HAL
open science

Content similarity analysis of written comments under posts in social media

Marzieh Mozafari, Reza Farahbakhsh, Noel Crespi

To cite this version:

Marzieh Mozafari, Reza Farahbakhsh, Noel Crespi. Content similarity analysis of written comments under posts in social media. SNAMS 2019: 6th International Conference on Social Networks Analysis, Management and Security, Oct 2019, Grenade, Spain. pp.158-165, 10.1109/SNAMS.2019.8931726 . hal-02363538

HAL Id: hal-02363538

https://hal.science/hal-02363538v1

Submitted on 14 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Content Similarity Analysis of Written Comments under Posts in Social Media

Marzieh Mozafari*, Reza Farahbakhsh*, Noël Crespi*
*CNRS UMR5157, Télécom SudParis, Institut Polytechnique de Paris, Évry, France.
{marzieh.mozafari, reza.farahbakhsh, noel.crespi}@telecom-sudparis.eu

***Abstract*—Written comments to the posts on social media are an important metric to measure the followers' feedback to the content of the posts. But the huge presence of unrelated comments following each post can impact many parts of people engagement as well as the visibility of the actual post. Related comments to a post' s topic usually provide readers more insight into the post content and can attract their attention. On the other hand, unrelated comments distract them from the original topic of the post or disturb them by worthless content and can mislead or impact their opinion. In this paper, we propose an effective framework to measure the similarity of given comments to a post in terms of the content and distinguish the related and unrelated written comments to the actual post. Toward that end, the proposed framework enhances a novel feature engineering by combining a syntactical, topical, and semantical set of features and leveraging word embeddings approach. A machine learning-based classification approach is used to label related and unrelated comments of each post. The proposed framework is evaluated on a dataset of 33,921 comments written under 30 posts from BBC News agency page on Facebook. The evaluation indicates that our model achieves in average the precision of 86% in identifying related and unrelated comments with an improvement of 9.6% in accuracy in comparison with previous work, without relying on the entire article of the posts or external web pages' content related to each post. As a case study, the learned classifier is applied on a bigger dataset of 278,370 comments written under 332 posts and we observed almost 60% of the written comments are not related to the actual posts' content. Investigating the content of both group of related and unrelated comments regarding the topics of their actual posts shows that most of the related comments are objective and they discuss the posts' content in terms of topics whereas unrelated comments usually contain subjective and very general words expressing feedback without any focus on the subject of the posts.**

***Index Terms*—Unrelated content, topic modeling, word embeddings, text similarity analysis, social media, Facebook feedback.**

I. Introduction

News agencies are disseminating the news through social media such as Facebook to a large community of people; meanwhile, people are more interested in following the breaking news and stories from this platform rather than the main news agencies' websites. Comments generated by users are one of the significant sources of information following the posts of news agencies pages in Facebook which can be truthful and related to a post's content, or they can be completely or partially untrue and unrelated. Some popular news agencies' pages in Facebook, such as the BBC News, have millions of readers per day and so leaving the unrelated comments by users can have a negative effect on their visiting traffic and reader's satisfaction [1], [2]. Since readers consider comments as a valid source of supplemental information, they prefer to see comments that are more meaningful and discuss a post's topics rather than unrelated concepts such as personal opinions, advertisements, bot-generated content, etc. Therefore, identifying such unrelated comments following a post is a big challenge in social media content analysis [3]–[5].

A growing body of research has focused on analyzing social media content generated by users [6]–[9]. Many approaches have been suggested, including lexical or syntactic matching, semantic knowledge, latent topic models such as Latent Semantic Analysis (LSA) and Latent Dirichlet allocation (LDA) [10], and word embedding [11], in which each term is represented as a vector computed from unlabeled data to identify similarities between short texts. These efforts have used external corpuses such as Wikipedia or webpages related to post content to enrich their corpus. Other studies have tried to identify unrelated comments that are generated to distribute spontaneous spam, influence public opinions, advertise products and events, etc. by leveraging on text contents or temporal and spatial user behavior in social media [3], [5], [12]. In some content analysis applications, where dealing with posts and following comments as short texts in social media, we may not have access to a post's complete story or to some external corpuses such as Wikipedia or Google web pages related to the post content to enhance existing short texts. On the other hand, in real-time content analysis, using these sorts of external corpora can be time-consuming and thus may have a negative effect on the efficiency of a real-time application.

To address these issues, we propose a combination of syntactical, topical, and semantical features by taking advantage of a word embedding approach to identify related and unrelated comments following the posts of a news agency page in Facebook without referring to a post's entire article. By applying word embedding and extracting abstract semantic concepts in numerical, vector form from both pre-trained corpora and our existing dataset, we can improve topical and semantical features to identify related and unrelated contents. This paper makes the following contributions:

- We propose an effective framework to extract three categories of features: syntactical, topical, and semantic from the posts and following comments of a news agency page on Facebook. We then use these features to identify related and unrelated contents.
- We use word embedding approach within both topical and semantical features to enhance similarity detection without having access to the entire story of a post or external corpuses related to each post content.
- Our experiment results show that by using a combination of topical and word embedding-based features, our model can outperform approaches that just use topical modeling methods to identify related/unrelated contents in terms of accuracy, precision, recall, and F-measure.

The rest of the paper is organized as follows. Section II presents a literature review of the content analysis in social media. The proposed framework, dataset, and its use of syntactical, topical, and semantical features are introduced in Section III. Section IV presents the experiments and the results in identifying related/unrelated comments on a news agency page on Facebook. Finally, Section V draws some conclusions and offers a view of possible future work.

