News Classification from Social Media Using Twitter-based Doc2Vec Model and Automatic Query Expansion

Lap Q. Trieu
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
tqlap@apcs.vn

Huy Q. Tran
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
tqhuy@apcs.vn

Minh-Triet Tran
University of Science, VNU-HCM
Ho Chi Minh city, Vietnam
tmtriet@fit.hcmus.edu.vn

ABSTRACT

News classification is among essential needs for people to organize, better understand, and utilize information from the Internet. This motivates the authors to propose a novel method to classify news from social media. First, we propose to vectorize an article with TD2V, our pre-trained Twitter-based universal document representation following Doc2Vec approach. We then define Modified Distance to better measure the semantic distance between two document vectors. Finally, we apply retrieval and automatic query expansion to get the most relevant labeled documents in a corpus to determine the category for a new article. As our TD2V is created from 297 million words in 420,351 news articles from more than one million tweets in Twitters from 2010 to 2017, it can be used as one of the efficient pre-trained models for English document representation in various applications. Experiments on datasets from different online sources show that our method achieves the classification accuracy better than existing methods, specifically $98.4 \pm 0.3\%$ (*BBC* dataset), $98.9 \pm 0.7\%$ (*BBC Sport* dataset), $94.1 \pm 0.2\%$ (*Amazon4* dataset), and 78.6% (*20NewsGroup* dataset). Furthermore, in the classification training process, we just encode all articles in the training set with TD2V, not to train a dedicated classification model for each of these datasets.

CCS CONCEPTS

• **Information systems → Clustering and classification**; *Content ranking*; *Top-k retrieval in databases*; *Web and social media search*;

KEYWORDS

news classification, Doc2Vec, Twitter, document embedding, automatic query expansion

ACM Reference Format:
Lap Q. Trieu, Huy Q. Tran, and Minh-Triet Tran. 2017. News Classification from Social Media Using Twitter-based Doc2Vec Model and Automatic Query Expansion. In *SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3155133.3155206

SoICT '17, December 7–8, 2017, Nha Trang City, Viet Nam

ACM ISBN 978-1-4503-5328-1/17/12…$15.00
https://doi.org/10.1145/3155133.3155206

1 INTRODUCTION

Increasing amount of digital information is disseminated worldwide via Internet [25]. However, the abundance of news from various online sources may cause distraction and confusion to netizens. Document classification [12, 21] is one of the effective techniques to better organize online news so that people can understand and exploit news articles more efficiently, and even detect and prevent sensitive data leakage.

A news article, or a text document, in general, should be converted into a numerical vector to be used as an input for different text analysis tasks, such as classification[25], clustering[4], or sentiment analysis [16]. One common approach for text document representation is Bag-of-Words (BoW) or Bag-of-n-grams models [5, 25]. However, BoW methods may not fully capture document semantic[11]. For example, the word order is not exploited in these methods, resulting in semantic loss in document representation.

To better represent the textual content and internal semantic relationships of a document, different techniques for word embedding are proposed [3, 14, 22, 23]. Quoc Le et.al propose an unsupervised framework named Paragraph Vector, which is Doc2Vec implementation, to address the problem related to semantic meaning [11]. As word/document embedding is considered as a promising approach for text representation, we follow this new trend in our proposed method for news classification from social media.

It would be ideal to train a dedicated model for document representation in a particular domain. However, this approach is feasible only if we have large enough data and sufficient resource (computing system and time), to train document embedding models. This motivates our proposal to create TD2V (Twitter-based Doc2Vec), a universal model to vectorize English documents in different domains. From Twitter, we gather 422,351 articles from different media channels in seven years (2010-2017) with 297,298,525 words. Then we process and select 271,620 unique English words and phrases as the universal vocabularies to train the Doc2Vec model (using distributed memory scheme).

We then define Modified Distance, a new metric to better measure the semantic distance between two document vectors. This new metric utilizes both the direct distance between a query article q and a known article x and the average indirect distance from k nearest neighbors of q to x. The direct distance can be any regular metrics, such as L1, Euclidean, or cosine.

Finally, we propose a novel method to classify a news article q using article retrieval and automatic query expansion (AQE [15]). From the initial ranklist containing k nearest neighbors of q, we use our proposed Modified Distance to re-rank articles in our corpus C. The label for q is determined by majority voting scheme from the top l in this re-ranked list. Our method not only determines

the label for a news article but also finds its related articles sorted in descending order of semantic similarity. Experiments show that predicting category for a new article based on the re-ranked list (with auto query expansion and Modified Distance) achieves higher accuracy than with kNN approach.

Experiments on different datasets show that our news classification method achieves the accuracy higher than those of existing methods/ Specifically, our method achieves the accuracy of 98.4 ± 0.3% (*BBC* dataset[4]), 98.9 ± 0.7% (*BBCSport* dataset[4]), 94.1 ± 0.2% (*Amazon4* dataset[2]) and 78.6% (*20NewsGroup* dataset).

