Probabilistic Relational Supervised Topic Modelling using Word Embeddings

Jabir Alshehabi Al-Ani
School of Computer Science and Electronic Engineering
University of Essex
United Kingdom, Colchester, CO4 3SQ
IEEE Member
Email: jajals@essex.ac.uk

Maria Fasli
Institute for Analytics and Data Science
School of Computer Science and Electronic Engineering
University of Essex
United Kingdom, Colchester, CO4 3SQ
Email: mfasli@essex.ac.uk

***Abstract*—The increasing pace of change in languages affects many applications and algorithms for text processing. Researchers in Natural Language Processing (NLP) have been striving for more generalized solutions that can cope with continuous change. This is even more challenging when applied on short text emanating from social media. Furthermore, increasingly social media have been casting a major influence on both the development and the use of language. Our work is motivated by the need to develop NLP techniques that can cope with short informal text as used in social media alongside the massive proliferation of textual data uploaded daily on social media. In this paper, we describe a novel approach for Short Text Topic Modelling using word embeddings and taking into account any informality of words in the social media text with the aim of addressing the challenge of reducing noise in messy text. We present a new algorithm derived from the Term Frequency - Inverse Document Frequency (TF-IDF), named Term Frequency - Inverse Context Term Frequency (TF-ICTF). TF-ICTF relies on a probabilistic relation between words and context with respect to time. Our experimental work shows promising results against other state-of-the-art methods.**

Keywords: *Topic Modeling,Term Frequency, Embeddings, TF-IDF, Short Text, Words Matching*

I. Introduction

Short text analysis is increasingly becoming more challenging given the rapid changes in the language. Social media play a major role in language development and evolution and are contributing to the fast pace of change. Meanwhile, the Oxford English Dictionary (OED) [1] has noted this change and started updating its database on a quarterly basis with an average 1000 words every three months. This provides clear evidence of the significant and rapid changes in language. Some words may quickly spread and tend to become adopted on social media and beyond – *trending* – and gradually become standard words that have meaning. For instance, hash tags written by anyone just to indicate the importance of a specific subject or event have become popular on social media.

Words co-occurrence and relations are the alternatives for the state of the art algorithms like Latent Drichlet Allocation (LDA) [2] and the old text mining techniques as described by [3]. Meanwhile, Chen and Kao [4] used the words co-occurrence for short text topic modelling. They presented a new approach for topic modelling in the Chinese language as it does not contain any break words. Their approach shows good results on news titles. Similarly, Lu et al [5] trained their proposed approach on news titles and Q&A questions. They adopted the relations between words as a main concept which has shown its effectiveness. They used Biterm [6] to construct their word co-occurrence matrix alongside Inverse Document Frequency (IDF) [7] to classify the news titles according to the main related topics.

In contrast, a generalized approach on social media short text analysis especially twitter is rare. The first example that shows good results depends on the author's topic. Lim et al. [8] presented a novel approach named Twitter Network (TN) to relate tweets to topics depending on a full Bayesian Treatment of the documents. These network relations might change according to co-occurrences of words over time, which cause less accurate results. Likewise, Biterm [6] is a novel approach using the Bi-Relation between terms to create a words representation with a Gibbs Sampling [9] as a probability distribution algorithm. The results show good performance on linking news tweets to new titles compared to the LDA state-of-the-art algorithm.

In this paper, we develop a novel approach on supervised topic modelling to classify tweets to related topics. Words embeddings have been constructed using words co-occurrence frequency differences with respect to time. Accordingly, we track the change in the words co-occurrence frequencies over time and calculate the features vector. The developed method which is derived from TF-IDF, named TF-ICTF converts words to vector spaces. In addition, words matching will be presented specifically to reduce the informal words and link them to the formal standard words through context and syntax similarity.

The rest of the paper is structured as follows. First we present the previous work that has been produced in topic modelling especially on short text. Then, the description of the methods developed follows including our motivation and analysis that have led to the development of TF-ICTF. The experimental set-up and results are discussed in the next section including the used datasets. Finally, the paper ends with the conclusions and avenues for future work to further extend our approach in addressing the current limitations.

II. Previous Work

Research on short text collected from the social media especially twitter is very popular nowadays. The rich information presented in the social media postings make it a popular topic to study, but challenging at the same time because of the messy and noisy nature of the social media posts like tweets. One of these works presented a novel approach including a self-aggregation process which is included within the topic modelling [10]. The good results shown were for Yahoo answers which are less likely to have noise or be messy as text. Therefore, the results are more favourable when compared against other similar topic models. Likewise, an empirical study [11] addressed topic modelling on the social media by using standard topic modelling techniques. The proposed approach was demonstrated on two real world classification problems and obtained good results when the tweets were aggregated depending on the user profile. This might have the drawback of directing most of the process according to the user's interests or finding the users who could fit in the topic. Therefore, it can not be considered as a generalized approach for topic modelling.

On the other hand, an unsupervised topic modelling scheme was produced by Sridhar [12] to cluster similar tweets. The approach shows better results in topic modelling than the LDA [2]. Chenliang Li et al [13] included auxiliary embeddings alongside the Generalized Polya Urn-Dirichlet Multinomial Mixture(GPU-DMM). GPU-DMM is a mix between a neural network and a GPU sampling process. The proposed model was applied on two standard datasets of Q&A and web snippets which are clean short text data. Compared to the Biterm model [6], the GPU-DMM shows better results but the Biterm model is more likely to be applied on noisy data like twitter.

