Conventional clustering-based method for event detection on social networks

Anonymous ACL submission

Abstract

Social networks are becoming the preferred channel to report and discuss events happening around the world. The information stream such channels contain can be used to detect and describe the ongoing events to take informed decisions in numerous domains. A typical framework for event detection is to first cluster the stream of tweets, and then analyze the clusters to decide which deal with real-world events. In this context, content representation models and clustering approaches are critical. Classical approaches are usually based on TF-IDF for the representation of the text content and on dynamic clustering for the clustering part. In this paper, we propose to compare TF-IDF with recent text representation models and we propose an event detection method based on conventional clustering. We show that, contrary to previous results, language models based on Transformer architectures are competitive with TF-IDF. We also show that our approach outperforms the most used approach of the literature.

1 Introduction

Social networks are some of the main contemporary information sources, used by people but also by professionals such as the journalists, business managers, politicians and so on. They can deliver information about numerous domains and can be used to predict the stock market (Bollen et al., 2011), (Oliveira et al., 2017), (Ruiz et al., 2012), (O'Connor et al., 2010), they can help authorities to react in emergency situations (Imran et al., 2015), (Kim and Hastak, 2018), (Sakaki et al., 2010), (Basu et al., 2017) and can be used in general to detect events happening around the world (Hasan et al., 2018), (Atefeh and Khreich, 2015), (Elsafoury, 2020).

Due to the abundance of information and noise on social networks, tools are necessary to keep track of important events. A classical task of information retrieval is to detect event on social media (Allan, 2012). In previous work by McMinn (McMinn et al., 2013), an event is a "significant thing that happens at some specific time and place". They identify an event by a group of entities (e.g. people, location) that is discussed in the messages from the social network. We borrow this definition for this work and apply it to the problem of event detection on Twitter.

A major challenge of this task is to group documents dealing with the same event together. The text content of each document usually contains unstructured language, slang words or abbreviation but also limited context about the topic, making its representation difficult. The other major factor is the clustering algorithm employed. The most classical approach in the literature is to use dynamic clustering and particularly the First Story Detection algorithm (FSD).

In this paper, we propose a new event detection method based on conventional clustering, called Conventional Clustering Event Detection Method (CCEDM) and compare the performances of our method with the FSD algorithm, a method commonly used in the literature and considered as the state-of-the-art (Hasan et al., 2019), (Mazoyer et al., 2020). We also propose to use Transformer-based language model for the representation of the textual content. These models currently achieves state-of-the-art results in Natural Language Processing (NLP) (Vaswani et al., 2017). In previous work, they showed that these models are outperformed by TF-IDF, the most classical text representation in information retrieval (Baeza-Yates and Ribeiro-Neto, 1999), in the context of the FSD algorithm (Mazoyer et al., 2020). We explore whether these results are confirmed in our context. We believe that proposing an event detection method in which Transformer-based language models perform correctly is an interesting goal considering the current path followed by the research in deep learning.

The rest of this paper is organized as follows:

Section 2 presents the related work. Section 3 describes our event detection approach. Section 4 describes the experiments and the results.

2 Related work

2.1 Text representation models

Text content representation models are one of the major issues in information retrieval. The current reference model is TF-IDF (Jones, 1972) which is an improvement of the Bag Of Words (Harris, 1954). TF-IDF allows to take into account the importance of the words in the representation of the document by weighting each word in inverse proportion to the number of documents in which the words appear. Thus, a word appearing frequently in a document while it appears rarely in the corpus is considered as carrying a lot of information about this document. This word will be highly weighted in the TF-IDF representation of the document. TF-IDF vectors are sparse in the context of Twitter due to the large vocabulary and short size of the documents. This representation is widely used, even nowadays, in information retrieval and obtains very good performances, particularly on short texts extracted from social networks.