The main contributions of our work are as follows:

(1) We create TD2V, a pre-trained model for English document representation. As we gather a large corpus (422K documents with 297M words) from many online channels to cover a wide range of topics and domains to train this model, TD2V is expected to efficiently capture semantics of news articles and text documents in various applications. Experiments show that TD2V can be used as one of the efficient pre-trained document embedding models, helping others saving their effort to train a dedicated document embedding model from a particular dataset as well as addressing the problem of lacking large data for training.
(2) We define Modified Distance to better measure the semantic distance between two document vectors. The key idea is the semantic distance from article q to article x should be the weighted combination of their direct distance and the average direct distance between the k neighbors of q to x.
(3) We propose a novel method to classify news based on retrieval and automatic query expansion. As we restate the classification problem into the retrieval task, our method can fetch the most relevant labeled articles corresponding to a new article, then determine its label. By this way, our method allows the set of categories to be updated easily without retraining the classification model.

The structure of this paper is as followed. In section 2, we briefly review existing methods and approaches for text document representation and classification. Section 3 presents our proposal for training TD2V, a universal pre-trained model based on Doc2Vec from Twitters, and the process to classify news from social media sources using retrieval and automatic query expansion. Section 4 presents datasets and experimental results. We also develop a system for news clustering and visualization, which is described in Section 5. Finally, the conclusion is in Section 6.

2 RELATED WORK

2.1 Document Representation

Bag-of-Words (BoW) model in linguistic domain, first proposed by Z.Harris in 1954 [5], is among the most common and efficient approaches for document representation. The method relies mainly on counting the number of occurrence in a document of a word, and these numbers are used to form the vector representation of that document [25]. Although these methods are simple and can achieve high accuracy for different tasks for document analysis, they still have limitations in capturing the semantic of a document [11], such as the lost of word order, data sparsity, high dimensionality, and lacks of semantic relationships between words.

Rumelhart et al. first proposed distributed representations for words [20], a new way to solve the problem of embedding words into a vector space while keeping the semantic meaning in the context. The method quickly became successful paradigm and adopted for different natural language processing fields such as named entity recognition, parsing and machine translation [3], [23]. T. Mikolov et.al. propose several methods to improve both training performance and semantic accuracy of vectors for distributed representation of words and phrases [13], including a method for finding phrases in text by combining words together, resulting in the reduction of memory space and improving training speed. Y. Kim et. al. show a series of experiments with trained convolutional neural networks (CNN) using the pre-trained word vectors to classify sentences. Data representing at phrase [14] or sentence [22] levels are also introduced.

Based on the idea of [13], Quoc Le et.al propose an approach for distributed representation of sentences and documents[11]. They introduce a method named Paragraph Vector, which has the ability to learn and represent document in an arbitrary length such as phrase, sentence and paragraph. Their model works well on embedding documents on vector space model with a surprising accuracy. We are inspired by their work and adopt Doc2Vec model in our process to classify news from social media.

2.2 Document Classification

As one of the common methods for managing unstructured data, document classification [12, 21] is used with different approaches. k-Nearest Neighbor is one simple yet efficient method for text and document classification [7]. The algorithm classifies objects using voting method comparing several nearest labeled samples to decide which label it should choose.

Naive Bayes classifier[18], [19] is a classical method for document classification. This method uses the posterior probability computed from the document to decide its class by comparing to the highest posterior probability. Support Vector Machine is also another common method to document[6] or document sentiment classification[16]. After vectorizing documents, SVM classifiers are trained to determine the label for a new document. As a supervised classification approach, SVM is good at handling large feature spaces and considered as one of the effective document classification methods due to its high precision [6, 17].

Using deep neural networks is a new trend for document classification. The first main approach is to propose a novel network for the whole process of document classification. For examples, in Hierarchical Attention Networks [24] or Recurrent Neural Network for text classification[9], a network structure for document/text classification is proposed without an explicit separation between document representation and document classification. Another main approach is to propose a network for document representation and apply a classical technique to classify a document vector. Quoc Le et.al. use Paragraph Vector to turn the datasets into a sets of vectors and then apply logistic regression for classification [11]. In this paper, we follow the second approach to divide the classification process into two phases: document representation and classification. By this way, we can reuse the document representation for other tasks, such as document retrieval or clustering.

3 PROPOSED METHOD

3.1 Overview of Method

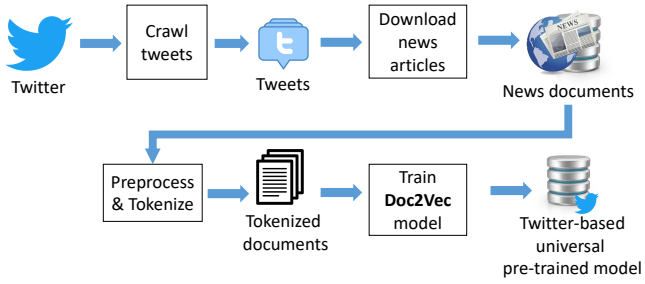

Figure 1: Train TD2V, a universal document embedding.

