, CNN is effective for natural language processing achieving better results than traditional machine learning algorithms in text categorization. The shallow CNN model proposed by

Kim [7] composed by one convolution layer and one pooling layer built on top of pre-trained Word2Vec performed better than SVM or even more sophisticated deep learning model with complex pooling schemes. Some variations of word vector were tested including pre-trained Word2Vec with static and non-static word vector. The experiments suggest that the choice of non-static word vector gives better results in most of NLP tasks. The model proposed by [7], besides not including multilingual text classification, differs from this work by not training the weights using language-specific Word2Vec.

The experiment of Johnson et al. [6] used parallel CNN to perform sentiment analysis and topic classification. The input text was converted into a one-hot vector and loaded into two convolution layers in parallel. The results suggest that with the parallel CNN model, several types of embedding can be learned and combined complementing each other for higher accuracy. Furthermore, Johnson et al. [6] argue that the strength of CNN is that n-grams (or region of n words) can contribute to accurate prediction even if they did not appear in the training data, as long as their constituent words did, because input of embedding is the constituent words of the region. For example, the model trained to assign large value to the words *"I love"* and small value to *"I hate"* is likely to assign a large value to *"We love"* and a small value to *"We hate"* as well, even though *"We love"* was never seen during training. In our solution, we used word embedding to represent text instead of using region vector. According to Sosa et al. [21], adding a convolution layer to the neural network would increase the overall accuracy by 2.8% for Twitter sentiment analysis. However, Lu et al. [10] suggest that for small datasets CNN model can underperform SVM due to bad generalization ability and lack of robustness reaching 99% of training accuracy rate after 15 epochs. Because of this, we used low values for the learning rate to assure a slow and gradual learning improvement of the model and avoid overfitting.

Other researchers use deep learning techniques to explore valuable information posted on social media in disaster scenarios [2, 15]. Caragea et al. [2] modeled CNN for detecting useful information posted on Twitter during crisis events. In [2], Twitter data from the CrisisLex project was used and CNN produced better results than SVM and other Neural Network models. In our solution, we used the Twitter data from CrisisLex for comparison as presented in Section 4.2, getting better results.

The previously presented studies applied CNN to words but CNN can also be applied to characters. The study of Zhang et al. [29] suggested that character-level CNN can be used for text categorization with promising results. They demonstrated in the experiments that when CNN based on a character level is trained on large-scale datasets, the model does not require the knowledge of words. This means that the model also doesn't need the knowledge of the syntactic or semantic structure of a language. Furthermore, empirical study suggests that character level encoding produces better results in CNN text categorization than word level for non-alphabetic languages such as Japanese and Chinese [27]. Zhang et al. [29] created 7 large datasets with sizes ranging from 120.000 to 3 million to conduct the character level experiment, but in our solution, we used a very small dataset.

In the literature, there are a few studies in the field of multilingual short text categorization derived from different language families. The main advantage of our solution compared with the existing researches in the literature is that we use a small training set on shallow CNN model using language-specific Word2Vec and adapted to character level approach for non-alphabetic languages and word-level for alphabetic languages. The key aspects that differentiate these researches are: (i) the word representation model, i.e, word embedding [7, 15] or region vector [6]; (ii) the depth and CNN architecture, i.e, shallow single CNN [7], deep parallel CNN [6], ensemble CNN RNN model [21]; (iii) word level approach [2, 6, 7, 10, 15, 21] or character level approach [27, 29].

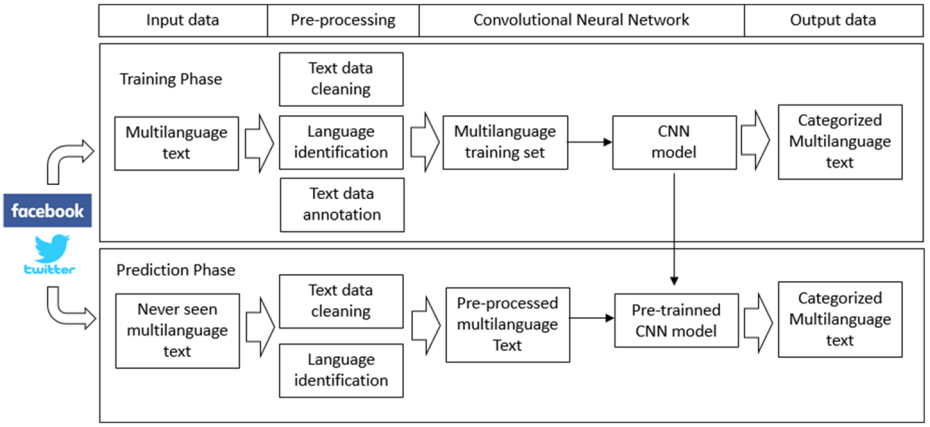


Fig. 1 GM-ShorT execution flow

3 GM-ShorT

This section describes GM-ShorT, a **G**eneric framework for **M**ultilingual **Short** **T**ext Categorization based on CNN. To facilitate the understanding of how GM-ShorT was modeled, we describe below the overview of the proposed framework, as well as dataset and CNN used. Finally, the applicability of GM-ShorT using social media is discussed.