These statistical representations are currently complemented by dense vector representations, called word embeddings, based on deep learning approaches. The authors of (Mikolov et al., 2013) introduce the Word2vec model which corresponds to a neural approach allowing to associate to a word a vector, which is computed depending the context in which the word appears in the training set. Thus, the vector representing a word contains information about it. The assumption made for the constitution of these vectors is that words whose contextual use is close will carry a similar meaning and thus will be represented by a close vector. The most recent models based on neural networks are based on Transformers architectures (Vaswani et al., 2017). The most notable implementation of the Transformer architecture in NLP is BERT (Devlin et al., 2018). BERT is a language model based on the principle of Transfer Learning (Pan and Yang, 2010). The idea is that learning some general task and then apply this knowledge to a more specific task can be improve the performances on the downstream task.

Most of the presented models allow to represent words but do not necessarily allow to represent sentences, which could be interesting in the context of short text documents such as tweets. The most recent are also based on Transformers. Universal Sentence Encoder (USE) (Cer et al., 2018) is trained on two types of tasks, a supervised one, based on the SNLI dataset (Bowman et al., 2015) in the same way as Infersent (Conneau et al., 2017), and on unsupervised tasks, like Skip-Thought (Kiros et al., 2015), which notably include social network documents. Transformers architectures can also be used in the form of Siamese networks (Bromley et al., 1994) i.e. two neural networks in parallel, having the same architecture and the same weights, but which will not take the same input. The vanilla BERT architecture performs poorly on short documents of the size similar to a sentence and performs better with longer documents so another approach is needed. The authors of (Reimers and Gurevych, 2019) propose S-BERT (Sentence BERT) which consists in creating a Siamese network of two BERT models which will be trained with the objective of producing similar vectors for sentences whose meaning is close and dissimilar vectors for sentences whose meaning is distant. Then, a last layer of neurons is added, so that it can be fine-tuned on specific tasks.

2.2 Event detection methods

We focus on the task of open-domain event detection on Twitter which consists in detecting events that are not known beforehand (Atefeh and Khreich, 2015). Event detection methods usually falls between two categories : feature pivot or document pivot (Atefeh and Khreich, 2015). We choose a document pivot approach because it allows to take into account more context and metadata, and present some of these methods hereafter.

One of the most common approach for event detection is the FSD (First Story Detection) algorithm, which was first introduced by Allan et al. in (Allan et al., 2000). The principle is to find the first document discussing an event and then group together new documents discussing the same event. To do so, the task is considered as a dynamic clustering task, using nearest neighbors algorithm to group the documents. Several papers improved this algorithm to speed it up (Petrović et al., 2010; Repp and Ramampiaro, 2018; Hasan et al., 2019), improvements being mostly focused on the nearest neighbor search. In all these papers, the tweets are represented using TF-IDF. In (Mazoyer et al., 2020), the authors compare the

Figure 1: A high level representation of a typical Event Detection Framework.

performances of different text representations for the tweets in the context of FSD. They compare TF-IDF and neural-based representation models such as Word2vec (Mikolov et al., 2013), ELMO (Peters et al., 2017), BERT (Devlin et al., 2019), S-BERT (Reimers and Gurevych, 2019) and Universal Sentence encoder (Cer et al., 2018). They evaluate individual models and try to use TF-IDF weights to weight neural-based representations. They conclude that representation models based on recent architectures such as Transformers perform worse than TF-IDF in the context of FSD, which is interesting considering that Transformers architectures are achieving state-of-the-art results in most NLP tasks.

Concerning the approaches that are not based on the FSD algorithm, TF-IDF is also the most common text representation model. The authors of (Becker et al., 2011) use it as well and then cluster topically similar tweets using an online incremental clustering algorithm. In (McMinn and Jose, 2015), the authors combine TF-IDF and named entities (NE) to cluster the tweets, based on similarity criteria but also the length of the tweets. In (Boom et al., 2016), the authors propose the first method combining TF-IDF and semantic representation. They learn a representation for the words in the documents and then weight them based on their TF-IDF score, creating weighted semantic representations. They consider that two tweets are semantically related if they are generated by the same event. The authors of (Zhou et al., 2017) extract events from Twitter using non-parametric Bayesian Mixture Model with Word Embeddings. They create event clusters from tweets and the events are modeled as a 4-tuple $< y, l, k, d >$, modeling non-location NE, location NE, event keywords and date. The components of the quadruple are generated using a multinomial distribution computed with Dirichlet process. Following the same idea of representing events using structured representation, the authors of (Li et al., 2017) include semantic by splitting tweets terms reflecting one or more event aspects. The semantic classes include NE, mention, location, hashtag, verb, noun and embedded link. They group tweets into clusters using class-wise similarity.