In this section, we present our proposed method to classify news articles from an arbitrary social media sources, such as newspaper, social networks, newsgroups, or reviews in e-commerce sites, etc. There are two main components in our method: news representation and news classification.

News Representation: We aim to create a universal document embedding model TD2V, based on Doc2Vec, to vectorize an arbitrary news article from different social media sources (Figure 1).

To cover a wide range of topics and categories of articles, we gather a large collection of more than 422,351 documents with more than 298 million English words from different channels within seven years (2010-2017) to train our universal TD2V. The details of the processes to train TD2V and to use it to encode an arbitrary news article into a fixed-length vector are presented in Section 3.2.

News Classification: Figure 2 illustrates the overview of our proposed news classification process. We first encode each labeled article in the training set using our universal document embedding TD2v. For a new article q, we retrieve a ranklist of relevant labeled documents, then perform automatic query expansion (AQE) to re-rank this list. The label for q is determined with majority vote from the top k items in this ranklist.

This approach seems to share similarity with k-NN classification. However, in our method, the final label of a new article is determined from the re-ranked list of retrieved documents, not directly from the k nearest neighbors of the article q. The details of this process is described in Section 3.3.

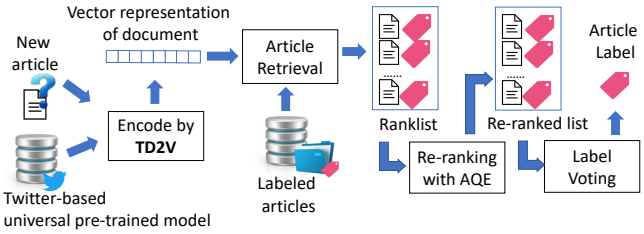

Figure 2: Overview of news classification process.

3.2 News Representation with Twitter-based Doc2Vec Model

3.2.1 Create Twitter-based Doc2Vec Model. Figure 3 illustrates our process to gather online news from tweets and build the dictionary of vocabulary terms for Doc2Vec model. Using our Twitter crawler, we gather millions of tweets from different channels within seven years (2010 to 2017). For each tweet, we extract the attached URL and retrieve the content of web page from the link. After filtering duplicated and short articles with less than $n_{minWords} = 100$ words, we obtain the collection of 422,351 documents.

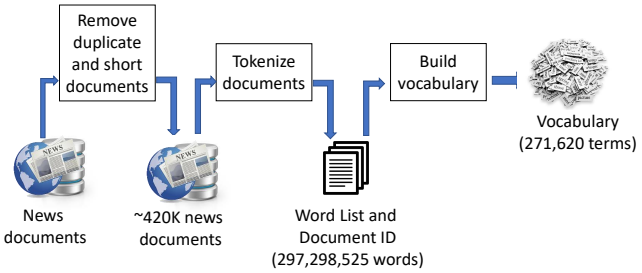

Figure 3: Build dictionary of vocabulary terms from online news.

Based on the crawled dataset, we tokenize each document to obtain a list of words and attach an unique id to that list for training purpose. Based on the set of 297,298,525 tokenized words, we obtain the global set of $n_V = 271,620$ terms after filtering the words with frequency lower than $n_{minFrequency} = 5$.

After collecting dataset for training, we initialize the model with specific parameters such as the number of dimensions (*size*) for document vector representation, the window size we need during training process (*window*) or the number of threads we should use to improve performance task (*workers*). The details of selecting these parameters are discussed in Section 4.1.

Among the two training algorithms, distributed memory (PV-DM) and distributed bag of words (PV-DBOW), we decide to use PV-DM to train our document embedding as it achieves better results in representing documents for news classification with different datasets. In fact, PV-DBOW does not involve any nearest neighbor input vectors per word, it would not be as good as PV-DM in capturing the relationships with similar words or phrases.

We build a list of tagged documents as an input for training. Actually, Doc2Vec uses two things to train our model, tags and documents. In our case, each tag is unique corresponding to each document. To make it understand our data input, we need to convert the document and tag to a TaggedDocument iterator object. TaggedDocument is an object to hold the pair of value of a list of tokenized words in a document and a list of tags attached to that document. In our case, for each document, we attach only one unique tag. Because we want to optimize memory usage during the process of building the object, we create a generator object instead of iterator object. The generator object not only returns an iterator, but it can be used to control the behavior of iteration, resulting in less memory requirement.

Each list of tokenized words of a document is filtered such that words that are not in the vocabulary of the whole training dataset are removed. Based on the set of new filtered words of that document, a list of training samples is generated using the movement of window context.

Based on the structure of network in Doc2Vec method, we design our network to train our document embedding as in Figure 4. The input vector is the concatenation of word ID vector (with $n_V = 271,620$ elements) and document ID vector (with $n_D = 422,351$ elements).The hidden layer consists of *size* nodes and the output layer has n_V nodes. In this figure, W is a $n_V \times size$ matrix to represent the word context in a window surrounding a target word while D is a $n_D \times size$ matrix to represent features representing the document context. The objective of the training process is to find W, D, and W' to maximize the probability to predict a target word given context words and a document context.

$$\max \sum \log P\,(\,\text{targetword}|\,\text{contextwords}, \text{documentcontext}) \quad (1)$$

After finishing the training process, we obtain word embeddings W and document embedding D for our model TD2V from all documents in the training corpus.