Finally, the previously described works show very good results on topic modeling but we have noticed that the topics were uncorrelated which is one of the drawbacks of the well known algorithms like LDA. The topics that will be shown in this paper were selected by annotators then aggregated in one data set to train our proposed model on using supervised machine learning technique. In addition, the approaches previously developed are not applied on raw twitter data, but very often they are supported by additional data sources – auxiliary or supportive data – to train the model.

III. Methodology

Probabilistic Topic modeling for short text is the focus of this paper. The data used in our research are Twitter data which are classified as messy noisy short text by many researchers like Lim et al. [8], and Chen and Kao [4]. They are also very rich in information that motivates researchers and companies to mine them. This section will be divided into several subsections which are: the motivation, developed methods from the state of the art, and other used methods.

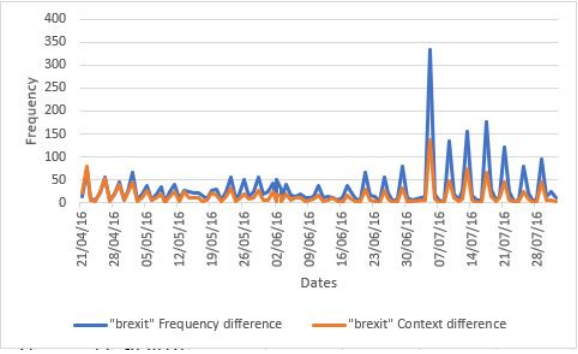

Figure 1: "brexit" Word frequency difference vs Context frequency difference

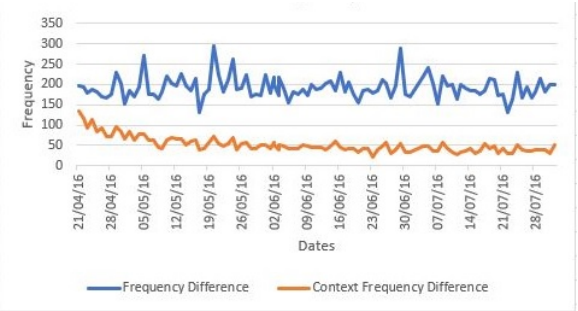

Figure 2: "car" Word frequency difference vs Context frequency difference

A. Words and Context frequency analysis

Recent probabilistic approaches like Self-Aggregation Topic Model (SATM) [10], pseudo-document-based Topic Model (PTM) [14] [15], and Sparsity-enhanced PTM (SPTM for short) [16] use words embeddings or pseudo-Documents for topic modeling with respect to words frequencies and context relations. These relations known as the words co-occurrence patterns that shows the likelihood of the word's occurrence with similar words within the same context(i.e. similar words co-occurrence) on many short text documents (tweets). Moreover, we could not find any obvious study that could of word-context co-occurrence changes over time for social media short text. Before we describe our model, we have studied these changes over time by calculating the frequency change with respect to time for the words as well as the context pattern co-occurrence over the time. A time window of three months from a twitter data had been selected as could be shown in the figures 1, 2, and 3.

We were inspired by the analysis on twitter data based on this word-context relation but with time as an additional factor. In figures 1 and 3, the difference between any day's frequency for the word "brexit" for example for the consequent chosen three months is very close to the difference of the context words frequency. Additionally, the ratio between both context and word frequencies more likely to be nearly coherent. This gives an indication of having similar context words for "brexit" on any tweet. These kind of words are more likely to be

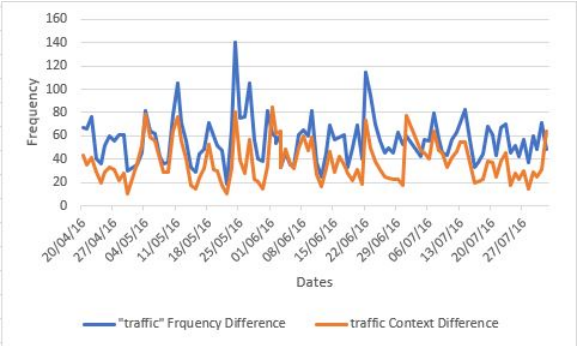

Figure 3: "traffic" Word frequency difference vs Context frequency difference

Dates	"brexit" frequency	Context word Frequency	Word frequency difference	Context-word frequency difference
01/05/16	285	287	-	-
02/05/16	350	332	65	45
03/05/16	355	335	5	3
04/05/16	373	347	18	12
05/05/16	410	373	37	26
06/05/16	416	379	6	6
07/05/16	436	390	20	11
08/05/16	471	409	35	19
09/05/16	472	409	1	1
10/05/16	495	423	23	14
11/05/16	535	448	40	25
12/05/16	543	451	8	3
13/05/16	569	476	26	25
14/05/16	592	488	-	-

TABLE I: Two weeks of Word frequencies and context-words frequencies

considered in a tweet when compared to other word such as "car" as shown in figure 2 . Hence, the word "car" will be removed from the evaluation process for any tweet before converting it to features because it shows less coherence between its context and words frequencies. Where figure 2 shows the orange curve that represents the context frequency difference for the word "car" that is described as dying. On the other hand, the blue that represent the word's frequency difference shows consistency over 200. Thus, less words will produce less cost as an additional criterion for the proposed model.