3.1 Generic framework for multilingual short text

Figure 1 presents an overview of GM-ShorT structure. In GM-ShorT, user posts are collected from online social media such as Twitter and Facebook using a keyword, for instance, "ebola". In the training phase, the pre-processing task is divided in three steps: (i) text data cleaning; (ii) language identification; (iii) text data annotation. First, the text data is cleaned by removing emoticon, emoji, URL, and words that start with '@' usually representing username. All other tokens such as abbreviations, slangs, misspelled words, and numbers were considered to compose the dataset. This cleaning task enables a better result in the following steps. In the second step, the language identification procedure is performed to separate the user posts into English, Japanese and Portuguese datasets. User posts in different languages than those before mentioned are discarded. In the third step of the pre-processing task, each dataset is manually annotated into five categories described in Table 1. These three steps generate the multilanguage training set. Once the text is cleaned, the language is identified and labeled accordingly, the multilanguage training set is processed by the CNN model which generates the final categorized multilanguage text. Deep learning models such as CNN are usually trained over large datasets to avoid overfitting and to obtain better generalization. However, create such large annotated data is a time-consuming, expensive and challenging task, especially with multilingual texts. In this framework, we used a small-labeled dataset in English, Portuguese and Japanese to train the CNN model. In the prediction phase, never seen text data collected from social media can be loaded into the GM-ShorT. After performing the pre-processing steps, the new data can be loaded into a pre-trained CNN model and get the text categorization prediction.

Table 1 Twitter posts classification criteria

Category	Description
Outbreak situation report	Reports about newly confirmed Ebola cases and deaths, official announcement about Ebola free countries.
Informative posts	Hospitals prepared for Ebola patients, vaccine researches, educational information on virus prevention, donation campaign.
Negative impact	Social and economic impact caused by the outbreak.
Negative information	Criticism against the government, panic, racism.
Need for preparedness	Lack of hospitals, body bags, food, and safety funeral protocols.

In order to perform online social media short text categorization using CNN, the following tools were used in the proposed GM-ShorT: (i) NodeXL[1] was used to collect posts from Twitter; (ii) Python, Pandas, and Numpy were used to transform and manipulate the datasets; (iii) Keras[2] with Tensorflow backend was used to create the vocabulary, and the layers of CNN model; (iv) Scikit-learn[3] was used to k-fold cross-validation and grid search; and (v) Gensim[4] to process Word2Vec.

3.2 Mechanism to categorize short text messages on GM-ShorT

To perform the process of multilingual short text categorization in GM-ShorT, it is necessary to collect online social media data, process and categorize them. This task is not trivial since it is necessary to treat qualitative data to perform the categorization. In our framework, the short text categorization was implemented using CNN. CNN consists of a sequence of one or multiple pairs of convolution and pooling layers. A convolution layer is composed by several computational units, each of which takes as input a region vector that represents a small region of the input image, and the small regions collectively cover the entire data [6]. A computational unit associated with the *l-th* region of input x calculates the (1): $r_l(x)$ is the input region vector that represents the *l-th* region, W represents the weight matrix, b the bias, and σ represents a nonlinear activation function such as Rectified Linear Units (ReLU).

$$\sigma(W.r_l(x) + b) \tag{1}$$

The matrix of weights W and the vector of biases b are learned through training, and they are shared by computational units in the same layer [6]. The output image of the convolution layer is passed to a pooling layer, which shrinks each region of the image into one unit by computing the average or maximum value of each region [6]. The idea of pooling layer is to capture the most important feature of each region. These features compose the last pooling layer and are passed to a fully connected layer, which returns a prediction based on features learned internally by previous layers [2].

In CNN for text, each sentence of input data is transformed into a matrix of word embedding [25]. Word embedding is a distributed representation of words that reduce data sparsity problem [1] and can be trained as part of CNN training or adopt pre-trained corpus such

[1] https://www.smrfoundation.org/nodexl
[2] https://keras.io/
[3] https://scikit-learn.org
[4] https://pypi.org/project/gensim/

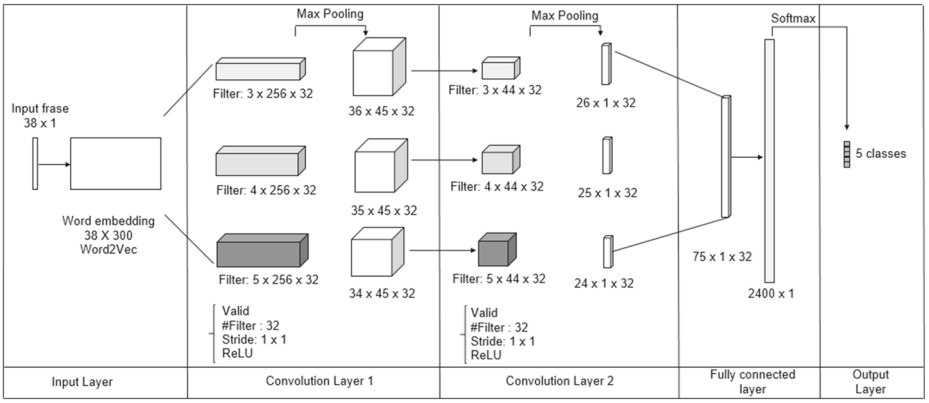


Fig. 2 Example of the operation of the CNN 2L architecture implemented in GM-ShorT