Usually, when training for the model, the correct output of the network is a one-hot vector that the correct output neuron will be 1 and for all of the other thousands of output neurons to output a 0. Instead of that, with negative sampling, we only need to choose a small number of negative words (incorrect output) to update weight for. We choose the number of negative samples is 5, so for each training sample, there are 5 random negative words and 1 positive word (correct output) in the output. Therefore, there are totally 6 word vectors updated in the weight matrix.

Since the learning rate decreases over the iteration of data, each tag of a single TaggedDocument object is only trained once with a fixed learning rate. Thus we repeat the training process over the data 10 times while each time decreases the learning rate by 0.002 and fixes it to prevent the decay.

Figure 4 (right) illustrates the process to use TD2V to encode a document d_{test}. We replace the document ID vector in the input layer by a single value node d_{test}, keep the learned matrix W and W' fixed, and learn the variable D (now with the size of $1 \times size$) as document embedding of d_{test}. The optimization problem is to maximize the probability predict a target word given context word and this specific document d_{test}.

$$\max \sum \log P\,(\,\text{targetword}|\,\text{contextwords}, \text{documentcontext} = \text{d}_{\text{test}}) \quad (2)$$

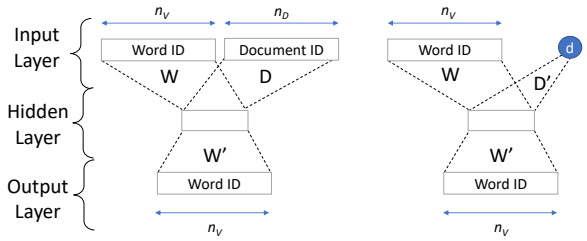

Figure 4: Train TD2V (left) and use TD2V to encode a document (right).

3.3 News Classification with Retrieval and Query Expansion

To determine the label of category for a new article q, we restate the classification problem into the retrieval task. The goal of our proposed method is not only to determine the appropriate category for q but also to retrieve related articles in the corpus (or previously known articles). Besides, in Section 4, we demonstrate that by applying query expansion technique, we can further boost the accuracy for article classification.

Let $d(x, y)$ be the distance between two articles x and y. We can use different metrics to evaluate the distance between two articles, such as Euclidean, L1, cosine, etc. In Section 4, we present our experiments to choose the optimum metric for our proposed method.

For a new article q, we retrieve a ranklist $RL_k\,(q)$ with k most relevant articles in corpus C, sorted in ascending order of distance from q.

$$RL_k\,(q) = \left(a_{1,q}, a_{2,q}, ..., a_{k,q}\right) \quad (3)$$

where $a_{i,q} \in C, 1 \le i \le k$ and $d\left(q, a_{i,q}\right) \le d\left(q, a_{i+1,q}\right), 1 \le i < k$.

Up to this step, our method is similar to the traditional classification technique kNN. The label for the article q can be directly determined by majority voting scheme from the labels of articles in the ranklist $RL_k\,(q)$, either by equal vote or weighted vote.

In kNN, the classification result is heavily dependent on how well the distance metric can represent the relationship between two articles. Besides, the result may be affected by some noisy neighbors of q. Thus, we aim to further exploit the external relationships from the k nearest neighbors of q to have better view about the context and semantic of q, not just its direct neighbors. Our hypothesis is that, by exploring indirect relationships of an article q, we can reduce both the noise effect of some neighbors of q and the dependency on the quality of the distance metric.

We adopt the idea of automatic query expansion (AQE) to create a re-ranked list $\widehat{RL}_{k,l}\,(q)$ with l articles from the corpus C from the ranklist $RL_k\,(q)$.

The distance from q to an article $x \in C$ is no longer determined by the direct distance $d(q, x)$. We define the ***modified distance*** $\hat{d}\,(q, x)$ as a weighted combination of ***direct distance*** $d(q, x)$ and the ***average direct distance from all*** k ***neighbors*** of q to x:

$$\hat{d}\,(q, x) = \alpha . d\,(q, x) + (1 - \alpha)\,\frac{\sum\limits_{a_i \in RL_k(q)} d\,(a_i, x)}{|RL_k\,(q)|} \quad (4)$$

The parameter α is used to weigh the level of importance between the direct and indirect relationships from q to x. In Figure 5, the solid line represents the direct distance from query q and an article x, and dotted lines are the direct distances from the neighbors $a_{i,q}$ of q to x.