Thus, the majority of the work relies on TF-IDF as a representation model and the FSD algorithm is one of the most represented in the literature. In the rest of this paper, we challenge the FSD with our approach CCEDM and study the performances of Transformer-based language models in the context of CCEDMk. The objective is to explore whether they perform better than classical representation models, contrary to the context of FSD.

3 Conventional Clustering Event Detection Method

We propose to treat the problem of event detection in textual data stream as a clustering task (Allan, 2012). This allows us to get out of the constraint imposed by dynamic clustering, i.e. we can consider all the documents published at the time of partitioning, and not have to work with fragmentary information over the flow of documents. We designed the method to be flexible, so any vectorial text representation model and any classical clustering algorithm can be used. This flexibility is particularly interesting because it is important to be able to modify the representation model/clustering algorithm pair, to adapt to the quickly evolving state-of-the-art of these domains. To be in a classical clustering context, we split the data stream using windows, i.e. fixed size windows (fixed number of documents). This approach ensures that the documents clustered together have a similar publication date, which improves the chances that the documents actually discuss the same event.

In this paper, we are interested in evaluating the performances of different representation models/clustering algorithms pairs. To properly do that, we focus on the beginning of the framework presented in Figure 1, which is a typical event detection framework. We stop after the "Documents clustering" step. Thus, we make the following hypothesis : (1) all the documents are event related, (2) each document is associated with exactly one event, (3), there is an unknown number of documents. Under these assumptions, we can reduce

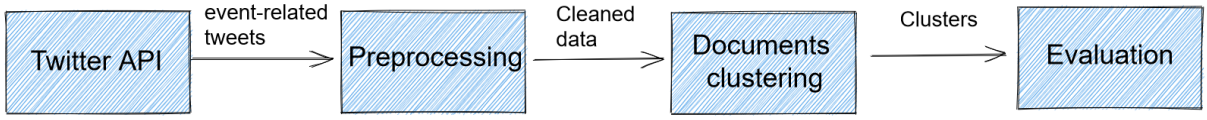

Figure 2: The framework on which CCEDM is based.

the framework and limit the steps that can affect the performances, which is commonly done in the literature (Becker et al., 2010; Boom et al., 2016; Mazoyer et al., 2020). No filtering will be performed on the documents as they are all event-related. In a more real-world setup, filtering steps are applied to filter spam and uninteresting documents. After the "Documents clustering" step, clusters are usually evaluated to determine whether they discuss an event or just a mundane conversation and then are summarized to be presented to humans. These steps are independent from the clustering phase in such framework and thus are out of the scope of this paper. Considering these modifications, we present the adapted framework in Figure 2. Both the FSD algorithm and CCEDM follow this framework. In the next section, we detail the steps of CCEDM in more formal way.