as Word2Vec [13]. Each convolution layer has a variable number of computational units, with each unit corresponding to a small region (one or more words) from the input text [2]. Similarly to CNN for image, CNN model for text can be composed by one or multiple pairs of convolution and pooling layers followed by fully connected layer which returns the prediction for the input text.

In the application of GM-ShorT for short text, we modeled three different CNN architectures. The first architecture (CNN 1L) is composed of one convolution layer and one pooling layer based on the CNN model of Kim [7]. The second architecture (CNN 2L) is composed of two convolution layers and two pooling layers as illustrated in Fig. 2. Similarly, the third architecture (CNN 3L) is composed of three convolution layers and three pooling layers. The CNN 2L architecture presented better results as described in Section 4.2 and was deployed on GM-ShorT.

3.3 Modeling the dataset to categorize short text messages

The dataset modeled[5] in this research was collected from Twitter during the 2014/2015 Ebola outbreak using the keyword "ebola". From a total of one million tweets downloaded during 6 months, 1162 tweets in English, 246 tweets in Portuguese and 157 tweets in Japanese were manually annotated into five categories following the rules described in Table 1. Text classification using traditional models, such as SVM, Naive Bayes, and others, frequently uses document indexing techniques such as a bag of words to document term representation and Term Frequency–Inverse Document Frequency (TFiDF) to compute the weights of each term [19]. These techniques usually perform better when the document is stemmed and the dimensionality is reduced, however, a bag of words does not preserve the word order and bag of n-grams could result in high data sparsity for large values. CNN for text classification make use of the internal structure of data and internally learn features that are useful for classification [2]. In this experiment, stemming was applied only to the English dataset and no dimensionality reduction was applied to any of the datasets.

[5] Available https://github.com/enmili/multilingualDataset

Table 2 Twitter datasets

Dataset	#Tweets	Examples	Translation
EN-T	1162	Ebola death toll tops 10000.	–
PT-T	246	Ebola mata mais de 10 mil pessoas	Ebola kills more than 10000 people.
JP-T	157	死者８０００人超に＝西アフリカの #エボラ熱 #WHO	More than 8000 deaths in West Africa #Ebola #WHO

Table 2 details each dataset used in this work. The column Datasets represents the multilingual datasets: EN-T contains English tweets, PT-T contains Portuguese tweets and JP-T contains Japanese tweets. The column #Tweets shows the number of tweets, the column Examples shows Twitter posts about the outbreak, and the column Translation shows the English translation of Portuguese and Japanese tweets. One of the difficulties found during this research is the lack of public available English, Japanese and Portuguese datasets related to a single subject. Even representing a small dataset, the annotated Twitter posts used in this work are public available for further multilingual text classification researches.

3.4 Word embedding mechanism

To perform the short text categorization, each word needs to be represented as a numeric value. The basic way of converting words into numbers might be assigning each word to a one-hot vector. This vector can be zero-padded except for those unique indexes corresponding to each word. However, representing words using a one-hot vector can lead to data sparsity problem [1, 11]. Word embedding is a distributed representation of words and alleviates the data sparsity problem. By mapping, semantically similar words to nearby points by cosine similarity word embedding can capture meaningful syntactic and semantic regularities between words [8, 13].

In this research, the word embedding was learned using publicly available language-specific pre-trained Word-2Vec. The details of each Word2Vec are described in Table 3. English Word2Vec was obtained from the Google News dataset trained on 100 billion words. Each word is represented by a 300-dimension vector. Portuguese Word2Vec was obtained from Hartman et al. [5] research trained on large Portuguese corpus including Google News, Wikipedia, Brazilian science divulgation text by FAPESP, news crawled from G1 news portal in 2014/2015 and other 13 corpus using 50-dimension vector. Japanese Word2Vec was obtained from the publicly available repository trained on Wikipedia. Each Japanese character is represented by a 50-dimension vector.