Table I shows 14 days of the word"brexit" frequencies on the word frequency column. The second one shows the words of context frequencies co-occur with"brexit" on the same tweets per day. For example, we will consider the first "brexit" word in the tweet " I m so sick of hearing Brexit means Brexit". The words "hearing" and "means" are occurring on both sides of the "brexit" are the context words. These words will not be counted on the context frequency if occurred again with "brexit" as it will be counted only once. So, the differences between each successive cells on the same column are shown in the columns of context and words frequencies differences. We took these differences in to account as it shows the relation between both context and word frequencies as

Anchor Words	Removed Words
traffic brexit weather rain stuck shopping stadium sunny football	the more london car new tfl

TABLE II: Anchor words and Removed words

described previously in figure 1. Moreover, the words' context frequency is calculated once by the occurrence of the context word and it will not be counted if appeared again as described in the previous example. Therefore, the differences in context words frequency mean the frequency of new words added.

As a result, the words that are similar to"brexit" and "traffic" as in figure 3 will be considered anchor words when appearing in each tweet. This will make the relations between tweets that have similar words easier to classify. Examples on anchor words and removed words can be seen in Table II. The anchor words are the ones that will be considered when calculating the features vector for each single tweet while the rest of words will be removed.

Anchor words and the ones that will be removed will depend on how they appear in the tweets. Thus, they may not fall under any English language grammar rules. Accordingly, there might be some changes on the words polarity (anchor or not) over the time depending on how people are using these terms. However, our proposed approach produces the facility to track these changes. As a result, differences between words and context will provide a clear view about the selected words from each tweet.

B. Proposed Model

The various stages in our approach are illustrated in Figure 4. These stages will process twitter text to be classified into several classes based on words' co-occurrence patterns. The main concept of the model built on a probabilistic relation between words and context frequencies. For this reason we named the approach Probabilistic Relational Supervised Topic Modeling (PRSTM). It will be described in several stages starting from cleaning the tweets until having several classes as an output. Moreover, other steps are defining which words will contribute in the process and the rest will be removed according to what had been explained previously in the Words and Context frequency analysis section. This process is included within the Word Embeddings stage that is shown after the tweets cleaning and it will be explained in detail in the experiments section.

Words matching is the second stage before words embeddings and is after cleaning tweets. It aims to reduce the informality by finding similar words. The state-of-art Approximate String Matching Algorithm [17] used to find the similarity percentage between words. This will reduce the syntax errors leading to noise reduction in general as it will produce less ambiguous text. The next stage is to convert words to vector space and it will be described in the following subsection.

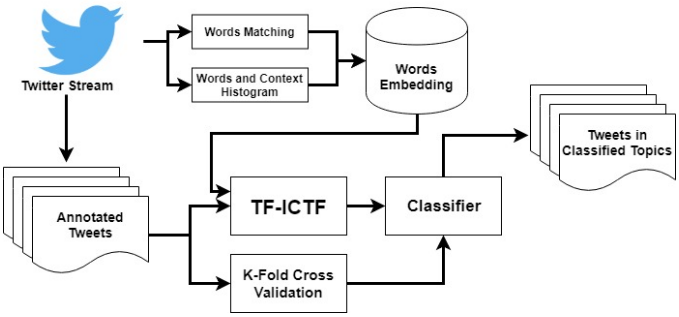

Figure 4: Probabilistic Relational Supervised Topic Modeling approach outline.

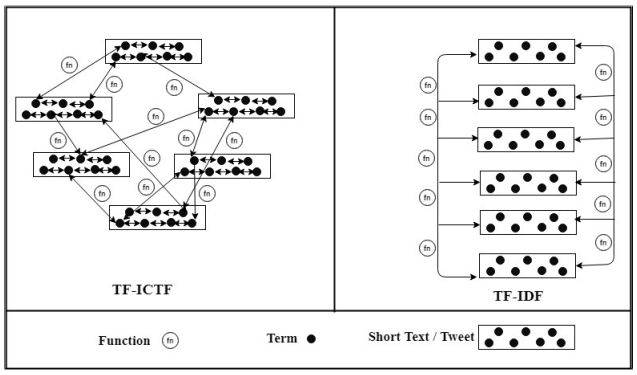

Figure 5: Proposed model (TF-ICTF) vs State-of-art (TF-IDF)

C. Term Frequency - Inverse Context Term Frequency

The TF-IDF algorithm stands for Term Frequency-Inverse Document Frequency. This algorithm performs well on long formal text to show the topic depending on words frequency. The equation below shows the calculation of *tf*:

$$tf(t,d) = 0.5 + 0.5 \cdot f_{t,d}/max\{f_{t',d} : t' \in d\} \quad (1)$$

where $tf(t,d)$ is the term frequency weight which means how many times the term appears in the document. t is the number of terms and d is the number of documents. Furthermore, 0.5 is a double normalization that had been used to reduce the impact of common and rare words. $f_{t,d}$ is the term frequency derived by $|maxf(t,d) : t \in d|$ which is the maximum number of this term appearing in all the documents. The *IDF* equation is:

$$idf(t,D) = log(N/|\{d \in D : t \in T\}|) \quad (2)$$

Where $idf(t,D)$ is the result of the inverse document log. N is the total number of documents and the base $|d \in D : t \in T|$ is the frequency of term in all of the documents.

We developed our new algorithm Term Frequency-Inverse Context Term Frequency (TF-ICTF) from the baseline well known algorithm TF-IDF. Many factors were included to make it applicable on short text. Our equations are as follow:

$$tf(t,d) = 0.5 + 0.5 \cdot f_{t,d}/C \quad (3)$$

Where C is the number of words surrounding the keyword depending on the embedding calculations as it is an ongoing process. It will vary within some range according to the pattern of this word within the tweet.