II. Previous Works

A major group of studies has focused on user-generated content (e.g., posts, comments, and reviews) analysis in social media by considering textual contents or temporal and spatial user behaviors

[13]–[15]. Spam content is a specific concept throughout the emails, web-page, blog posts, and comments. Short text type spam such as spam comments following posts in blogs and social networks has attracted further attention [16], [17]. Mishne et al. [18] followed a language-based model to create a statistical model for text generation to identify spam comments in blogs. Bhattarai et al. [19] investigated the characteristics of spam comments in the blogosphere based on their content, with an effort to extract the features of the blog spam comments and classify them by applying a semi-supervised and supervised learning method. Wang et al. [16] aimed to identify diversionary comments as comments designed to deliberately divert readers' attention to another topic on political blog posts. They applied a combination of co-reference resolution and Wikipedia embedding to replace pronouns with corresponding nouns and used the topic modeling method LDA to group related terms in the same topics. A context-aware approach to detect irrelevant comments following posts was proposed by Xie et al. in [3]. Their approach assumed that the context-aware semantics of a comment are determined by the semantic environment where the comment is located. They also focused on facilitating the early detection of irrelevant comments by constructing a corpus of the most similar previous comments to the current posts in the same topic.

As a common approach for topical similarity of texts, topic modeling is used to find hidden topical patterns of words in similar texts [20]. Latent Semantic Analysis (LSA) is the foundational model for the development of a topic model. Since it is not a probabilistic model and thus cannot handle polysemy, other topic models such as probabilistic Latent Semantic Analysis (pLSA) and LDA have been proposed based on LSA [10]. In a corpus, LDA tries to discover a topic distribution over each document and a word distribution over each topic. Both pLSA and LDA need the number of topics and they do not capture the relationship among topics. While topic models can discover latent topics in a large corpus, Dat et al. [21] recently proposed a new approach to make a combination between Dirichlet multinomial topic models such as LDA and latent feature (LF) vectors of words called word embeddings to improve word-topic mapping learned on a smaller corpus. They showed that in the case of datasets with few or short texts, the LF-LDA model outperforms LDA, significantly improving topic coherence and document clustering tasks. Here for the first time, we use LF-LDA as a feature to determine topical similarity in related/unrelated short text identification tasks. We describe this model in detail in Section III-C.

Regarding short text mining, a number of recent efforts focus on using topic modeling methods such as LSA, NMF, and LDA [10] to find similarities between short texts in social media. For the first time, Hieu et al. [22] used LDA to enhance the bag-of-word approach and thereby deal with short and sparse texts by finding most of the hidden topics similar to them from large scale data collections. Xie et al. [3] proposed a framework to identify relevant and irrelevant texts by capturing the semantic of short texts in a context-aware approach. Their work considered topic similarity in short texts to capture their relevancy to each other.

Considering all the previous mentioned studies in identifying related/unrelated comments following a post, we believe that our model has gone beyond the state of the art in using a combination of syntax, topic, and semantic-based features to find similarity between short texts. Our model does not rely on the entire story of a post or external webpages content related to the post in comparison with previous studies [3], [16] and we leverage word embedding approach to enrich the short text corpus. Therefore, it can be applied in different social media applications in which we are just dealing with short texts to categorize them as related/unrelated contents.

III. Methodology and Framework

The proposed framework is depicted in Figure 1. To categorize comments as related/unrelated to a post, our framework takes each post P_i and all comments C_{ij} $(j = 1, 2, \ldots, numberofcomments)$ following it as input, and returns the predicted label as related/unrelated for each comment as output. As a classification problem, our framework has two main parts; Training and Prediction. The Training part has three main components: Pre-processing, Similarity-based feature extraction, and Supervised algorithm. Pre-processing is where we clean the input data by applying some pre-processing methods. The Similarity-based feature extraction component is the most important part of our framework, as it is where features are defined to capture the degree of similarity between post and comments more effectively. It contains three different feature categories: syntactical, topical, and semantical. We try to capture not only the syntactical and topical features of texts but also the context of a word, its relation with other words, the context-dependent semantic similarity, etc. by applying the word embedding approach in topical and semantical categories. To the best of our knowledge, this is the first time that this combination of syntactical and topical features is being linked with a word embedding approach to solve the problem of related/unrelated comments to news agencies' posts in social media. We use both pre-trained word2vec models on Google-News corpus [11] and Wikipedia [23] and word embeddings learned from our corpus. After extracting the features, we apply Support Vector Machines (SVM) to evaluate the performance of the model in identifying related/unrelated comments. After training our model, our framework will be able to predict the label of each new comment through this classification process.

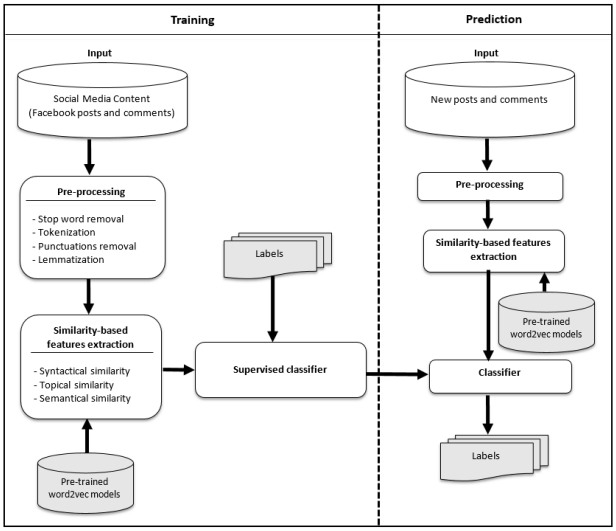

Fig. 1. Schema of the proposed framework.

A. Dataset description

We focused on Facebook news posts originated by news media pages. As a use-case, we identified one popular news agency on Facebook, BBC News because it is the world's largest broadcast news organization and it has global audiences around the world. The news posts and comments were collected using Python scraper for a two-month interval: 10th Dec 2017-20th Feb 2018, and Facebook Graph API Explorer was used to access the token and page id of the BBC News on Facebook. We gathered a total of 362 news posts and 398476

comments. Since the dataset is noisy, we filtered out some comments: those not in English, posts or comments that contain only pictures or videos, and comments with a length of fewer than 2 words. Filtered data used in this study contained 362 posts and 312291 comments. Our dataset is a bit large compared to those of previous studies on identifying related/unrelated content in social media [3], [16].