For the English dataset word-level approach was used resulting in a vocabulary size of 2509 words and a maximum tweet length of 38 words. Each word of one tweet was

Table 3 Word2Vec description

Language	Model Name	Dimension	Vocabulary size
English	GoogleNews-vectors-negative300	300	3,000,000
Portuguese	CBOW 50	50	929,606
Japanese	Japanese Word2Vec	50	1,046,708

tokenized and encoded into numbers starting from 1 to 2509, where 2509 represents the vocabulary size. In the next step, this vector was zero-padded to have the maximum tweet length, e.g 38 words resulting in a 38×1 dimension word vector. After this tokenization process, 2509×300 embedding matrix was randomly initialized between -0.25 and $+0.25$, in which 2509 represents the vocabulary size and 300 represents the dimension of Word2Vec. In the next step, English Word2Vec was loaded into the embedding matrix with pre-trained GoogleNews-vector values. 1994 words out of 2509 words were found in GoogleNews-vector representing 79,4% of matching. The most common words not found in Word2Vec are misspelling words, abbreviations or concatenated words used in hashtags, such as "wecanbeatebola", "westafrica" and "evd". In the following step, the 38×1 dimension word vector was transformed into a word embedding matrix resulting in a 38×300 dimension matrix filled by GoogleNews-vector values.

For the Portuguese dataset, the same word-level approach was used with a vocabulary size of 1222 words and a maximum tweet length of 32 words. 842 words out of 1222 words were found in the CBOW 50 vector representing 68,9% of matching. The most common words not found in Word2Vec are news media site names, such as "g1", "jornaloglobo" and "bbcbrasil".

Japanese text usually does not have space separation between words as shown in Table 2 examples. Unlike alphabetic languages such as English, there is no clear word boundary for Chinese, Japanese and Korean texts making difficult to apply language processing methods that assume word as the basic construct. Therefore, to produce better results, the character level approach was used for the Japanese dataset as suggested by Zhang et al. [27]. The vocabulary was built with 769 characters and a maximum tweet length of 146 characters. In the case of the Japanese dataset, 755 characters out of 769 characters were found in the Japanese Word2Vec vector representing 98.17% of matching. The common characters not found in Word2Vec are double-byte special characters, such as "○".

User comments on social media may contain a variety of words, including words not found in Word2Vec. Slangs, abbreviations, and misspelled words follow the same process. First, they are transformed into numbers via word embedding. In this process, tokens such as slangs not found in the pre-trained Word2Vec are randomly initialized. After the embedding process, the convolution layer and max pooling layer help to learn all features values, including slangs, and the model is optimized via backpropagation [7].

3.5 Proposal's applicability

The information posted on social media during crises varies significantly in data quality [15]. Most messages are noisy data and may contain information not related to crisis response. Finding useful words from massive noisy data can be challenging for social media users who need quick answers to their needs during emergencies. Additionally, in a real emergency, it is not realistic to create a large annotated corpus to train the CNN model because it is a very time-consuming task. GM-ShorT presented in this paper can be used to categorize social media posts using a small dataset, and it is adapted for multilanguage classification. The information can be classified according to the categories related to each emergency. The categorized messages can be used for various purposes such as situation report, inform hospitals prepared for the emergency, donation campaign, emotion recognition, and consensus mechanisms in smart environments [4, 14, 17].

GM-ShorT can be used in different domains of context other than an emergency, for instance, in fake news detection. The exponential growth of data produced by online social media users has brought some threats [3]. One of the threats is fake news written in an

Table 4 Hyperparameters details for CNN 2L

Data	Vocab. size	Param.	Emb. size	Filter size	Filter	Batch size	Learn. rate	Drop-out	L2
EN-T	2509	1,403,873	300	3/4/5	32	128	0.001	0.5	0.2
PT-T	1222	470,545	50	3/4/5	32	30	0.001	0.5	0.2
JP-T	769	1,708,965	128	3/4/5	32	32	0.001	0.4	0.4

intentional and unverifiable language to mislead the reader [26]. Despite having some challenges to overcome, such as the lack of publicly available fake news dataset [26] and the difficulty to label data manually, GM-ShorT can be used in fake news detection. This can be done by adapting the model to binary-class, in which the input text should be classified in one of the two possible classes [19], i.e., "Fake" or "Not Fake". To this end, the output layer should be changed from 5 to 2 classes, the loss function should be set to binary cross-entropy, and the model should be trained with an annotated fake news dataset.

4 Performance evaluation and methodology

This section presents the results of CNN 2L model and compares the results with CNN 3L and two baseline models: CNN 1L and SVM. GM-ShorT is validated using English, Portuguese and Japanese datasets. In addition, the English dataset used in Caragea et al. [2] research was processed into CNN 2L model for evaluation purposes. K-Fold Cross-Validation was used in all experiments where k was fixed to 10, being 9 for training and the remainder for testing.