The ICTF equation is:

$$ictf(t,D) = log(N/C) + RF \quad (4)$$

Where C as above and RF is the relation factor. This RF is calculated as in the following equation:

$$RF(t) = \sum_{i=1}^{n} f(t_i) \quad (5)$$

RF will be an additional value that represents how much the anchor word relates to the other word within the same tweet.

$$f(t_i) = \textit{Term in tweet frequency } / \textit{keyword frequency} \quad (6)$$

The Logarithmic Scale that is nonlinear has been chosen in TF-ICTF equations because of the large range of quantities. These quantities are enormous number of tweets and the proposed model classifies these tweets under certain topics.

Accordingly, the number of documents of the TF-ICTF algorithm is the average number of context words' differences as described previously in the section of context and word relation. If we consider each single tweet as a document like in TF-IDF, then we will not get an accurate weight for each word. Hence, the growing tweets frequency will affect the value of each word's feature value within the tweet. The alternative is to calculate the term weight within its topic (the context prospective as suggested in TF-ICTF).

The difference between the two algorithms is shown in figure 5. Thus, the TF-IDF considers the single tweet as document. Additionally, the TF-IDF value for each term will be very small because of the number of documents which could reach millions. On the other hand, TF-ICTF will localize the problem to the term with respect to the adjacent ones within the tweet as will be described in the Embeddings section. Accordingly, the value of each term is unique and calculated with respect to other terms within the same tweet and co-occurrence pattern. Nevertheless, there is a possibility of the related words to occur in other tweets. Though, the TF-ICTF value for each tweet will be biased by the main term value.

In figure 5, the small arrows between the terms in the TF-ICTF part represent the RF function which finds the related factor between the words with respect to context. Furthermore, the"fn" sign which refers to function is defined as the TF-ICTF value. Likewise, the same symbol is a function as well but it is TF-IDF value on its part of the figure.

Moreover, the TF-ICTF values for each entity are more coherent than the TF-IDF which is affected by the frequency of the documents (Tweets). On the other hand, the TF-ICTF value will keep it much coherent because it depends on the term context rather than the term frequency as shown in Figure

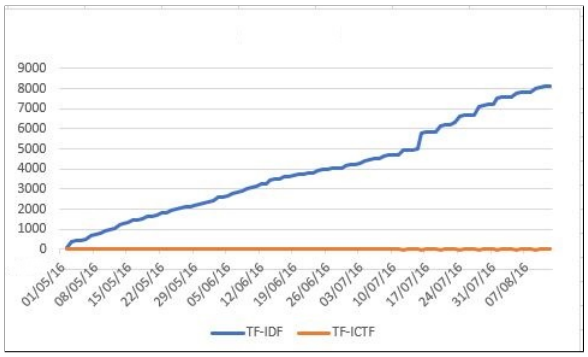

Figure 6: Comparison between the values of TF-IDF and TF-ICTF for the word"brexit"

6. The following figures show the huge differences between the two algorithms for different terms on randomly selected days where the term frequency might change.

IV. EXPERIMENTS

In this section, we describe our datasets, embeddings, and results.

A. Datasets

The proposed model designed mainly for short noisy and informal text making Twitter the most convenient domain to target. Thus, the Twitter dataset has been collected using Twitter API"Listener" which obtains an authorization from Twitter for streaming through a double key authentication. This streaming process allows 1 % of free streaming of the actual twitter data. Morstatter et al [18] discuss the issue of sufficiency of the free streamed tweets against Firehose streamed tweets for research reasons and they surmise that it is reasonable to work on the first type of streamed tweets.

The first twitter dataset was collected within the geographic location of the city of London (51.263117 Longitude/-0.659189 Latitude) from the West South, to (51.700991 Longitude/ 0.302114 Latitude) in the North East. These data have been collected from the 20th of April 2016 and continue to be collected with a total up until now of 5 GB and around 3.7 Million tweets all posted in English language. These data set will be used to build the words embedding as it will be shown in details in the words embedding section IV-B.

The second dataset comprises some randomly selected tweets from the first dataset. Thirty thousand randomly selected tweets have been annotated using Amazon Mechanical Turk [19] into several topics as shown in figure 7. The annotation was made by dozens of annotators with a condition of a linguistic background and specific steps that they should follow for annotation. This dataset was then used on the supervised Topic Modeling that will be described in details in the results section.

B. Embeddings

Several models have been produced for words embeddings like Word2vec [20] and GloVe [21]. Both embeddings convert

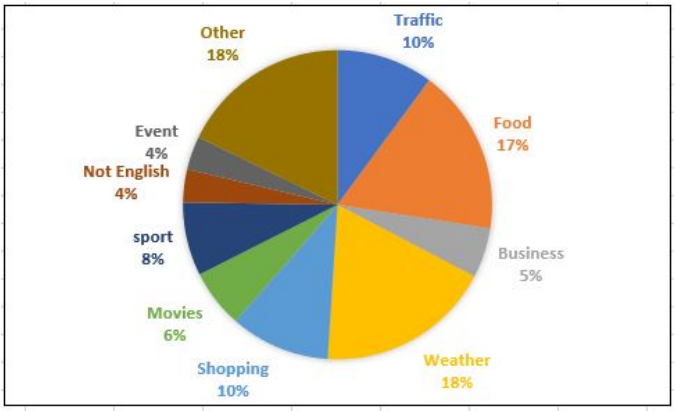

Figure 7: Annotated tweets classes.

Figure 8: How a tweet look like after cleaning process.