B. Pre-processing

Before extracting features from posts and comments, they must be pre-processed. We eliminate comments with fewer than two words and all non-English texts. Since stop words such as a, the, etc., do not have much meaning in our application, we remove them from all post and comments. All post and comment sentences are tokenized to words, and then the lemma for each word is derived by using the NLTK package in Python.

C. Feature description

The three categories of features used in this study are shown in Table I. To the best of our knowledge, this is the first time that we are using the combination of both bag of words-based and word embedding-based similarity measures to estimate similarity between post and comments as short texts without including the entire story of the post or external corpus related to the post itself. Among all features shown in Table I, three word embedding-based features: Google-word2vec, GloVe-word2vec, and Native context-word2vec are proposed as new features based on post and comments corpus. We examine these features to determine the similarity between a post and comments following it. Here, we use the native context of a post [3] as a set of one post and all comments following the post, and try to consider not only the pair of post and comment but also to pair a comment and all comments following a post, since these comments are more likely to be similar to each other in terms of language and topics. We also consider the native context of all posts as a corpus and employ some models like the LF-LDA and word embedding to capture the context-dependent semantics from short comments. According to Table I, these different similarity measures are described next.

TABLE I
FEATURE SETS OF THE PROPOSED FRAMEWORK

Syntactical	Topical	Semantical	
Cosine	Latent Feature-Latent Dirichlet Allocation (LF-LDA)	**String-based**	**Word Embedding (Word2Vec)**
Native context		WordNet	Google-word2vec GloVe-word2vec Native context-word2vec

Syntactical similarity: The Syntactical similarity is a measure of the degree to which the word sets of two given sentences are similar. Commentators discuss a post in the comment section, and their comments can be lexically similar to the post or similar to other comments following the post. To capture these kinds of similarities we use Cosine and Native context similarities as follows:

1) *Cosine similarity*: by considering each pair of a post and following comment as P_i and C_{ij}, Cosine similarity calculates the similarity between P_i and C_{ij} by measuring the cosine of angle between the term frequency-inverse document frequency (tf-idf) vectors of P_i and C_{ij} determined according to the bag of words approach.

2) *Native context*: by defining all comments following a post and the post itself as NC_i (native context), the similarity between each comment C_{ij} and post P_i or other comments following the post is formulated as:

$$similarity(C_{ij}) = cos(m(NC_i), C_{ij}) = \frac{m(NC_i) \cdot C_{ij}}{\|m(NC_i)\| \, \|C_{ij}\|} \quad (1)$$

According to Equation 1, a tf-idf matrix of the post and all following comments is created. Then for each comment, the cosine similarity between its vector and the mean of other native context vectors is calculated to capture the comments similar to the native context.

If each of the above syntactical similarity functions is applied to two semantically related sentences with different lexical terms, the similarity score will be zero because they cannot capture the semantics in the sentences. Therefore, we consider topical and semantical approaches based on word embedding to include semantic in our model.

Topical similarity: Comments can be related to posts in terms of different topics, which are common between posts and comment and that commentators discuss. One of the most frequently used methods to investigate how short texts are similar in terms of topics is LDA. The LDA models each document as a probability distribution over topics, and each topic as a probability distribution over words based on the co-occurrence of words within documents via tf-idf matrix. Thus, for short documents in a small corpus, LDA results might be based on little evidence and so external corpuses such as Webpage search results or Wikipedia content must be used to improve the topic representations [16], [24]. To deal with this challenge in our study, we use LF-LDA [21] to make topical similarity detection more efficient by leveraging both a latent feature trained on a large corpus and the topic modeling method. In the following, we describe both the LDA and LF-LDA models and explain how we adapt them to identifying related/unrelated content.

1) *Latent Dirichlet Allocation (LDA)*: for each post P_i, we apply the topic model LDA to learn the topics from all the comments in native context C_i. LDA assumes that each document has a probabilistic multinomial distribution θ over latent topics, where each topic is characterized by a probabilistic multinomial distribution φ over the words. Both the topic distribution in all documents and the word distribution of topics share a common Dirichlet prior [10], [25]. By assuming α as the parameter of the Dirichlet prior on the per-document topic distribution (θ) and β as the parameter of the Dirichlet prior on the per-topic word distribution (φ), two distributions θ and φ can be given by:

$$\theta = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T\alpha} \quad (2)$$

where D and T stand for documents and the number of topics, respectively. C_{dj}^{DT} is the number of occurrences of terms in document d that have been assigned to topic j, and

$$\varphi = \frac{C_{ij}^{WT} + \beta}{\sum_{k=1}^{W} C_{kj}^{WT} + W\beta} \quad (3)$$

where W and T stand for the number of terms and topics, respectively. C_{ij}^{WT} is the number of times that term i has been assigned to topic j. To estimate LDA parameters more accurately, we use the Gibbs sampling approximation method. Here we consider each post and all the following comments as documents D and find 8 topics that are discussed in the comments. After training the LDA on native context C_i, we estimate the topical similarity between each post P_i and its following comment C_{ij} by applying the Jensen-Shannon divergence metric based on the Kullback-Leibler (KL) divergence. Since Jensen-Shannon is a measure of the distance between two probability distributions, we consider the topic distribution over each post and comment as P and Q and calculate their similarity using the following function:

$$JSD\,(P, Q) = \frac{1}{2}\left(D_{kl}\,(P, M) + D_{kl}\,(Q, M)\right) \quad (4)$$

$$D_{kl}\,(P, M) = \sum P\,(i)\, log \frac{P\,(i)}{M\,(i)} \quad (5)$$

where $M = 1/2(P + Q)$.