4.1 Hyperparameters

The CNN 2L model architecture is the same for English, Portuguese and Japanese datasets as illustrated in Fig. 2. Grid search was used as hyperparameter optimization technique in which the following configurations were tested: Dropout rate = {0.4, 0.5, 0.7}, learning rate = {0.001, 0.005} and L2 lambda = {0.2, 0.4}. The best combination of hyperparameters for each dataset is shown in Table 4. For English and Portuguese datasets, the dimensionality of input embedding was adjusted to the Word2Vec dimension. For Japanese data-set using the Word2Vec dimension of 50 was giving low accuracy and was adjusted to 128. For all datasets filter windows of {3, 4, 5} with 32 filters each was used, so that filters slide over 3, 4, and 5 words with no padding and stride set to 1. Rectified Linear Units was applied as activation function from convolution layer to pooling layer and Adam optimizer with learning rate 0.001. The batch size was set to 128 for English, 30 for Portuguese and 32 for the Japanese dataset. For regularization purpose, dropout rate 0.5 and L2 lambda 0.2 was set to English and Portuguese datasets. For the Japanese dataset dropout rate 0.4 and L2 lambda 0.4 gave the best result. Max pooling was used in the pooling layer to get the maximum value as the most important feature corresponding to each filter. Then the pooling layer vectors were concatenated into a fully connected layer in which softmax function was applied to finally categorize each input into one of five classes described in Table 1. All three datasets were trained over 50 epochs.

4.2 Impact of CNN models in GM-ShorT

First, we compared the results between the three CNN models: CNN1L, CNN2L, and CNN3L. In addition, we tested each model with and without Word2Vec. Table 5 shows the results of the experiment where the rows contain the different datasets and the columns contain the models. CNN2L using Word2Vec gave the best results with 0.903 accuracy for the English dataset, 0.923 for the Portuguese dataset, 0.803 for the Japanese dataset, as indicated in bold in Table 5.

CNN2L performs better when compared to CNN1L due to a combination of two factors: (i) in the convolution layer, we use multiple filters representing n-grams, i.e., 32 filters that slide over 3, 4, and 5 words, which can contribute to better accuracy; and (ii) the max pooling layer helped to extract the maximum value of each region. The results suggest that two layers of convolution and max pooling operations are sufficient to obtain good results when using short text and small datasets.

CNN3L results were worse than CNN2L. This occurs because a small dataset using the before mentioned CNN2L architecture helped the model to achieve the local minimum with two convolution layers.

4.3 Comparison with baseline

The SVM and CNN1L models were used as a baseline to compare with the CNN 2L model, as presented in Table 5. The CNN1L was built based on the architecture of Kim's CNN model [7]. Such CNN is composed of one convolution layer and one max pooling layer; filter windows of {3, 4, 5}; Google News Word2Vec and embedding dimensionality of 300. However, the following adjustments have been made to process our multilingual datasets: the number of filters was set to 64; the batch-size to 64; a character-level approach was used to Japanese dataset and word-level method for Portuguese and English datasets. The results of CNN2L were 1.53% higher than CNN1L in the Japanese dataset, 12.57% higher in the English dataset, and 14.73% higher in the Portuguese dataset.

The accuracy of SVM was 0.795 using the English dataset, 0.825 in the Portuguese dataset, and 0.725 in the Japanese dataset. The experiment suggests that even with a small dataset and a simple CNN architecture with two convolution layers and two pooling layers, by adjusting the hyperparameters it is possible to get superior results (13.58% higher in English dataset, 11.87% in Portuguese dataset and 10.75% in Japanese dataset) comparing with traditional SVM model.

For evaluation purposes, English Twitter dataset related to Philipine floods (2012), Colorado floods (2013), Queensland floods (2013) and Manila floods (2013) used in Caragea et al. [2] research and available at CrisisLex project were processed through the CNN 2L

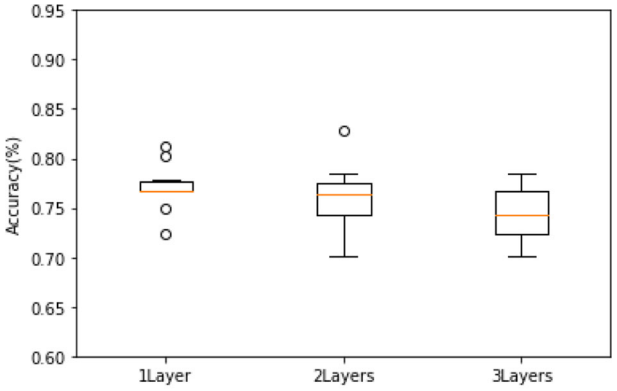
Table 5 Results

Dataset	CNN1L		CNN2L		CNN3L		SVM	CNN
	None	W2V	None	W2V	None	W2V	None	W2V
EN-T	0.771	0.802	0.759	**0.903**	0.774	0.885	0.795	–
PT-T	0.781	0.805	0,785	**0.923**	0.720	0.686	0.825	–
JP-T	0.789	0.792	0.776	**0.803**	0.732	0.744	0.725	–
CLEX	–	–	–	**0.848**	–	–	–	0.825

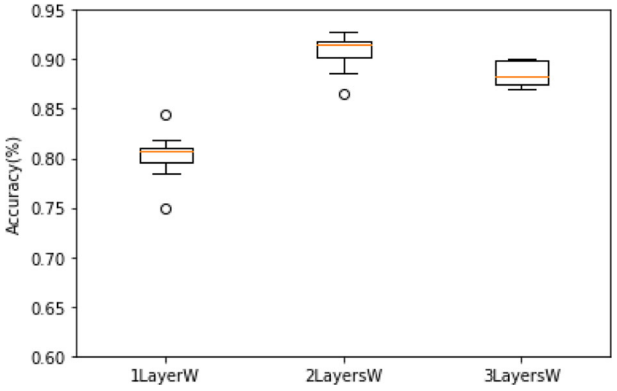
model. The flood dataset has 3671 tweets manually annotated into informative and noninformative posts. The accuracy of the CNN 2L model was 0.837 which was slightly greater than 0.825 reported on Caragea et al. [2] research which uses similar CNN architecture with one convolution layer and one pooling layer.