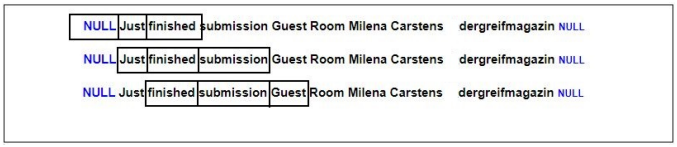

Figure 9: Sliding window along the tweet.

words to vector space with a slight difference of prediction that word2vec provides. Unlike the two examples, we produce our word embedding for this model that take into account the time as a factor. Word matching is also a second layer to aggregate similar words in one bag as it will be explained in details in this section.

Firstly, all of the streamed tweets described previously in the dataset section were collected with respect to frequency and context in one big corpus. Accordingly, a window of words has been swiped over each single tweet after cleaning. The cleaning process ensures removing the following:

1) The re-tweeted tweets which contain "rt".
2) Any websites and links.
3) English language stop words which can be easily detected and removed using Natural Language Tool Kit (NLTK) in Python language.
4) Special characters that are not part of any word and mostly mentioned in a form of emojis.

After cleaning tweets, "NULL" was added to both ends of each tweet as shown in figure 8. This is due to applying a window of size three and sliding it along the whole tweet with a calculation of each occurrence of the word. The co-occurrence of each word with the words on both sides is calculated as well as shown in Figure 9. This process was

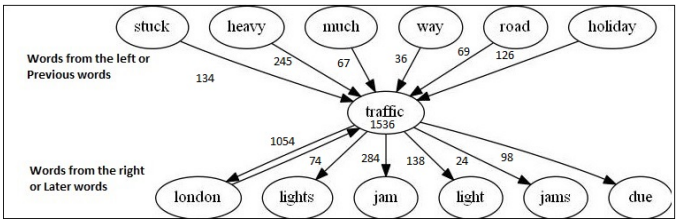

Figure 10: The output of embedding for the word "traffic" for one month.

Word	Word Length >	Word Length <	Word Length =
traffic	@london_traffic #slowtraffic #traffic #atxtraffic	trafic trafik	trafficccc
weather	#weirdweather #amazingsunnyweathertoday @skynewsweather	wether	weatheeeeer
brexit	#brexiters #thebrexitbill #thisisbrexit #brexitdefeat	-----	brexittt
m25	#m25 @m25news #worsethanm25 #m25traffic	------	------
congestion	congestione #congestion #beatcongestion #trafficcongestion	congesti	-------
goal	goals #goals2017 goalkeepers @101greatgoals	--------	goooooooooaaal

TABLE III: Finding matched words according to length and matching criteria.

applied on all tweets of the dataset to generate one big document that holds all of the words and their frequencies alongside with the co-occurrence frequencies of the words related as shown in Figure 10. For example, "traffic" word saved as a record of this words' embeddings big document. The frequency of "traffic" is 1536 and the co-occurrence frequency with the word "stuck" is 134. Similarly "london" appeared on both sides of the "traffic" with a co-occurrence frequency of 1054.

After creating embeddings, we found that there was more than 1.3 million words. When compared to the total number of words in OED [1] that does not exceed 273 thousand words, we found that most of these words were not standard. For this reason, a bag-of-words technique was used but in a form of informal related words to an anchor standard word. Each bag is tagged here by an anchor word and all of these words in the bag will be treated evenly with respect to the anchor word. The matching criteria will decide which words could fall under the related formal word bag.

Three different matching criteria were taken into consideration to create the informal bag-of-words:

1) Syntax matching to find the words similar to each anchor word as shown in Table III. For example if we take the word "weather" we will find that similar words within the corpus are either longer or shorter. Longer words could be similar to the words #whataniceweather or "Londonweather" which gave the same indication for the anchor word. These words were considered in the same bag after being accepted on the second criterion.
2) The context words (as the example shown on both sides of the word "traffic" in Figure 10) that can be taken from the embeddings created should match the anchor word.
3) An approximate string matching algorithm [17](Fuzzy String Searching) was used as another supportive tool to find anchor words with syntactical error as we see the word "wether" for example. This will also be linked to the second criterion of having the same context.

C. Results

In this section, we will show the results of the classified tweets under several topics. Table IV shows three examples of tweets for each topic selected randomly from the annotated dataset that has 30,000 observations (tweets). The chosen anchor words can be shown against each tweet on the same row. These words were detected according to words' co-occurrence patterns as described previously. The Supervised Machine Learning Topic Modeling will be trained according to TF-ICTF features vector of these anchor words.

Furthermore, several words were removed from the process because no co-occurrence patterns can be detected. On the other hand, anchor words in Table IV show how the selection of these words could represent the whole tweet. The selected anchor words are not exactly as mentioned in the tweets, for example, "drink" which is mentioned as "drinks" in the tweet and "market" that appears as "Marketing". This is because of the word matching that links any chosen word to the anchor word as previously described in the embeddings section.

As a result, the several stages of the PRSTM will classify tweets as per the related topics using the detected patterns from words embeddings built from 3.7 million tweets with respect to time and words' frequency changes. Thus, we have to evaluate our approach. The next sections will show how much the PRSTM is significant and what are the limitations that could be addressed in the future. k-fold cross validation was used with k =10 to produce better evaluation of our model performance. Furthermore, a comparison to other models will be produced with the differences against our approach.

1) k-fold Cross Validation: k-fold cross validation is classified into two types: Exhaustive and Non-Exhaustive cross validation. The first is dedicated to dividing the observations (annotated tweets) into even folds (sets) of training and validation, the observations in each validation set should appear only once during the whole process. Thus, the number of these sets will depend on the k number that defines the number of folds.