2) *Latent Feature-Latent Dirichlet Allocation (LF-LDA)*: is a probabilistic topic model that combines a latent feature model with an LDA model. Recently, neural network methods have been used to learn and represent words as vectors in real numbers, known as word embedding. These vectors have latent features that capture the context of a word in a document, its semantics, and relation with other words [26]. The word2vec model is one of the most famous word embedding models [11]. Based on this vector representation, Dat et al. [21] proposed the LF-LDA model to go beyond LDA for topic modeling. In LF-LDA, the Dirichlet multinomial distribution for topic-to-words has two components: a topic-to-word Dirichlet multinomial component and a latent feature component. This model can perform well on corpora with few or short documents compared to the LDA's requirements. Here we use a pre-trained word2vec model named Google Word2Vec, which is trained on a 100 billion word subset of the Google News corpus [11]. For each post P_i, we apply the topic model LF-LDA to learn topics from the post and all its comments. We eliminate each word from the post and comments that is not in the pre-trained word2vec models. To estimate the topical similarity between each post and comment through the LF-LDA learned topic-to-word distributions, we use Jensen-Shannon divergence defined in equation 4.

Semantical similarity:

1) *WordNet-based*: WordNet is one of the earliest methods for extracting semantic similarity or relatedness between a pair of concepts or word senses. It is a large lexical database of English words including nouns, verbs, adjectives, adverbs, etc., and their sets of cognitive synonyms. Since WordNet contains information on nouns, verbs, adjectives, and adverbs, we use Part Of Speech (POS) tagging on each post and comment pair and then find semantic similarity between them by WordNet using NLTK package in Python.
2) *Word Embedding-based*: We use three word embedding methods to capture the semantic similarity between a post and its comments. We use a combination of pre-trained models Google Word2Vec [11] and Stanford GloVe [23] and we also train a model based on the comments corpus in our dataset. A brief comparison between the effect of these vector-based word representation methods is presented in the Experiment Section.
   a) *Google-word2vec*: Google word2vec is a word embedding model pre-trained on the Google News corpus. Every two words that are similar in context and semantics will tend to have more similar (close) feature vectors of real numbers to each other [11]. For each word in a post, its vector representation with 300 dimensions is extracted from the Google News corpus pre-trained model. The average value is then calculated among all vectors as a 1*300 dimension vector. This process is repeated for each comment. Finally, the cosine similarity between the post and comment vectors is calculated as a word embedding similarity measure between them. For two documents d_1 and d_2 as post P_i and comment C_{ij}, word embedding similarity (WESim) between post and comment is defined as follows:

$$WESim\left(d_1, d_2\right) = Cosine\left(\frac{\sum_{j=1}^{|W_{d_1}|} v_j(d_1)}{\left|W_{d_1}\right|}, \frac{\sum_{k=1}^{|W_{d_2}|} v_k(d_2)}{\left|W_{d_2}\right|}\right) \tag{6}$$

   where $v_j\left(d_1\right)$ and $v_k\left(d_2\right)$ are vector representations of jth and kth word in document d_1 and d_2 respectively. $|W_{d_1}|$ and $|W_{d_2}|$ are the number of words in d_1 and d_2 respectively. Here we remove the words in post and comments that do not exist in the pre-trained Google News corpus.
   b) *GloVe-word2vec*: In word embedding based models, the corpus used for training vectors is an important issue, as the meaning of the vector representation of words will be different depending on the context and the semantics of the corpus in which words are represented. Therefore, we include the GloVe word embeddings pre-trained model in addition to the Google News corpus to see how a corpus can be effective in applying word embedding similarity measures to identify related/unrelated content. The GloVe vectors were trained from 840 billion tokens of Common Crawl web data and have 300 dimensions [23]. This feature is extracted similar to the Google-word2vec similarity by using equation 6 for each post and comment pair.
   c) *Native context-word2vec*: We considered all the posts and following comments in our filtered BBC News dataset to train a word embedding model (word2vec model) named Native context-word2vec. To extract word embeddings, we trained a neural network with a single hidden layer in our corpus, so that the weights of the hidden layer will be vector representation of words according to the word2vec approach in [11]. We used the Gensim library in Python to train our word2vec model with the Google Word2Vec toolkit. The word embedding similarity between each post and comment pair can then be estimated with equation 6.

IV. Evaluation and Results

As our model consists of several features, first we conducted experiments by comparing our model to baselines that only apply one feature or that combine fewer features to test the necessity of combining these features. We also compared our model with LDA as a most frequently used method for topic detection in previous studies [2], [16] to investigate the effect of using LF-LDA in comparison with LDA. Finally, to evaluate the performance of our model in comparison with previous studies, we used a proposed model by Xie et al. [3]. Toward that end, first, we annotated 10% of posts with all their following comments as training data. Then following [16], [17], [24], we used the scikit-learn's implementation SVM algorithm for learning a binary classifier on the training dataset.

A. Gold standard annotations

We sampled 10% of all posts (30 posts from 362 posts in our collected data) with all their comments using Stratified random sampling that branches off the entire dataset into multiple non-overlapping, homogeneous subgroups and randomly chooses final members from the various subgroups as train dataset. In accordance with the distribution of comments (max and min number of comments, mean of all comments, and standard deviation) following all the posts, we observed that 5% of the posts have fewer than 164 comments and 5% of them have more than 2,766, therefore, we chose the fifth and ninety-fifth percentiles as criteria to create three subgroups. Table II lists the breakdown of the sampled posts.