4.4 Impact of accuracy by using GM-ShorT

In this subsection, we analyze the distributional characteristics of a group of scores. Figure 3a shows the results concerning the accuracy distribution of the CNN models with one, two and three layers using the English dataset without Word2Vec. It is observed that CNN models with one and two layers obtained similar results (0.7672 and 0.7639 on median respectively) when compared to CNN models three layers. It is noteworthy that when using an English dataset with Word2Vec (Fig. 3b), there is a higher hit ratio (0.9146 on median) to perform the process of multilingual short text categorization in GM-Short, regardless of the layers. It is observed that the median value is higher in the two layers CNN model and the interquartile range are shorter and less distributed compared with the boxplot without using Word2Vec. This presents stability in the results for categorizing short text.

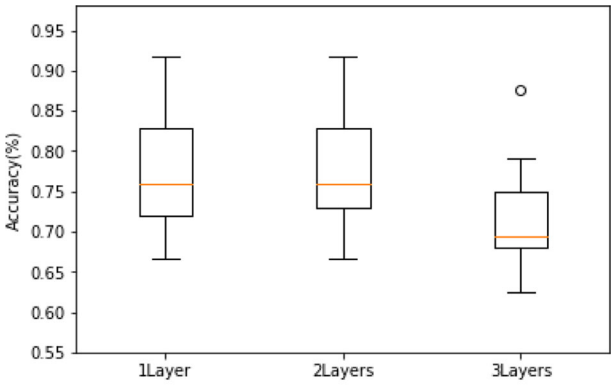


(a) English dataset without Word2Vec

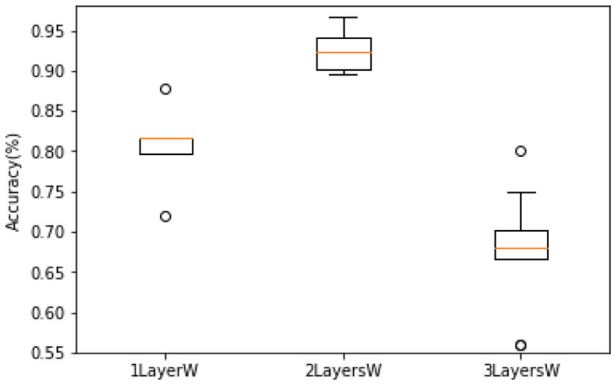


(b) English dataset with Word2Vec

Fig. 3 English dataset



(a) Portuguese dataset without Word2Vec

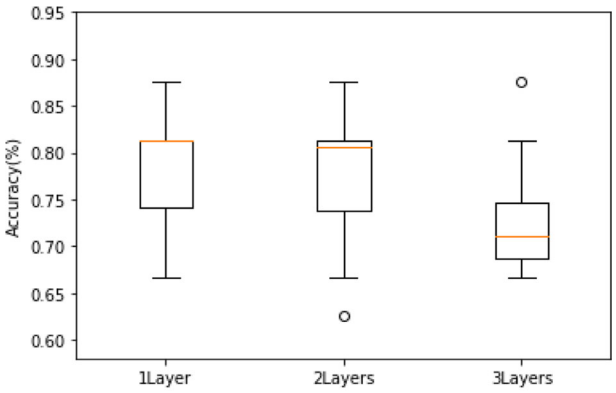


(b) Portuguese dataset with Word2Vec

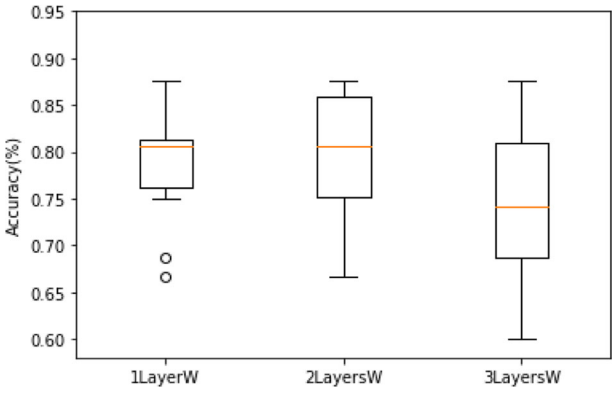
Fig. 4 Portuguese dataset

Figure 4 illustrates box plot for Portuguese dataset. The results are quite similar to the English dataset. This ratifies our solution for a generic framework for multilingual short text categorization. While the experiments without pre-trained vocabulary resulted in lower median values and more distributed accuracy values, the experiments using Word2Vec resulted in higher median values (i.e. 0.9240), less distributed and shorter interquartile range. This is because without the Word2Vec on GM-ShorT it is not possible to generalize the results to categorize the short texts.