3.1 Formal description of the clustering process

First, we receive a stream of event-related input documents annotated as $D = \{d_1, ..., d_N\}$. We define a document as a $\forall i \in [1..N], d_i = (txt_i, dte_i, tag_i, url_i, src_i)$ where txt_i refers to the text content, dte_i to the publication date, tag_i refers to the tags and url_i refers to the urls shared and src_i to the source which posted the i^{th} document. We perform different cleaning steps described in Section 4.1 to obtain a set of cleaned documents. Then, we discretize the stream using windows which is classical in the literature (McMinn and Jose, 2015; Naaman et al., 2011; Guille and Favre, 2014) because it is important to ensure that documents clustered together have a similar publication date, since documents dealing with the same events are usually posted during a similar period of time. They are annotated as $W = \{W^1, ..., W^m\}$ where $\forall k \in [1..m], W^k = \{d_1^k, ..., d_\tau^k\}$, where k refers to the k^{th} window and τ to the number of documents in each window. The windows are considered as independent from each others; i.e., $\forall k \in [1..m], \forall l \in [1..m], l \neq k, W^k \cap W^l = \emptyset$. Each window is partitioned in groups of similar documents known as clusters. The documents in W^k are then clustered according to similarity metrics (e.g. text similarity) to obtain a set of clusters such as $\forall i \in [1..n], \forall j \in [1..n], i \neq j, C_i^k \cap C_j^k = \emptyset$ and $\bigcup_{j=1}^{n} C_j^k = W^k$. Thus, our event detection framework is a succession of clustering process as a result of the discritization of the stream using fixed size windows. This process is illustrated in Figure 3. This differs from the FSD algorithm which treats the problem of event detection as a dynamic clustering problem. We now present the different algorithms and models used for each step of the framework.

3.1.1 Representation models

We compare two types of text document representations : statistical approaches, also called lexical approaches and Transformer-based language models, also called semantic approaches.

Lexical approaches - We use TF-IDF, which is the most common text document representation model in information retrieval (Baeza-Yates et al., 1999). We use an IDF calculated on the whole dataset Event2012 (McMinn et al., 2013), presented in section 4.1, provided by (Mazoyer et al., 2020) and do not take into account term-frequency (TF) because most of the word appears only once in short documents.

Semantic approaches - Semantic representations of text documents are currently the state-of-the-art in NLP, particularly using Transformer-based language models (Vaswani et al., 2017). In particular, we will compare two languages models : S-BERT (Reimers and Gurevych, 2019) and Universal Sentence Encoder (USE) (Cer et al., 2018).

3.1.2 Clustering

For each pair of documents and for each document representation model, we compute its similarity to constitute a similarity matrix S_{model,W_k} used to compute the clusters. We chose Cosine Similarity as it is the most common similarity measure in NLP (Aggarwal and Zhai, 2012). It is important to note that the performances of the clustering are directly affected by the similarity measures making it a critical step of the event detection process. Using

these similarities, clusters are computed using the Louvain algorithm (Blondel et al., 2008), a well-known community detection algorithm which automatically computes the optimal number of clusters. This aspect is especially important in our context of open-domain event detection, in which the number of event is not known beforehand. The only parameter that this algorithm need is a similarity threshold, which will be different for each representation model.

Now that we have presented the different algorithms used for CCEDM, we present the different experiments we conducted and the results obtained.

4 CCEDM and FSD : experiments and results

In this section, we present two experiments, conducted to evaluate different aspects. The goal of the first experiment is to validate that CCEDM, based on classical clustering, has better performances than the FSD. The goal of second experiments is to evaluate the performances of Transformer-based language models compared to TF-IDF in the context of CCEDM.

For each of these experiments, we first present the experimental protocol and then the results. We include significance tests, using $\alpha = 0,05$. We use the "Wilcoxon signed-rank test", which is the method which fits the best our context (Yeh, 2000). Indeed, we use non parametric test methods due to the characteristics of our data.

4.1 Experimental configuration

4.1.1 Evaluation measures

We use the B-cubed measure for the evaluation of the clusters produced. B-cubed is a generalization of Precision, Recall, F1-score for clustering and is the most complete cluster evaluation measure (Amigó et al., 2009). Precision P is defined as the proportion of documents in the document's cluster that correspond to the same event. Recall R is defined as the proportion of documents that correspond to the same event, which are also in the document's cluster. To obtain the F1 Score, we use the following formula: $F1 = \frac{2*P*R}{P+R}$.