TABLE II
Data sampeling

	Posts	#Sample
Subgroup 1	#comments<164	2
Subgroup 2	164<#comments<2766	26
Subgroup 3	#comments>2766	2

The sampling method produced 33,921 pairs of post and comments. We define comments in which commentators are discussing the topic of a post or the topic of other comments following that post which are similar to the post's topic as related comments. These types of comments offer arguments and are similar to the post's content and therefore give readers some potentially good information. On

the other hand, comments that contain contents merely to attract a reader's attention and do not have useful information are considered as unrelated comments. We have defined some main clues to select unrelated comments: 1) Comments with advertising contents referring to websites, companies, or to a product advertising mechanism in social media. For example, using commercial URLs without any textual data or with texts that are unrelated to a post's content. 2) Comments with very little contents, that are very brief and without words in common with a post's content. This category includes comments that just show a commentator's sentiment in reaction to a post, such as "I love this" or "I hate that" and do not give readers any additional information related to the post content. 3) Comments in which commentators are only arguing with each other without discussing the topic of a post. These kinds of comments usually do not have a common context with the post. 4) Comments in which commentators are giving their opinion about a news agency page on Facebook and not about a post's content. Due to the high diversity of contents in Facebook [16] we considered these kinds of contents as unrelated and defined these clues to have a unique definition for labeling the train data.

The corpus was annotated by five graduate students as follows: First, two annotators conducted a labeling process of two separated sets of 15 posts (among 30 sampled posts) and all their following comments. Next, 3000 pairs of posts and comments, which were annotated before, were randomly sampled and given to three other annotators to annotate again. Finally, the accuracy of the labels annotated by the first two annotators was estimated based on the three other labels. We selected a label for each sample (3000 pairs of posts and comments) using the majority vote among the three annotators' labels and then compared that label with the first two annotators' labels. This comparison results in a 6.2% error rate, which shows the annotation process achieved a high level of trustworthiness. Therefore, we considered the first two annotators' labels as gold standard labels of the training corpus in the rest of the paper.

B. Experiment results

After extracting the features, they are taken to the SVM classifier. The average accuracy, precision, recall, and F-measure are calculated based on k-fold cross-validation (k=10) to evaluate the quality of the classifier. The results are showed in: 1) the impact of combination of features in the efficiency of the model, 2) the effect of combining word embeddings with topic modeling method in identifying related/unrelated content when we have short texts within small corpus, and 3) the performance evaluation of the proposed model in comparison with Xie et al. [3] approach.

The performance metrics evaluation is reported in Table III, in which it is shown that the proposed model with a combination of all features obtains 86% accuracy on average and it outperforms all other combination of features. We analysis classification results by eliminating each category of features and it indicates that eliminating the syntactical category has a small effect on reducing the accuracy of the model (W/O Syntactical column in Table III). The accuracy of the model without syntactical features is 85% because these features can not capture related words with different lexical context and semantics of context in which words are represented. On the other hand, eliminating the semantical category (W/O Semantical column in Table III) has the most effect on the accuracy of the model. The accuracy of the model without the semantical category will be 74% because these features play the main role in including context-based semantics to the model especially by using the word embedding method. Eliminating topical category has also effect on the efficiency of the model since the accuracy reduces to 84% when the topical feature is eliminated.

TABLE III
PERFORMANCE OF DIFFERENT FEATURE COMBINATIONS

	Accuracy	**Precision**	**Recall**	**F-Measure**
All features	86.1	85.5	84.4	84.9
W/O Syntactical	85.3	85.4	83.5	84.4
W/O Topical	84.3	85.7	84.5	85.0
W/O Semantical	73.9	65.6	75.1	70.0
Just Syntactical	60.3	64.3	74.8	69.1
Just Topical	64.6	54.3	64.0	58.7
Just Semantical	82.4	85.4	83.4	84.3

W/O = exclude one kind each time; Just = include one kind each time

To show the necessity of combining three categories, we examine the effect of each category alone in identifying related/unrelated comments following a post too. From Table III it is obvious that using syntactical features only is not efficient in this problem because cosine and native-context similarities are incapable of matching a post with a comment if they have related meanings but different terms. Even applying only topical feature results in low accuracy. Among three categories, just semantical features give the high accuracy of 82.4% in identifying correct labels for each comment whereas it is still capable to be increased by involving other categories (all features).

LDA vs LF-LDA: To the best of our knowledge, we are the first to propose a combination of topical and word embedding-based approaches in identifying related/unrelated comments following a post on social media. Therefore, we examine the efficiency of our model with the LDA [10] as a baseline, which has been used in previous studies to find topical similarity between texts, and LF-LDA along with semantic-based features. According to our experiments, we set hyper-parameters α and β in both LDA and LF-LDA to 0.1 and 0.01 and the number of topics to 8. For Native context word2vec the window size and embedding vector dimension are set to 5 and 300, respectively, and words with a frequency of less than 2 are eliminated. Table IV shows the classification results using LDA or LF-LDA with the semantical category. Although syntactical features make a little bit of change in the accuracy of the proposed model, based on Table III, we do not consider it in the rest of the analysis.

TABLE IV
IMPACT OF COMBINING A TOPICAL APPROACH WITH WORD EMBEDDING ON IDENTIFYING RELATED/UNRELATED CONTENTS

	LDA + Semantical	LF-LDA + Semantical
Accuracy(%)	82.7	85.3
Precision(%)	81.1	84.4
Recall(%)	84.0	83.5
F-Measure(%)	82.5	84.4

Syntactical features are not considered.

The results show that LF-LDA can outperform LDA in combination with semantical features. The accuracy results from LDA along with semantical features is 82.7% whereas this value is 85.3% for LF-LDA among with semantical features. Because LF-LDA uses latent features resulted from Google word2vec pre-trained model to provide more sufficient information for topic distribution modeling. Therefore in LF-LDA, the coherence between topics is more than LDA and more context-based semantic is included in the model through latent feature vector of words. Considering that we do not have access to the entire story of a post and any external web pages related to the post content specifically, LDA trains topic distributions based on our existing corpus. Whereas, LF-LDA uses a pre-trained model (here, Google Word2Vec) to leverage the latent feature vector of words for improving the topic distributions learned from our existing corpus.

Word embedding based features: We are using Google-word2vec, GloVe-word2vec, and Native context-word2vec in the semantical

category. To see the effect of each word embedding methods in the accuracy of our model, we eliminate each of them from the set of features and evaluate the accuracy of the model. The result of this experiment is given in Table V.