Figure 5 illustrates box plot for Japanese dataset. In Fig. 5b, CNN model of two layers shows that even though the third quartile value is higher compared with the other CNN models of the Japanese dataset, we cannot observe a great improvement in the experiment using Word2Vec and without pre-trained vocabulary. This is because Japanese tweets usually contain a mix of Japanese characters and English words making difficult to categorize correctly using Japanese Word2Vec.



(a) Japanese dataset without Word2Vec



(b) Japanese dataset with Word2Vec

Fig. 5 Japanese dataset

Figure 6 illustrates comparative analysis among English, Portuguese and Japanese datasets for the CNN model with two layers with/without Word2Vec. For the best-case scenario, the English dataset, which is the largest dataset, resulted in a short box plot comparing with the Japanese and Portuguese suggesting that the test results were more stable and less distributed. It is worth noting, however, that regardless of the dataset used, the results are satisfactory to categorize short text messages in GM-ShorT.

It is better to mention the following factors for analyzing the obtained results: (i) pre-trained Word2Vec helps to improve the results, and we used three language-specific Word2Vec with the average matching of 81%, i.e., 81% of the words used in the datasets were found in Word2Vec; (ii) multiple filters representing n-grams convolving over 3, 4 and 5 words contribute to better accuracy; (iii) by using character-level CNN for a non-alphabetical language such as Japanese and Chinese enhances the possibility of the model convergence; and (iv) the results are in line with the general fact that complex deep CNN models require more training data, and simpler and shallow CNN models can outperform with small datasets composed by short texts.

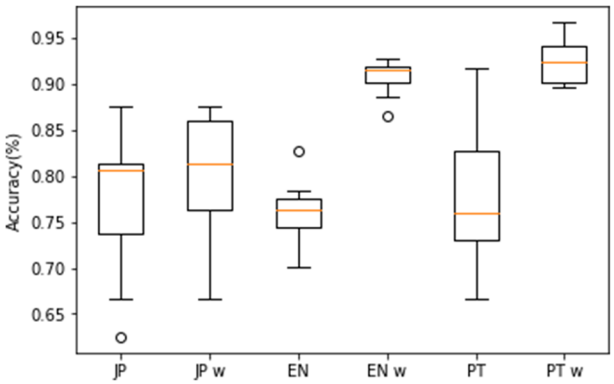


Fig. 6 Comparative analysis of English, Portuguese and Japanese datasets for CNN model with two layer with/without Word2Vec. For instance, 'JP' is Japanese dataset and 'JP w' is Japanese dataset with Word2Vec

4.5 Impact of computation time

In this subsection, we analyze the computation time taken to training and testing CNN2L with Word2Vec. CNN2L model was run in Intel i5 CPU 2.4 GHz because we used small datasets based on short text. Table 6 shows the parameters that most influenced on the computation time. All three datasets were trained over 50 epochs. Despite the difference between English dataset size (1162 tweets) and Japanese dataset size (157 tweets), both datasets took an average of 60 min. The computation time with Japanese dataset was influenced by the embedding dimension (128), the maximum length of each tweet (146) considering the character-level approach, resulting in 1,708,965 parameters. On the other hand, in the English dataset, which used a word-level approach, the computation time was influenced by the word embedding dimension (300) and dataset size (1162 tweets), resulting in 1,403,873 parameters. Finally, the low computation time (11 min) taken by Portuguese dataset was due to small embedding dimension (50), small maximum length (32), and small dataset size (246 tweets), resulting in 470,545 parameters.

5 Conclusion and future works

Online social media generates a massive amount of data daily and many types of research have been done in the field of text categorization. However, find useful information to satisfy user's needs is not an easy task. There are many challenges to overcome especially in short text categorization such as misspellings, typos, irony words and lack of context. In this context, online social media is a powerful source of information capable of influence

Table 6 CNN 2L computation time

Data	#Twts	Emb. size	Max. len.	Param.	Epoch	Time (min)
EN-T	1162	300	38	1,403,873	50	60
PT-T	246	50	32	470,545	50	11
JP-T	157	128	146	1,708,965	50	60

users' decisions and help people in case of disaster and emergency. During emergencies, the amount of information posted on social media exceeds the capacity of the public to consume it. In addition, crises and emergency events may spread fast in different regions and countries affecting people who use different languages.

In this research, we developed GM-ShorT, a Generic framework for Multilingual Short Text Categorization based on CNN. For this, GM-ShorT collects online social media data. Such data were used as input on CNN which is combined with a word embedding mechanism to categorize short text messages. GM-ShorT has been validated in three different languages: English, Japanese and Portuguese. The results show that a simple CNN model with two layers outperforms SVM in text categorization, independently of the language used in the text. The accuracy is 13.58% higher than SVM with the English dataset, 11.87% with Portuguese dataset and 10.75% with Japanese dataset. Furthermore, the experiments suggest that with few hyperparameter changes and using a word-level approach for English and Portuguese, and character-level approach for Japanese texts, the same CNN model can be used in multilingual text categorization.