4.2 Dataset

We use Event2012 (McMinn et al., 2013), a corpus of 120 millions tweets, collected from the 10th of October to the 7th of November 2012 from the Twitter streaming API. 159,952 tweets are labeled as event-related, distributed into 506 events, which are distributed into 8 categories. We only work on the annotated part of the dataset in order to be able to evaluate properly our results. Due to the TREC policy, only tweet ids can be shared and the actual content of the tweets have to be retrieved using the Twitter API. Some tweets are not available anymore, due to deletion of the tweet, of the account which posted the tweet, or because the account is not public anymore. Thus, we collected 69,875 labeled tweets, which are distributed into 504 events. To simulate a stream of data as it would be in a real-world context, we sorted the dataset according the date of publication of each tweet. We divide the dataset into two equal sets : the train set and the test set. We use windows of $\tau = 2000$ tweets to have a representative number of documents while keeping the windows short in terms of time. We used the whole annotated dataset for the first experiment, and the test set for the second experiment.

4.2.1 Representation models

We use two variations of TF-IDF and S-BERT, and we use the model USE-LARGE[1], called USE in the rest of this paper. Concerning TF-IDF, we use the implementation proposed by (Mazoyer et al., 2020). The first one, named **TF-IDF dataset**, calculated IDF on the labeled tweets of the dataset. The second, **TF-IDF all tweets**, calculated IDF on the whole dataset. Concerning S-BERT, the first version, named **S-BERT nli** is the pretrained version on the NLI dataset, available using the implentations proposed by the authors of (Reimers and Gurevych, 2019)[2]. We chose this model because the NLI dataset is known to improve the performances of the models for clustering tasks (Bowman et al., 2015). The second version of S-BERT is **S-BERT fine-tuned**. It is a fine-tuned version of S-BERT on the training set, which is the first half of the labeled dataset. The events are used as the target labels. The particularity of this training set is it is ordered according to the publication date of the documents, thus, the major part of the event in the training set are not in the test set. The fine-tuning is done on 36 000 tweets, to fit with the size of the windows we chose. We assigned to each tweet a pair of tweets, a tweet from the same label, and a tweet from a different label, as it is usually done to train siamese neural networks.