Using our proposed distance metric $\hat{d}$, we create a re-ranked list $\widehat{RL}_{k,l}\,(q)$ of top l relevant labeled articles in corpus C from the original ranklist $RL_k\,(q)$.

$$\widehat{RL}_{k,l}\,(q) = \left(\hat{a}_{1,q}, \hat{a}_{2,q}, ..., \hat{a}_{l,q}\right) \quad (5)$$

where $\hat{a}_{i,q} \in C, 1 \le i \le l$ and $\hat{d}\left(q, \hat{a}_{i,q}\right) \le \hat{d}\left(q, \hat{a}_{i+1,q}\right), 1 \le i < l$.

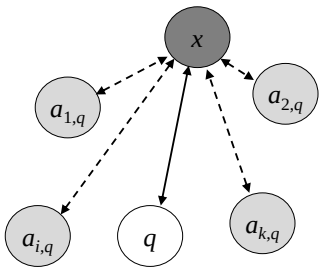

Figure 5: Modified distance from an article q to x.

The label for the new article q can now be determined using majority voting scheme from the labels of articles in the re-ranked list $\widehat{RL}_{k,l}(q)$.

4 EXPERIMENTS

4.1 Train TD2V

The most important parameter is *size*. With the higher number of features in a vector, the embedding model can capture more details. It is recommended that with the dimension about 200, it would generate a good result of output and more than 300 dimensions would produce the higher precision.

However, from experiments, we observe that the higher value does not improve the result much and the amount of time as well as the required memory for training would increase significantly. Therefore, we choose the dimensionality of the feature vectors to be 300 as in other works.

Window size is the value indicates the maximum distance between the predicted word and context words for prediction. For each word in the processing document, the window size is randomly chosen so that the selected value is not beyond the value that we set during the initialization process ($window = 10$).

After selecting the window size, the list of the surrounding words of the current word is selected to be the neuron input while the current word is used for neuron output. For example, if 10 is the randomly selected window size, 20 surrounding words will be used for the input context. If the current word and context words are beyond the sentence boundary, the rest of the context words are ignored.

The collection of 422,351 lists of tokenized words is divided for 10 workers to execute the training process such that each worker processes no more than 10,000 words at any time. The training process for TD2V takes 12 hours in our system (core i7, 16GB) using gensim.

4.2 Description of Datasets

In the following experiments, we evaluate the classification accuracy of our news classification method using the pretrained TD2V document embedding model with the following datasets:

BBCSport[4] is a collection of 737 documents obtained from the BBCSport website in the total of five fields containing athletics, rugby, tennis, cricket and football from 2004-2005. The vocabulary size is 3669.

BBC [4] contains 2225 documents collected from the BBC news website. These documents are from 2004-2005 and classified in five topical areas including business, sport, entertainment, politics, and tech. The vocabulary size is 8865.

Amazon4[2] , a multi-domain sentiment dataset, contains product reviews collected from Amazon website for 4 product types: Kitchen, Books, DVDs and Electronics. Even though this dataset is used for the purpose of sentiment analysis, we can still use it for classification purpose of product types.

20Newsgroups is a collection of nearly 20,000 newsgroup documents, which is divided into 20 different classes. The dataset contains 18846 documents, and the vocabulary size is 32716.

4.3 Distance Metrics

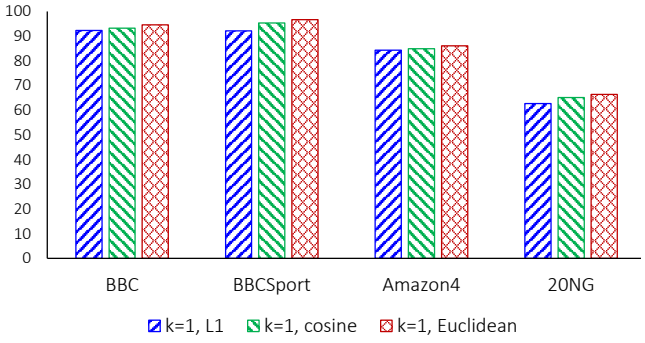

Figure 6: Classification accuracy with $k = 1$ and different distance metrics.

The first experiment is to select the appropriate distance metric to evaluate the semantic relationship between two articles using our pre-trained universal TD2V document embedding. Figure 6 illustrates the classification accuracy on 4 datasets using only one nearest neighbor and different distance metrics, including L1, cosine, and Euclidean. The details information of this experiment are presented in the first three cases of Table 1.

For all datasets, Euclidean metric provides the best accuracy. Therefore, we decide to use Euclidean metric to measure the semantic distance between two document vectors using our pre-trained document embedding.

Table 1: Classification accuracy with majority vote from k nearest neighbors

k	Metric	BBC	BBCSport	Amazon4	20NG
1	L1	92.3	92.1	84.3	62.7
1	cosine	93.9	95.3	84.9	65.1
1	Euclidean	94.6	96.7	86.1	66.4
3	Euclidean	95.8	98.2	88.9	**68.2**
5	Euclidean	**96.5**	**98.3**	**89.0**	67.8
7	Euclidean	96.2	97.9	88.6	67.7

4.4 Classification with Majority Vote from Ranklist

In this experiment, our goal is to evaluate the accuracy of classification with majority vote from the ranklist $RL_k(q)$ with k direct nearest neighbors of a new article q. We use Euclidean metric for this experiment.