The challenge that was considered in this process is how to make it even if the number of observations in each class is different. Accordingly, we will have several unbalanced datasets. The proposed solution in our case will leave us with two choices: The first is to divide each class observation on

Topics	Tweets	Co-occurrence patterns
Traffic	poferriesfr great news makes worse im stuck middle lane surrounded trucks cant pull	stuck worse lane truck
	southernrailuk daily dose stress due train delays cancellations referendum anxiety enough	rail daily stress train delays
	Another weekend slow traffic leisure centre paid garden recycling harrow_council	weekend slow traffic
Food	Mmmmm dinner time Our new Chicken Livers crispy pancetta almonds toasted brioche	dinner chicken crispy toast
	nice lunch friends yesterday corona lime lunch pub summer sun red lion	nice lunch lime pub
	oyster shucking masterclass amp course meal w paired drinks yes pls clubdandd	course meal drink
Bussiness	Stock Market Investment Like game Chess Focus present planning next strategy gt gt NTSLequity	stock market investment planning
	Euro depression deliberate EU choice says former Bank England chief via telebusiness Should know	bank bussiness
	Remarkable Email Marketing Tips You Need Implement Right Away via HuffPostBiz	market implement
	Wind mph NE Barometer mb Rising Temperature Rain today mm Humidity	wind ne rising temprature rain today humidity
Weather	creativemadhaus Karenanne Good morning ladies I hope good weekend weather horrible cold grey	morning weekend weather horrible
	cositohoracio really good thanks beautiful blue skies suffolk today amp getting warmer	blue sky today warm
	HeathrowExpress would amazing its light brown cloth hanging suit bag Inside suit jacket pair jeans	cloth hanging suit bag jacket jeans
Shopping	adidas techfall available online instore ever seen boxing wrestling boots sugarrays	online store boot
	check uk size womens azzurra steampunk brown suede ankle boots gold detail ebay	size brown seude boot ebay
	KerryInTheCity Hiya lovely luck films LMK cos get out X	lovely film
Movies	StarWars Rogue One A Star Wars Story Underground armiesamp super troopers Released later year NewMovies	story release movie
	Help shape future Cinema Guildford Vote films youd like see CGi Autumn here nhttps	cinema film autumn
	Why shouldnt Gino amp Scott want aim more turgid football pts last games Thank Quique I back Gino watfordfc	football game
Sport	bmsleight We big game tomorrow prot Ranieri Huth Youll get one day	big game
	Our grass looking fantastic ready junior football tomorrow se selkent football	grass looking fantastic
	drscratch donc les pi sont aux normes cest juste la paperasse qui mal faite je r raison de plus pour regarder de pr	donc les pi sont cest mal qui faite je mal
Not English	La situa oggi non sono andato palestra sono grasso e povero con voglia di pizza la mangio	oggi sono andato palestra grasso
	Vampida Jajaja te entiendo Pero ir todo bien ya ver	vampida te entiendo todo bien
	My memorable Eurovision yrs back mates Bottle vodka amp trying But English subs	memorable Eurovision bottle vodka
Event	Great night Yellow_Comedy Thanks chrisogle And huge thanks And huge thanks AndrewCarberry croxley	great night comedy
	Last night I went see Twelth Night grassrootsLON I thought entertaining I thought cast excellent	last night entertain cast
	So much camera zoom its like trying video call parents	camera zoom video call parent
Other	Pink white classic colour combination Spiral Design makes arrangement contemporary stylish	pink white classic design stylish
	One revealing days life travel growth parkour journey pilgrimage Big	revealing travel growth image

TABLE IV: Chosen anchor words from tweets according to TF-ICTF and co-occurrence patterns

the number of folds leaving a remainder of a maximum *k-1* observations for each class. As a result, the total number of uncounted observation within the whole process will be *k*(k-1)*.

The second proposed solution is a repetition of the remainder samples for a padding (i.e. complement) purpose to complete *k* number of observations either by choosing random samples from the remainder or depending on the observation index. Both of the suggestions will produce a number of repetition between 1 to *k-1* with a maximum total number of observation reach to *k*(k-1)*.

Accordingly, the number of any added or left observations will be in a maximum of *k*(k-1)*. If we proposed *k* to be 10 for example, this will leave us with a maximum 90 observations. As we have 30,000 observations (as mentioned previously in the dataset section), the remainder percentage will be 0.003 which is a percentage of the error in case these observations do not work on training as it is supposed to do. As a result, it is a very small error ratio that could be accepted with a fact that it is proportionally divergent with the number of observations. The following figure 11 shows how the *k*-fold works with the above described hypotheses.

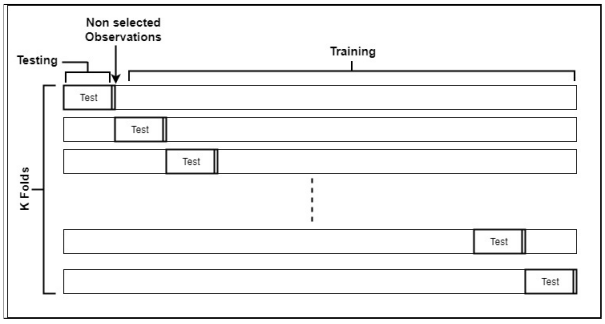

Figure 11: K-folds Cross Validation with the error

In the stage of classification, 11 classification kernels were applied on the 10 folds. The Kernels are as follow: Radial Basis Function (RBF), three Support Vector Machine (SVM) kernels (Polynomial, Sigmoid, and Linear), K-Nearest Neighbor(KNN), Multi-Layer Perception (MLP), Logestic Regression, Random Forest, Gaussian Naive Bayes (GNB), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). The results of the classifiers over the 10 folds are shown in Table V.