[1] https://tfhub.dev/google/universal-sentence-encoder-large/5

[2] https://github.com/UKPLab/sentence-transformers

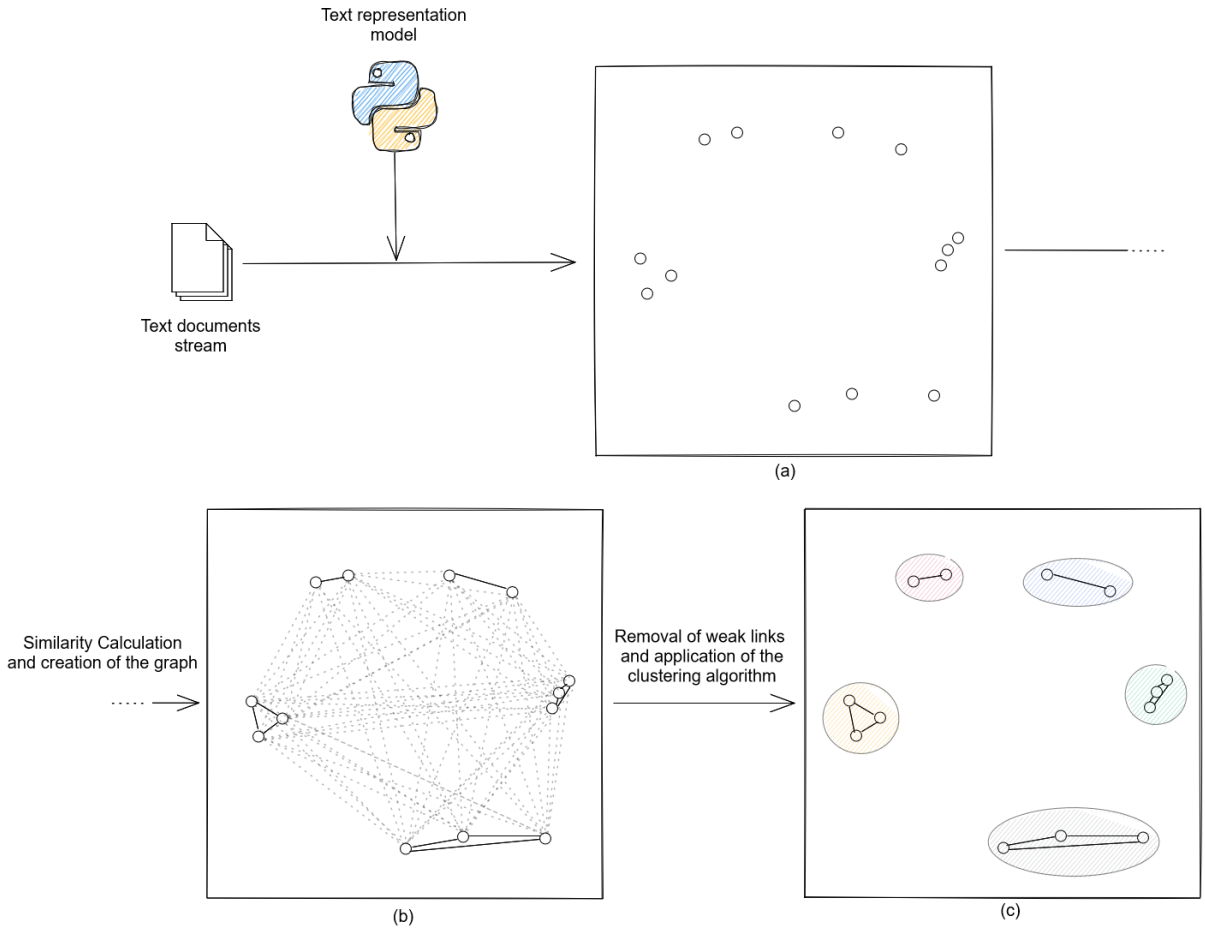

Figure 3: Data treatment process performed by CCEDM for each window. (a) Documents representations in vector space. Each document is represented by a point. (b) A graph is created using the similarity matrix. Each document is a vertex and each edge is weighted using the similarity between documents. (c) Creation of the clusters, by deleting edges with a low weight.

Table 1: Clustering quality according to the metric B-Cubed for each textual representation, according to the clustering algorithm. In nearly every case, CCEDM performs better than FSD.

Model	Approach	Precision	Recall	F1 Score
TF-IDF dataset	FSD	0.727 ± 0.128	0.523 ± 0.184	0.573 ± 0.150
	CCEDM	$\mathbf{0.930} \pm 0.048$	$\mathbf{0.702} \pm 0.276$	$\mathbf{0.756} \pm 0.240$
TF-IDF all tweets	FSD	0.781 ± 0.107	0.552 ± 0.199	0.613 ± 0.161
	CCEDM	$\mathbf{0.929} \pm 0.039$	$\mathbf{0.751} \pm 0.272$	$\mathbf{0.805} \pm 0.245$
USE	FSD	$\mathbf{0.919} \pm 0.001$	0.379 ± 0.01	0.500 ± 0.01
	CCEDM	0.918 ± 0.01	$\mathbf{0.664} \pm 0.01$	$\mathbf{0.729} \pm 0.01$
S-BERT-nli	FSD	$\mathbf{0.968} \pm 0.023$	0.323 ± 0.159	0.460 ± 0.195
	CCEDM	0.880 ± 0.075	$\mathbf{0.611} \pm 0.244$	$\mathbf{0.680} \pm 0.207$

Each of these two tweets is randomly chosen in the training set, using the rules defined about the labels. **S-BERT nli** was used during the first experiment, and **S-BERT fine-tuned** during the second.

4.2.2 Preprocessing

To clean the tweets, we remove from the text the user and retweet mentions and the URLs.