The main contribution of this article is an end-to-end Generic Framework that can be used to categorize social media posts using small datasets and adapted for multilingual classification. The fact that the Framework underlying architecture is not a complex deep CNN model enables to be easily adapted to new text categorization cases. Moreover, the model does not require a large annotated dataset and can be trained with a small dataset with good results. The information can be classified according to the categories related to each emergency. The categorized messages can be used for various purposes such as situation report, inform hospitals prepared for the emergency, donation campaign and others.

In the future research, we intend to evaluate GM-ShorT's adaptability in different contexts and improve the Framework performance using BiDirectional LSTM with the attention model. We also plan to use a meta-learning approach to train small datasets for improving the robust applicability of GM-ShorT.

References

1. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137
2. Caragea C, Silvescu A, Tapia AH (2016) Identifying informative messages in disaster events using convolutional neural networks. In: International conference on information systems for crisis response and management, pp 137–147
3. Georgakopoulos SV, Tasoulis SK, Vrahatis AG, Plagianakos VP (2018) Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th hellenic conference on artificial intelligence, pp 1–6
4. Geraldo Filho P, Villas LA, Gonçalves VP, Pessin G, Loureiro AA, Ueyama J (2019) Energy-efficient smart home systems: infrastructure and decision-making process. Internet Things 5:153
5. Hartmann N, Fonseca E, Shulby C, Treviso M, Rodrigues J, Aluisio S (2017) Portuguese word embeddings: evaluating on word analogies and natural language tasks. arXiv:1708.06025
6. Johnson R, Zhang T (2014) Effective use of word order for text categorization with convolutional neural networks. arXiv:1412.1058
7. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv:1408.5882
8. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence
9. Lu Y, Sakamoto K, Shibuki H, Mori T (2017) Construction of a multilingual annotated corpus for deeper sentiment understanding in social media. Inf Media Technol 12:111
10. Lu Y, Sakamoto K, Shibuki H, Mori T (2017) Are deep learning methods better for twitter sentiment analysis. In: Proceedings of the 23rd annual meeting of natural language processing (Japan), pp 787–790

11. Mandelbaum A, Shalev A (2016) Word embeddings and their use in sentence classification tasks. arXiv:1610.08229
12. Merchant RM, Elmer S, Lurie N (2011) Integrating social media into emergency-preparedness efforts. New Engl J Med 365(4):289
13. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
14. Neto J, Filho G, Mano L, Ueyama J (2018) Verbo: voice emotion recognition database in Portuguese language. J Comput Sci 14(11):1420
15. Nguyen DT, Joty S, Imran M, Sajjad H, Mitra P (2016) Applications of online deep learning for crisis response using social media information. arXiv:1610.01030
16. Oliveira DF, Chan KS (2019) The effects of trust and influence on the spreading of low and high quality information. Phys A: Stat Mech Appl 525:657
17. Rocha Filho GP, Meneguette RI, Maia G, Pessin G, Gonçalves VP, Weigang L, Ueyama J, Villas LA (2020) A fog-enabled smart home solution for decision-making using smart objects. Future Gener Comput Syst 103:18
18. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. ACM, pp 851–860, Proceedings of the 19th international conference on World wide web
19. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR) 34(1):1
20. Simon T, Goldberg A, Adini B (2015) Socializing in emergencies—a review of the use of social media in emergency situations. Int J Inf Manag 35(5):609
21. Sosa PM, Sadigh S (2016) Twitter sentiment analysis with neural networks. Academia. edu
22. Steiner-Correa F, Viedma-del Jesus MI, Lopez-Herrera A (2018) A survey of multilingual human-tagged short message datasets for sentiment analysis tasks. Soft Comput 22(24):8227
23. Sun F, Belatreche A, Coleman S, McGinnity TM, Li Y (2014) Pre-processing online financial text for sentiment classification: a natural language processing approach. In: 2014 IEEE conference on computational intelligence for financial engineering & economics (CIFEr). IEEE, pp 122–129
24. Vilas AF, Redondo RPD, Crockett K, Owda M, Evans L (2019) Twitter permeability to financial events: an experiment towards a model for sensing irregularities. Multimed Tools Appl 78(7):9217
25. Wang J, Wang Z, Zhang D, Yan J (2017) Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI, pp 2915–2921
26. Yang Y, Zheng L, Zhang J, Cui Q, Li Z, Yu PS (2018) TI-CNN: convolutional neural networks for fake news detection. arXiv:1806.00749
27. Zhang X, LeCun Y (2017) Which encoding is the best for text classification in Chinese, English, Japanese and Korean? arXiv:1708.02657
28. Zhang Y, Wallace B (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv:1510.03820
29. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, pp 649–657