Each fold was divided into 30% as tested data and 70% as training. This percentage was also applied evenly on all of the topics. For example, the Weather topic is 18% from the second data set which is 5,400 tweets in total. Testing is 1,620 tweets and the rest is for training. Any single fold from the 10 folds will have 162 tweets for testing just from the Weather topic. The same calculation was applied on the training. This will produce an even evaluation processes when classifying the tweets.

The F1 Score [22] was chosen as the score measure for the classification accuracy prediction because it represents the balance between the two popular measures Precision [23] and

Classifiers	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10
GaussianNB	0.784501	0.794933	0.796423	0.740238	0.78152	0.796423	0.78301	0.799404	0.796423	0.659314
LDA	0.82772	0.821759	0.796423	0.844769	0.830462	0.818778	0.837705	0.796423	0.818778	0.829553
RBF	0.796423	0.757914	0.734423	0.746557	0.787481	0.766423	0.78152	0.799404	0.724123	0.702534
SVM Polynomial	0.830462	0.800894	0.796423	0.822504	0.794501	0.840894	0.790656	0.797914	0.803875	0.751669
SVM Sigmoid	0.705365	0.730894	0.696423	0.672355	0.691952	0.629404	0.666617	0.703875	0.650894	0.669747
SVM Linear	0.731133	0.726662	0.796423	0.784918	0.790462	0.816662	0.796423	0.743875	0.706662	0.689121
KNN	0.812817	0.821759	0.856423	0.842832	0.850462	0.817288	0.835991	0.797914	0.817288	0.869747
MLP	0.833681	0.8307	0.796423	0.843428	0.790462	0.8307	0.834501	0.78152	0.8307	0.781043
Logistic Regression	0.700894	0.760656	0.697914	0.700894	0.721759	0.796423	0.644769	0.721759	0.700894	0.696423
Random Forest	0.699404	0.666617	0.703875	0.699404	0.657914	0.636423	0.746557	0.797914	0.676662	0.696423
QDA	0.850894	0.860656	0.837914	0.820894	0.840894	0.796423	0.862504	0.830894	0.809642	0.856423

TABLE V: 10 folds Cross Validation for 11 kernels

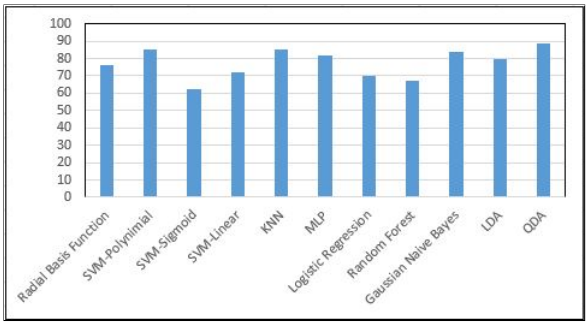

Figure 12: F1 Score Classification Accuracy Measure

Topics	PRSTM	Word2vec	Glove	TF-IDF
Traffic	0.9564	0.8097	0.7953	0.4567
Food	0.9476	0.7034	0.7812	0.3487
Business	0.7245	0.8078	0.6	0.5489
Weather	0.9389	0.9194	0.9426	0.6578
Shopping	0.8623	0.8657	0.5792	0.4378
Movies	0.9478	0.9536	0.8945	0.3576
Sport	0.9324	0.8543	0.8734	0.5634
Not English	0.6215	0.7813	0.7023	0.5367
Event	0.7567	0.8436	0.7378	0.2657
Other	0.8653	0.7392	0.8189	0.4367
Accuracy	0.8905	0.8166	0.7985	0.4172

TABLE VI: Accuracy comparison between word2vec, Glove, TF-IDF, and PRSTM

Recall [24]. The average of the F1 accuracy for the classifiers is shown in Figure 12

The results in figure 12 show that QDA was the most successful kernel with 89% accuracy, followed by SVM-Polynomial and KNN with 85% as well as GNB kernel. The accuracy results are reasonable for a noisy text and without any training on external database. However, the results will be validated with other models to show the differences.

2) Evaluation: Compared to Glove and word2vec after using the same annotated dataset in the dataset section, TF-ICTF shows better performance in total. On the other hand, Glove and word2vec took the lead in some topics as shown in Table VI.

Compared to TF-IDF, the TF-ICTF shows better performance in all classes. The representation of the words using TF-IDF did not take into account the evaluation of words with respect to context. Additionally, TF-IDF is affected by the frequency unlike TF-ICTF which work on more coherent relation between context and words frequency when calculated over time. In other words, the TF-ICTF value for each word less affected by the changed of frequency for both frequencies of the word and it context words. Working within the PRSTM also provides less noise by using the proposed word matching stage with respect to context.

On the other hand, word2vec shows better performance on topics with lower representation in the dataset. For example, Business, Not English, and Event are 5% , 4%, and 4% respectively which is reasonable as we rely on words co-occurrence patterns. Thus, more patterns will provide better performance. Despite that, the average accuracy of our proposed model shows better performance. Likewise, compared to Glove approach, PRSTM shows better performance in all of the topics.

Finally, the conclusion and future work will show the limitations of our proposed model alongside with the suggestions to develop it.