Figure 7 shows the classification accuracy with majority vote from the ranklist with different number of nearest neighbors (using direct distance d). The details of this experiment are in the last four cases in Table 1.

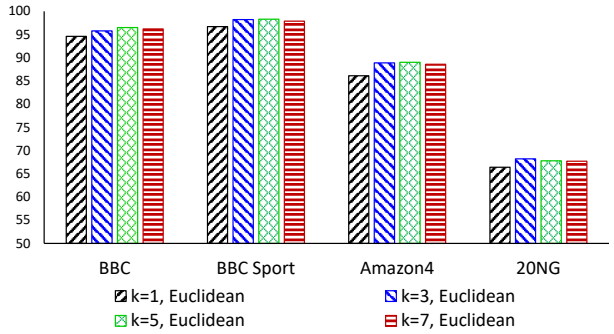

Figure 7: Classification accuracy with different numbers of nearest neighbors and Euclidean distance metric.

It is obvious that classification based on only $k = 1$ neighbor is the worst among the four cases. For $k = 5$, we achieve the best result for the first three datasets (BBC, BBCSport, and Amazon4). For the most difficult 20NG with the lowest accuracy, comparing to the three other datasets, $k = 3$ provides the best result. This might be because we may have higher probability to select noisy neighbors into the ranklist when we expand the ranklist with higher value of k.

Finally, when we increase the value of k, the accuracy does not increase but even slightly decreases because there can be more chance for a noisy neighbor to enter the retrieved ranklist.

From this experiment, we decide to use $k = 5$ and Euclidean metric as the baseline to compare with other configurations with query expansion and re-ranked list in Section4.5.

4.5 Classification with Query Expansion and Re-ranked List

In this experiment, we evaluate the accuracy of news article classification using majority vote from re-ranked list, the result of our method using AQE. From Section 4.4, we decide to use $k = 5$ direct neighbors of a query article q to estimate the modified distance from q to a labeled article x in corpus C. We choose $\alpha = 0.5$ to balance the contribution of direct and average indirect distances. Formula 4 is now rewritten as follows:

$$\hat{d}(q,x) = 0.5d(q,x) + 0.1\sum_{i=1}^{5} d(a_i,x) \quad (6)$$

where $a_i \in RL_5(q)$.

Figure 8 visualizes the classification accuracy for the cases using AQE (with Euclidean distance and $k = 5$) with different length of re-ranked list ($l = 1, 3, 5, 7$). For $l = 1$, the accuracy for BBC and BBCSport datasets slightly decrease while the result for Amazon4 and 20NG increase comparing to kNN with $k = 5$. Thus using only one nearest neighbor (with modified distance) cannot achieve better result than direct distance (with $k = 5$ neighbors). However, if we retrieve more items in the re-ranked list ($l = 3, 5, 7$), the accuracy for all four datasets are improved comparing to the result with the original ranklist ($k = 5$). This demonstrates that our modified distance can better represent the relationship between two articles than the direct distance.

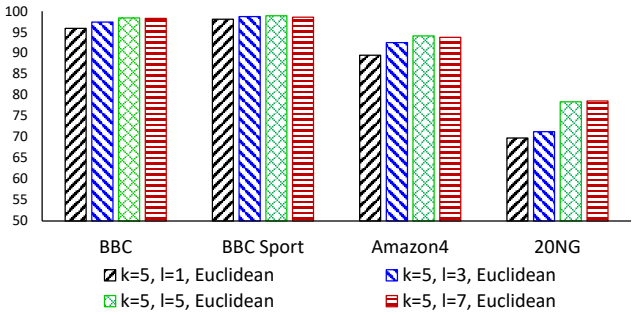

Figure 8: Classification accuracy with AQE and re-ranked list with different length l of re-ranked list.

Table 2: Accuracy comparison between existing methods and our proposed method

Method	BBC	BBCSport	Amazon4	20NG
BoW	97.1 ± 0.4	98.1 ± 0.7	91.3 ± 0.3	68.0
BoW-Full	96.5 ± 0.4	97.5 ± 0.5	90.1 ± 0.2	70.1
LDA	97.3 ± 0.3	97.1 ± 0.6	91.1 ± 0.3	73.2
LSA	97.3 ± 0.4	98.3 ± 0.8	92.0 ± 0.2	74.2
AE	97.2 ± 0.3	98.2 ± 0.7	92.2 ± 0.3	72.5
WMD	96.1 ± 0.3	97.6 ± 0.5	92.1 ± 0.3	74.1
FBoWC$_{mean}$	97.9 ± 0.3	98.5 ± 0.7	93.5 ± 0.3	78.9
FBOWC$_{min}$	97.5 ± 0.3	98.3 ± 0.8	93.5 ± 0.3	75.5
FBoWC$_{max}$	98.2 ± 0.4	98.9 ± 0.8	93.9 ± 0.3	78.5
TD2V	96.5 ± 0.3	98.3 ± 0.8	89.0 ± 0.3	67.8
TD2V+AQE ($l = 5$)	**98.4 ± 0.3**	**98.9 ± 0.7**	**94.1 ± 0.2**	78.4
TD2V+AQE ($l = 7$)	98.3 ± 0.4	98.6 ± 0.7	93.8 ± 0.3	**78.6**