4.3 First Experiment

4.3.1 Experimental protocol

This first experiment is the comparison of four text representation models, **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT nli** and **USE**, in two different contexts, i.e. in the context of FSD or in the context of CCEDM. For the FSD implementation, we use the one proposed by (Mazoyer et al., 2020)[3] and adapt this solution. Indeed, we chose to apply this algorithm to windows of 2000 tweets and use B-Cubed as a performance measure. Thus, we formulate the following H0 hypothesis : "There is no statistically significant difference between the performance of FSD and CCEDM". To validate this hypothesis, we use the "Wilcoxon signed-rank test". Concerning the threshold values used for the FSD algorithm, we used the same as the one presented in (Mazoyer et al., 2020), i.e. t=0.65 for TF-IDF dataset, t=0.75 for TF-IDF all tweets, t=0.39 for S-BERT and t=022 for USE. The threshold values used for CCEDM are the following: t=0.39 for models based on TF-IDF, t=0.79 for S-BERT, t=0.59 for USE. As a reminder, these similarity values are computed using Cosine Similarity. These threshold values were determined empirically.

4.3.2 Results

Table 1 show the results of this experiment. The number presented are the mean of each metric for each window and the standard deviation. In most cases, CCEDM performs better than FSD. The results of the significance tests are presented in Table 2. The test is done between the values of all metrics, for each method, for each window for tweets. In every case, we can see that the p-value is always less than α.

4.4 Second experiment

4.4.1 Experimental protocol

The second experiment goal is to compare **TF-IDF dataset**, **TF-IDF all tweets**, **S-BERT fine-tuned** and **USE** in the context of CCEDM, on the test dataset. The performances are evaluated using B-cubed. We formulate the following H0 hypothesis: "None of the approach is significantly better than the others". The threshold values used for this experiment as the same as before, i.e. t=0.39 for TF-IDF based models, t=0.79 for S-BERT, and t=0.59 for USE. This experiment is useful to compare these representation methods to each other, to determine which is the most efficient method. In particular, we want to investigate the relative performances of the Transformer-based language models compared to the models based on TF-IDF. As a reminder, in (Mazoyer et al., 2020), the authors showed that the Transformer-based language models were poorly performing on this dataset in the context of the FSD algorithm and that the models based on TF-IDF performed the best. We did not fine-tune USE because it cannot be easily done, and this issue was raised multiple times on the official Github repository of USE. Anyway, BERT is currently the most standard language model, so it is logical to focus on this particular language model.

Table 2: P-value for the Wilcoxon signed-rank "FSD vs CCEDM". In every case, P-value<α.

Model	Precision	Recall	F1 Score
TF-IDF dataset	2.47 e-07	1.14 e-06	8.21e-05
TF-IDF all tweets	2.47e-07	1.31e-07	2.21e-05
S-BERT nli	3.65e-07	2.47e-07	2.47e-07

4.4.2 Results

Results are presented in Table 3 and the results of the significance tests in Table 4.

Thus, the performances are better for the approach based on TF-IDF in terms of Precision but in terms of recall and F1 score, the Transformer models perform better. The significance tests show that TF-IDF methods performs significantly better in terms of Precision, Transformers in terms of Recall. USE performs significantly better in terms of F1 score, but S-BERT not.

4.5 Discussion of the results

The first experiment showed that CCEDM performs better than the FSD algorithm in most of the presented cases. This finding is especially

[3] https://github.com/ina-foss/twembeddings

Table 3: Clustering quality according to the metric B-Cubed for each textual representation, in a supervised context, on the test dataset.

	Précision	Rappel	F1 Score
TF-IDF dataset	0.904 ± 0.044	0.769 ± 0.216	0.805 ± 0.170
TF-IDF all tweets	$\mathbf{0.929} \pm 0.035$	0.750 ± 0.215	0.805 ± 0.184
S-BERT fine tuned	0.851 ± 0.067	0.837 ± 0.170	0.828 ± 0.106
USE	0.875 ± 0.061	$\mathbf{0.855} \pm 0.211$	$\mathbf{0.839} \pm 0.158$

Table 4: P-value for the Wilcoxon signed-rank test. Not all the results are significant.