V. Conclusion and Future Work

Tweets are typically informal text containing irregular terms and therefore they pose additional challenges when attempting to extract useful information from them. For this reason, it was a challenge to use traditional algorithms or state of the art algorithms to analyze twitter text. Our proposed approach worked on extracting a relation that could be translated to features using mathematical probabilistic approach in a form of words' co-occurrence patterns. Thus, TF-ICTF was presented to translate the relation between each word and its context to numerical representation that could be more coherent through time and word frequency changes. TF-ICTF is part of the PRSTM which classifies tweets to topics like

Traffic, Weather,...,etc that is considered our main contribution. The results show better performance with 89 % accuracy using our version of words' embeddings when to some proposed techniques like word2vec, Glove, and TF-IDF with accuracy of 81 %, 79 %, and 42 % consequently. Additionally, our approach had been challenged using k-fold cross validation to measure its prediction capability. Furthermore, our version of words embeddings builds more realistic words co-occurrence relations considering the words-context frequency changes over time. This study is realistic as it worked on pure data from real life social media stream. One of the things that we noticed in our approach is the good performance in big datasets which is reasonable due to extraction of words' co-occurrence patterns. Thus, we will address the issue of finding relations between words for any possible topic to overcome the small dataset co-occurrence patterns extraction. Moreover, we will work on finding more relations between the extracted patterns to produce new ones which can be considered as a second layer of patterns extraction. This will produce more prediction power to the proposed model.

REFERENCES

[1] OED, "New words list September 2017 — Oxford English Dictionary," 2017. [Online]. Available: http://public.oed.com/the-oed-today/recent-updates-to-the-oed/september-2017-update/new-words-list-september-2017/

[2] D. M. Blei, B. B. Edu, A. Y. Ng, A. S. Edu, M. I. Jordan, and J. B. Edu, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] D. M. Blei and J. D. Lafferty, "Topic Models," *Text Mining: Classification, Clustering, and Applications*, pp. 71–89, 2009.

[4] G. B. Chen and H. Y. Kao, "Word co-occurrence augmented topic model in short text," *Intelligent Data Analysis*, vol. 21, no. S1, pp. S55–S70, 2017.

[5] H.-y. Lu, L.-y. Xie, N. Kang, C.-J. Wang, and J.-Y. Xie, "Don't Forget the Quantifiable Relationship between Words : Using Recurrent Neural Network for Short Text Topic Discovery," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 1192–1198, 2017.

[6] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," *WWW '13 Proceedings of the 22nd international conference on World Wide Web*, pp. 1445–1456, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488388.2488514

[7] K. Spärck Jones, "A Statistical Interpretation of Term Specificity and its Retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972. [Online]. Available: http://www.emeraldinsight.com/doi/abs/10.1108/eb026526

[8] K. W. Lim, C. Chen, and W. Buntine, "Twitter-Network Topic Model: A Full Bayesian Treatment for Social Network and Text Modeling," pp. 1–6, 2016. [Online]. Available: http://arxiv.org/abs/1609.06791

[9] I. Yildirim, "Bayesian Inference : Gibbs Sampling," vol. 14627, pp. 1–6, 2012.

[10] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2015-Janua, no. Ijcai, pp. 2270–2276, 2015.

[11] L. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," *Proceedings of the First Workshop on Social Media Analytics - SOMA '10*, pp. 80–88, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1964858.1964870

[12] V. K. Rangarajan Sridhar, "Unsupervised Topic Modeling for Short Texts Using Distributed Representations of Words," *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 192–200, 2015. [Online]. Available: http://aclweb.org/anthology/W15-1526

[13] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Enhancing Topic Modeling for Short Texts with Auxiliary Word Embeddings," *ACM Transactions on Information Systems*, vol. 36, no. 2, pp. 1–30, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3133943.3091108

[14] D. M. Blei and J. D. McAuliffe, "Supervised Topic Models," pp. 1–22, 2010. [Online]. Available: http://arxiv.org/abs/1003.0783

[15] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, "Topic Modeling of Short Texts: A Pseudo-Document View Yuan," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp. 2105–2114, 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939880{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2939672.2939880

[16] Y. Zuo, J. Zhao, and K. Xu, "Word network topic model: a simple but general solution for short and imbalanced texts," *Knowledge and Information Systems*, vol. 48, no. 2, pp. 379–398, 2016.

[17] E. Ukkonen, "Algorithms for approximate string matching," *Information and Control*, vol. 64, no. 1-3, pp. 100–118, 1985.

[18] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley, "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose," pp. 400–408, 2013. [Online]. Available: http://arxiv.org/abs/1306.5204

[19] Amazon Mechanical Turk, "Human Intelligence through an APIAccess a Global," 2005. [Online]. Available: www.mturk.com/

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," pp. 1–12, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[21] J. Pennington, R. Socher, and C. D. Manning, "GloVe : Global Vectors for Word Representation." [Online]. Available: https://nlp.stanford.edu/pubs/glove.pdf

[22] Y. Sasaki, "The truth of the F-measure," *Teach Tutor mater*, pp. 1–5, 2007. [Online]. Available: http://www.cs.odu.edu/{∼}mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf

[23] J. W. Perry, A. Kent, and M. M. Berry, "Machine literature searching x. machine language; factors underlying its design and development," *American Documentation*, vol. 6, no. 4, pp. 242–254. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090060411

[24] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the Ieee Computer Society Technical Committee on Data Engineering*, vol. 24, no. 4, pp. 1–9, 2001. [Online]. Available: http://160592857366.free.fr/joe/ebooks/ShareData/ModernInformationRetrieval-ABriefOverview.pdf