TABLE V
IMPACT OF PRE-TRAINED AND NEW WORD EMBEDDING MODELS ON IDENTIFYING RELATED/UNRELATED CONTENTS

	Accuracy (%)
W All word embedding methods	86.1
W/O Google-word2vec	69.2
W/O GloVe-word2vec	74.3
W/O Native contex-word2vec	80.1

W: include all word embeddings; W/O: exclude one kind each time

According to Table V, using pre-trained Google word2vec model gives the highest accuracy among all word embeddings approaches because eliminating it from the set of features reduces the accuracy to 69.2% where eliminating GolVe word2vec pre-trained model reduces the accuracy to 74.3%. It shows that feature vector of words in pre-trained Google word2vec model have more context-based semantic to words from our existing corpus and it produces high quality word embeddings. We use posts and comments related to the BBC News agency page on Facebook and they have more common context and words with Google News corpus which is used to train Google word2vec model. Therefore, eliminating this feature has a negative effect on capturing semantic between posts and comments and reduces the accuracy of the model. On the other hand, eliminating Native context word2vec has the lowest effect on the accuracy because our corpus, posts and all comments, is small and provides insufficient information for word2vec training model to extract the underlying feature vector of words robustly. By using Native context word2vec we can alleviate missing words from two previous pre-trained models because Native context word2vec model trains a feature vector for each word in the corpus according to its context and semantic.

Previous research: Xie et al. [3] proposed a model to derive context-dependent (i.e. context-aware) semantics of short comments and detect short irrelevant texts. They leveraged both native context and transferred contexts, the neighboring comments on a specific topic instead of all comments in the corpus, based on LDA topic similarity between articles and following comments. To compare our model with this study, we crawled the entire story of each post in train dataset from the BBC news agency webpage and applied context-aware approach proposed in [3], the results are shown in Tabel VI.

TABLE VI
PERFORMANCE METRICS EVALUATION IN DIFFERENT APPROACHES

	the proposed method	Xie et al. [3]
Accuracy(%)	86.1	76.5
Precision(%)	85.5	74.0
Recall(%)	84.4	77.8
F-Measure(%)	84.9	75.8

From VI, we observe that our proposed method performs better in terms of evaluation metrics. As context-aware approach proposed by Xie et al. [3] represents comments and the whole content of the post just by building vectors based on term frequencies and then applies matrix factorization to build topics, they can not include the semantic behind the related but different words in their model. Therefore, it causes to lower precision and recall. Especially lower precision in Xie et al. [3] approach shows that using LDA alone without word vector embeddings extracted from semantic relation between words in both total comments and pre-trained word embedding models, leads to more false positive rate in identifying related/unrelated comments.

C. Case study

We apply the learned classifier on the rest of our dataset (278,370 pairs of posts and comments) to predict their labels as *related/unrelated* comments written to the post. The classifier's result shows 41% of all comments are related and 59% of them are unrelated. This is an interesting observation that shows around 60% of the written comments to the posts in a news agency account are not related to the actual post in terms of the topic of discussion. This huge number of unrelated comments potentially biases a lot the readers perspective on the posts and provides a large noise on the available users' feedback. By analyzing the distribution of *related/unrelated* comments across the posts, we observed that news posts containing a specific action or speech of popular people in a specific time have more unrelated comments than the posts which are announcing a fact or telling a story of daily events.

To investigate how the content of related and unrelated comments are different from the topic of the posts, how they are spreading during the lifetime of posts, and how they are similar to each other we analysis 4 randomly chosen posts with all their comments (after applying the learned classifier) as follows:

Content analysis of written comments under a post: To understand better the relation of written comments to the posts, we sampled 4 posts randomly and investigated the discussed topics on each two group of identified comments (*related/unrelated*). The texts of sampled posts are shown in Table VII. Post 1 is mainly related to students, young people, and their usage of safe internet. Post 2 and Post 4 are announcing some daily events or facts and post 3 is related to a political issue. We create a word cloud from *related* and *unrelated* comments following the 4 selected posts to show which topics are more discussed among *related* and *unrelated* comments in each post depicted in Figure 2 and Figure 3.

TABLE VII
FOUR SAMPLED POSTS FROM BBC NEWS AGENCY PAGE ON FACEBOOK.

Posts	Text
post 1	School pupils read out some of the worst comments they've seen posted online for Safer Internet Day.'BBC Own It' is a new website to help young people stay safe online and navigate their digital lives with confidence.
post 2	Indian police have arrested a man who allegedly shot dead his neighbor by mistake at a pre-wedding party.
post 3	US President Donald Trump has sparked a backlash from UK politicians by attacking the National Health Service.
post 4	Who says make-up is just for girls?? South Korean men spend more on beauty and skincare than anywhere else in the world. Take a look at their quest to challenge beauty standards.

By considering the word cloud from *related* comments shown in Figure 2, it is obvious that users are discussing explicit subjects related to the topics of each post. For example in post 1, the most frequently used words in *related* cluster are "children", "Kid", "parent", "school", "internet", "bullying", "social media", etc. which are mainly discussing the topic of post 1 and they give readers significant information related to the post. Or in post 2, people are using words such as "people", "Indian", "gun", "celebration", "wedding", "culture" and etc. in their comments. For post 3, the words in larger size such as "Trump", "NHS", "people", "government", "healthcare", "insurance", "hospital", etc. are closely related to the topic of post 3. Finally, in the word cloud of *related* comments written under post 4, users are using "men", "women", "makeup", "wear", "look like", etc. words more frequently in their comments to discuss the topic of post 4. Since readers are more interested in reading strictly on-topic

information from the comment section, filtering the *related* cluster for each post can be very useful and informative to users.