Table 2 shows the comparison of news classification accuracy on four datasets between some existing methods and our proposed ones (with Euclidean metric and $k = 5$ nearest neighbors). All results are reported with mean and standard deviation, except for 20NG as this dataset explicitly splits the train and test sets. We compare our method with Latent Dirichlet Allocation (LDA[1]), Latent Semantic Analysis (LSA[10]), Word Mover's Distance (WMD[8]), Fuzzy Bag-of-Words (FBoWC$_{mean}$,FBoWC$_{min}$, and FBoWC$_{max}$[25]). The results of some other methods, including Bag-of-Words (BoW) and Average Embeddings (AE) [25] are also included in this table.

Experiment results show that our method to use our pre-trained TD2V for document embedding and AQE with modified distance and re-ranked list achieves high accuracy approximating or even better the results of state-of-the-art methods. Specifically, our method achieves the accuracy of 98.4 ± 0.3%, 98.9 ± 0.7%, and 94.1 ± 0.2% for BBC, BBCSport, and Amazon4 datasets, respectively (with $l = 5$ items in re-ranked list), and the accuracy of 78.6% for 20NG (with $l = 7$ items in re-ranked list).

5 NEWS ENCODING, CLUSTERING AND VISUALIZATION

5.1 System Overview

Using our proposed method to encode a document with the pre-trained TD2V model and the news classification method with retrieval and query expansion, we develop ANTENNA, an online system for news analysis and visualization.

Figure 9 shows our process to crawl news from multi online sources. We use Twitter streaming API to crawl feeds from mainstream media with python twitter. After that, we extract specific fields including the URL to the original news and store them in a collection named Tweets DB. Then we use multi threads to download news articles corresponding to new URLs. We use MongoDB to save all data. Currently, our system monitors more then 40 online news sites, including CNN, BBC News, Time, Reuters World, Forbes, Wall Street Journal, New York Times, etc.

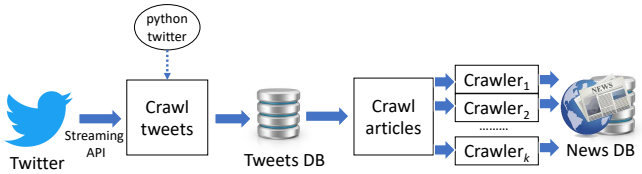

Figure 9: News crawling from multi online sources in ANTENNA.

As illustrated in Figure 10, we analyse a retrieved article in three different aspects:

- **Article Classification**: we use our TD2V to encode an article into a vector with 300 dimensions. Then we apply our classification method to collect related articles in our archive for two tasks: category classification and topic/story clustering. In the first task, we simply assign a category label to the new article from the labels of retrieved articles. However, we take a further step to find the cluster of articles with the story/topic highly related to this new article. The second task can be considered as a fine-grained classification.
- **Named Entity Recognition**: We use *Natural Language Toolkit* (NLTK) to extract named entities from the article. By this way, we can create a graph to represent the relationships between articles and persons/celebrities.
- **Location Recognition**: we use *GeoLite* and*geograpy* to extract known locations in an article and their geolocations. We also use the output from NLTK to collect locations in the article. By this way, we can group articles based on their related locations and visualized on world map.

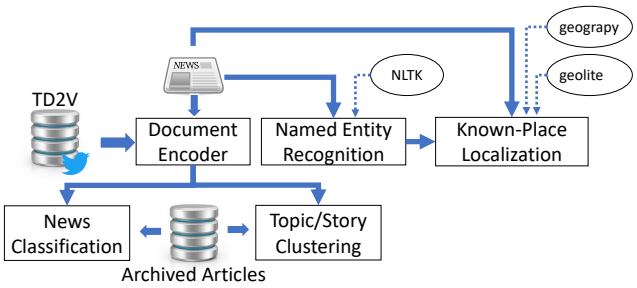

Figure 10: News analysis in ANTENNA.

5.2 ANTENNA - API Web Services

The first subsystem of ANTENNA is a collection of APIs to encode and analyse news articles, and to provide some utilities related to news classification and clustering, relationships of news articles with locations, persons, and organizations. Here is the list of main APIs in ANTENNA:

- **Primitive APIs**: encode an arbitrary English document using our pre-trained model into a fixed length vector and return the extracted named entities from a text document (our wrapper of NLTK to extract named entities)
- **Article Classification and Clustering APIs**: get the category label of an article, get articles related to an article, and get the list of interesting news recently
- **Location-based APIs**: get the list of known locations in an article, query news articles associated with a location, and get the top locations that are mentioned most recently.
- **Person-Article APIs**: get the list of known persons in an article and get the articles related to a person.