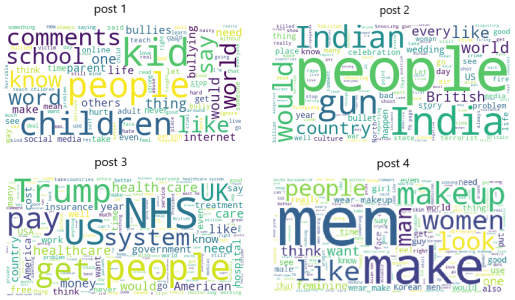

Fig. 2. WordCloud of *related* comments following the sampled posts; the more important a word makes the larger its size.

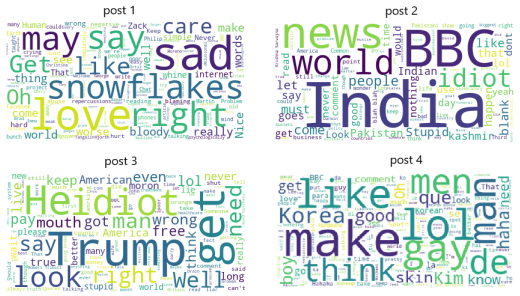

Fig. 3. WordCloud of *unrelated* comments following the sampled posts; the more important a word makes the larger its size.

By investigating the word clouds from *unrelated* comments of the four sampled posts in Figure 3, we observed different kinds of unrelated comments written under the posts. For example words such as "love", "sad", "right", "wrong", "worse", "oh", "idiot", "stupid", "lol", etc. are more frequently used words in *unrelated* comments. This observation shows that users are mostly expressing their opinion or point of view related to the posts' entities (here India, Trump, Korea as the posts' content are mostly about them) or other comments written by users which do not have significant information for readers because they do not discuss the topic of posts. Another interesting observation is that some most frequently used words such as "snowflakes" in post 1 are completely far from the topic of the post and they come from unrelated comments such as advertisements or bot-generated contents. For example in post 1, we observed that there are some comments in *unrelated* cluster that were advertising about "Amazing Macro Photographs of Snowflakes". On the other hand in post 2, a lot of comments are targeting BBC news agency in Facebook since the words "News" and "BBC" are one of the most frequent used words in the word cloud from *unrelated* cluster.

Analyzing the content of *related/unrelated* written comments under the posts shows that most of *related* comments are objective and more topically coherent with posts' content in terms of topics whereas *unrelated* comments usually contain subjective and very general words expressing users' feedback without any focus on the subject of the posts. In *unrelated* cluster the most frequent words are not mainly related to the posts' topics and commentators are generally discussing similar topics which show personal feelings or opinions, or they are arguing about news agency itself. There are also some completely unrelated comments under posts that may be generated by users or bots for advertising or spreading information across different posts on Facebook that our model could identify them correctly. Since this type of comments are not informative and maybe readers are not interested in reading such off-topic information, it is better to identify and filter out these unrelated comments.

Timestamp analysis of written comments under posts: To see how users are disseminating *related/unrelated* comments under posts, we first look at the distribution of *related/unrelated* comments within a period of 24 hours after publishing each post (on the rest of our dataset: 278,370 pairs of posts and comments). For each comment following a post, the difference between a timestamp when the post was uploaded and the timestamp of the written comment is considered. Figure 4 depicts the portion of *related/unrelated* comments written under posts within the first 24 hours. It is evident that the portion of *unrelated* comments written under all posts, in general, are more than *related* one in the first hours after publishing posts however the number of written comments under each post are diverse and we can not say that this evidence is true through all posts.

To go more deeply into this subject and see how *related/unrelated* comments are spreading per post, we look at the portion of *related/unrelated* comments following each sampled post based on their written time within a period of 12 hours. Figure 5 depicts the portion of *related/unrelated* comments written under posts during the first 12 hours after each post creation time.

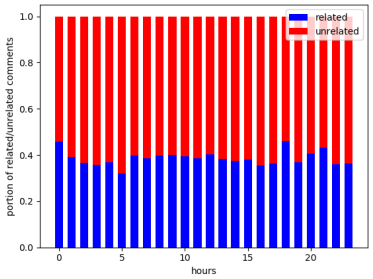

Fig. 4. Distribution of *related/unrelated* comments following all the posts within a period of 24 hours.

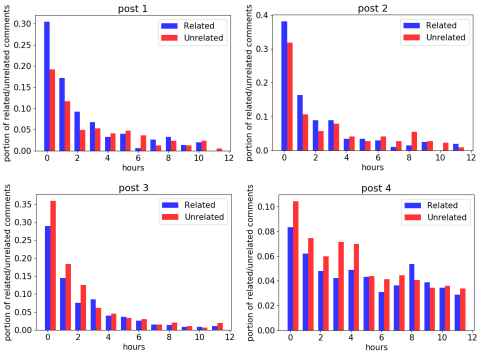

Fig. 5. The portion of *related/unrelated* comments written under 4 sampled posts within the first 12 hours.

As it is obvious from Figure 5, there is not a specific pattern among all posts in spreading *related/unrelated* comments. However, an interesting observation is that in some posts such as post 1 and post 2, the portion of *related* comments are more than *unrelated* comments in the first hours. Then by passing the time, the portion of *unrelated* comments increases. Whereas, in post 3 and post 4 the portion of *unrelated* comments are more than *related* comments over the period of 12 hours. By considering the text of sampled posts (Table VII), it can be inferred that the topic of a post plays an important role in the content of the following comments. For example, the topic of two first posts are about a scientific context or daily event, commentators are more discussing the topics in the first hours. Whereas in the two last posts, commentators are posting *unrelated* comments more than *related* comments in the first hours since the topics of post 3 and

post 4 are more attractive to different users in terms of topics; they are related to politic and gender issues. A lot of users come to these hot topic posts to just show their feeling by putting uninformative comments or attract other users' attention by putting advertisements or off-topic comments to the post.

Similarity within related/unrelated comments: Another aspect that we aim to study is to understand the similarity degree of comments inside *related*/*unrelated* clusters. To see how similar a comment is to other comments following a post, based on word feature vector similarity, we extract comments with a degree of similarity more than 90% to another comment following the same post. The result shows that only 0.4% of comments in *related* cluster and 0.7% of comments in *unrelated* cluster have degree of similarity more than 90% to at least another comment. To go deeper into details and see when these similar comments are published, we explore *unrelated* comments in sampled posts. In average 0.8% of *unrelated* comments in the sampled posts are similar to each other with a degree of similarity more than 90%. By checking these types of comments, we find that they are frequently duplicate comments posted by users within a duration in seconds. In addition, they are also short texts with common words. Since the number of these types of comments are very low, in general, they cannot be generated for a specific purpose by bots. It can be inferred that users are posting this kind of duplicate content to emphasize their feedback and feeling or it happens during the commenting process in social media with their faults.

V. Conclusion and Future Works

We build a model to identify related and unrelated comments to the corresponding posts on Facebook by considering the content of the comments. The framework consists of three categories of features: syntax, topic, and semantic. To be independent of the entire story of a post or external webpage contents related to the post, we use combination of word embeddings in both topical and syntactical features. The results show that the model can identify related/unrelated comments written to the posts with more than 85% accuracy. We next investigate the distribution of the related/unrelated comments across the posts and also look to the main discussed topics in each cluster. This provides a better understanding of the phenomena of unrelated comments in social media. In future, we will include analysis of the portion of related/unrelated comments across different categories such as politicians, celebrities, and companies in Facebook. We will also dig into the unrelated comments according to diversity of contents and will try to find machine generated comments in *unrelated* content. Finally, we would like to study this phenomena across social media e.g. Instagram and twitter.

References

[1] P. Rajapaksha, R. Farahbakhsh, N. Crespi, and B. Defude, "Inspecting interactions: Online news media synergies in social media," *CoRR*, vol. abs/1809.05834, 2018. [Online]. Available: http://arxiv.org/abs/1809.05834

[2] J. B. Houston, G. J. Hansen, and G. S. Nisbett, "Influence of user comments on perceptions of media bias and third-person effect in online news," *Electronic News*, vol. 5, no. 2, pp. 79–92, 2011.

[3] S. Xie, J. Wang, M. S. Amin, B. Yan, A. Bhasin, C. Yu, and P. S. Yu, "A context-aware approach to detection of short irrelevant texts," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2015, pp. 1–10.

[4] A. Zhang, B. Culbertson, and P. Paritosh, "Characterizing online discussion using coarse discourse sequences," 2017.

[5] N. C. Dang, F. De la Prieta, J. M. Corchado, and M. N. Moreno, "Framework for retrieving relevant contents related to fashion from online social network data," in *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*. Springer International Publishing, 2016, pp. 335–347.

[6] S. A. Salloum, M. Al-emran, A. A. Monem, and K. Shaalan, "A Survey of Text Mining in Social Media : Facebook and Twitter Perspectives," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 1, pp. 127–133, 2017.

[7] A. Kothari, W. Magdy, K. Darwish, A. Mourad, and A. Taei, "Detecting comments on news articles in microblogs," in *ICWSM*, 2013.

[8] A. Sureka, "Mining user comment activity for detecting forum spammers in youtube," *CoRR*, vol. abs/1103.5044, 2011.

[9] A. H. Wang, "Detecting spam bots in online social networking sites: A machine learning approach," in *Data and Applications Security and Privacy XXIV*, S. Foresti and S. Jajodia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 335–342.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, mar 2003.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Curran Associates Inc., 2013, pp. 3111–3119.

[12] H. Liu, J. Han, and H. Motoda, "Uncovering deception in social media," *Social Network Analysis and Mining*, vol. 4, no. 1, p. 162, Feb 2014.

[13] A. Suarez, D. Albakour, D. Corney, M. Martinez, and J. Esquivel, "A data collection for evaluating the retrieval of related tweets to news articles," in *Advances in Information Retrieval*, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds. Springer International Publishing, 2018, pp. 780–786.

[14] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "Netspam: A network-based spam detection framework for reviews in online social media," *Trans. Info. For. Sec.*, vol. 12, no. 7, pp. 1585–1595, Jul. 2017. [Online]. Available: https://doi.org/10.1109/TIFS.2017.2675361

[15] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD, 2013, pp. 632–640.

[16] J. Wang, C. T. Yu, P. S. Yu, B. Liu, and W. Meng, "Diversionary comments under blog posts," *ACM Trans. Web*, vol. 9, no. 4, pp. 18:1–18:34, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2789211

[17] G. Fei, A. Mukherjee, B. Liu, M. Hsu, and M. C. et al, "Exploiting burstiness in reviews for review spammer detection," in *ICWSM*, 2013.

[18] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," in *AIRWeb*, 2005.

[19] A. Bhattarai, V. Rus, and D. Dasgupta, "Characterizing comment spam in the blogosphere through content analysis," in *2009 IEEE Symposium on Computational Intelligence in Cyber Security*, March 2009.

[20] J. Zhu, K. Wang, Y. Wu, Z. Hu, and H. Wang, "Mining user-aware rare sequential topic patterns in document streams," *IEEE Transactions on Knowledge & Data Engineering*, vol. 28, no. 07, jul 2016.

[21] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *CoRR*, vol. abs/1810.06306, 2018. [Online]. Available: http://arxiv.org/abs/1810.06306

[22] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08, 2008, pp. 91–100.

[23] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[24] X. H. Phan, C.-T. Nguyen, D.-T. Le, M. L. Nguyen, S. Horiguchi, and Q.-T. Ha, "A hidden topic-based framework toward building applications with short web documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 961–976, 2011.

[25] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[26] Z. Bouraoui, S. Jameel, and S. Schockaert, "Relation induction in word embeddings revisited," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 1627–1637.