5.3 ANTENNA - News Visualization

We develop a prototype for news visualization using our APIs. In Figure 11, we visualize news articles within the last 24 hours in GoogleMaps. News articles are organized into groups based on related locations. As there can be more than one location mentioned in an article, it may appear in multi groups. At each location, we group the articles based on their categories and stories. The more articles associated with a location, the larger the icon appears on the map at that location. Users can click on a marker in the map to view the list of articles or stories at that location. For each article, our system allows users to explore its related articles based on our Modified Distance.

Figure 11: Data visualization on Google Maps.

Figure 12: Visualization as a Force-Directed Graph to represent the relationships between persons and articles.

Figure 12 shows a Force-Directed Graph to represent the relationships between persons and articles. Each article contains a set of related persons. Users can explore the relationship between persons and articles and easily find out a set of articles related to a person. It can also visualize how strong the connection between a person and an article is based on the distance between its nodes.

6 CONCLUSION

In this paper, we propose a new method for news classification from various online social media sources. We first create a pre-trained document embedding model TD2V. As TD2V is trained with a sufficient large corpus (422,351 news articles in 7 years with 298 million words), it is expected to be used as a pre-trained Doc2Vec model to encode an arbitrary English text document for different applications in various domains. Using TD2V can be the solution to the problem of lacking large enough dataset for training a document embedding model and help users save time and effort to train models. Besides, TD2V can also be refined (transfer learning) to adapt to a new document set in a specific domain.

In our method, we use retrieval and query expansion to get the re-ranked list of documents related semantically to a new article, then determine the classification label for this article. This approach, together with our proposed Modified Distance, can be applied to another language when we have a new document embedding model for that language.

Experiments on four different datasets (*BBC*, *BBCSport*, *Amazon4*, and *20NewsGroup*) demonstrate that our method can achieve classification accuracy better than existing methods. Besides, our method does not require to train a classification model for each dataset. For each dataset, we only need to encode all articles in the training set using TD2V.

We also develop the first version of ANTENNA, an online system for news analysis and visualization. The system starts to run in June 2017 and there are more than 120,000 articles from more then 40 social media sources retrieved and processed in our system. Furthermore, our proposed method for document classification with TD2V and AQE is the core component in our system for sensitive document classification for data leakage prevention.

ACKNOWLEDGEMENT

This research is funded by Department of Science and Technology, Ho Chi Minh city, under grant number 40/2015/HD-SKHCN.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022.
[2] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, 440–447.
[3] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning.. In *ICML* (2008-08-14) *(ACM International Conference Proceeding Series)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.), Vol. 307. ACM, 160–167.
[4] Derek Greene and Pádraig Cunningham. 2006. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*. ACM, New York, NY, USA, 377–384.
[5] Zellig Harris. 1954. Distributional structure. *Word* 10, 23 (1954), 146–162.
[6] Thorsten Joachims. 1998. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*. Springer-Verlag, London, UK, UK, 137–142.
[7] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings to Document Distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, 957–966.
[8] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings to Document Distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, 957–966.
[9] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In *AAAI*, Blai Bonet and Sven Koenig (Eds.), Vol. 333. 2267–2273.
[10] T.K. Landauer, P.W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse processes* 25 (1998), 259–284.
[11] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML (JMLR Workshop and Conference Proceedings)*, Vol. 32. JMLR.org, 1188–1196.
[12] Larry M. Manevitz and Malik Yousef. 2002. One-class Svms for Document Classification. *J. Mach. Learn. Res.* 2 (March 2002), 139–154.
[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).
[14] Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Journal of Cognitive Science* 34, 1 (2010), 1388–1429.
[15] Mandar Mitra, Amit Singhal, and Chris Buckley. 1998. Improving Automatic Query Expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*. ACM, New York, NY, USA, 206–214.
[16] Rodrigo Moraes, JoÃčo Francisco Valiati, and Wilson P. GaviÃčo Neto. 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Syst. Appl.* 40, 2 (2013), 621–633.
[17] Pratiksha Y. Pawar and S. H. Gawande. 2012. A Comparative Study on Different Types of Approaches to Text Categorization. *International Journal of Machine Learning and Computing* 2, 4 (2012), 423–426.
[18] Michael Pazzani and Daniel Billsus. 1997. Learning and Revising User Profiles: The Identification ofInteresting Web Sites. *Mach. Learn.* 27, 3 (June 1997), 313–331.
[19] Irina Rish. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3. IBM New York, 41–46.
[20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (Oct. 1986), 533–536.
[21] F. Sebastiani. 2002. Machine learning in automated text categorization. *Comput. Surveys* 34, 1 (2002), 1–47.
[22] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *TACL* 2 (2014), 207–218.
[23] Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37, 1 (2010), 141–188.
[24] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *HLT-NAACL*.
[25] Rui Zhao and Kezhi Mao. 2017. Fuzzy Bag-of-Words Model for Document Representation. *IEEE Transactions on Fuzzy Systems* PP (2017). Issue 99.