	Précision	Rappel	F1 Score
S-BERT nli fine-tuned / TF-IDF dataset	8.39e-04	6.65e-03	*0.963*
S-BERT nli fine-tuned / TF-IDF all tweets	7.62e-05	7.62e-05	*0.889*
USE / TF-IDF dataset	1.49e-02	1.34e-02	6.38e-02
USE / TF-IDF all tweets	3.81e-04	4.57e-05	2.32e-02

true for the recall measure. Concerning precision, and particularly for Transformer-based language models, the values of FSD and CCEDM are close. We believe that the FSD algorithm allow in these cases to obtain coherent clusters (high precision). However, the FSD seems to have a tendency to segment documents of a same label in different clusters, resulting in a drop in recall. This is probably due to the fact that the FSD algorithm can create a new cluster when a new document arrives, without taking into account all of the documents of the window. This segmentation is less frequent with CCEDM, explaining the better recall values.

We also showed that the Transformer-based language models, especially USE, can be competitive with classical methods (TF-IDF). We can note that in a unsupervised context (experiment 1), S-BERT performs worse than USE. We believe this is due to the dataset used for the pre-training of the different language models. Indeed, the S-BERT model that we used is based on BERT NLI, which is trained on the English Wikipedia Corpus, on BookCorpus and fine-tuned on SNLI. USE is, for its part, trained on a more diverse dataset, including data from discussion forums, and question-answer websites. These data are closer to the one we encounter in the dataset Event2012, which are extracted from Twitter. Thus, data extracted from social network, for which the syntax is very specific because of the destructuration of the language, are a problem for the vanilla S-BERT because it is trained on data written in a more conventional English. Once S-BERT is fine-tuned on social network data, the performances rise and they become similar to the performances of other models. Thus, the fine-tuning phase is particularly important and it shows that fine-tuning S-BERT on data extracted from social network allows us to obtain better results in our context. It is an interesting result considering that most of the events of the training set, the targets, are not present in the test set. Thus, the training is useful, even in a context where some concept drift happens.

4.6 Conclusion

In this article, we showed that considering the problem of event detection as a clustering problem (CCEDM) rather than a dynamic clustering problem (FSD) allows to achieve better performances. We also showed that in certain context, Transformer-based language models can have performances similar to classical models (TF-IDF). Finally, we showed that the fine-tuning of these language models is particularly interesting to adapt to the specific data extracted from the social networks. In future work, we plan to apply our method to a more realistic context by including non-event related documents. A major issue in this context is to be able to evaluate the methods while most of the documents are not annotated. We plan to propose new evaluation metrics in order to facilitate the evaluation of the models and the reproducibility of the experiments. We also plan to investigate the other building blocks of the classical event detection framework, namely the event detection phase, exploiting graph neural networks.

References

Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer.

James Allan. 2012. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. 2000. Detections, bounds, and timelines: Umass and tdt-3. *Proceedings of Topic Detection and Tracking Workshop*.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486.

Farzindar Atefeh and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Moumita Basu, Anurag Roy, Kripabandhu Ghosh, Somprakash Bandyopadhyay, and Saptarshi Ghosh. 2017. Microblog retrieval in a disaster situation: A new test collection for evaluation. In *SMERP@ ECIR*, pages 22–31.

Hila Becker, Mor Naaman, and Luis Gravano. 2010. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300.

Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. volume 11.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, P10008:1–12.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Cedric De Boom, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. 2016. Representation learning for very short texts using weighted word embedding aggregation. *CoRR*, abs/1607.00570.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, pages 737–737.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fatma Elsafoury. 2020. Teargas, water cannons and twitter: A case study on detecting protest repression events in turkey 2013. In *Text2Story@ ECIR*, pages 5–13.

Adrien Guille and Cécile Favre. 2014. Mention-anomaly-based event detection and tracking in twitter. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 375–382. IEEE.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. 2018. A survey on real-time event detection from the twitter data stream. *Journal of Information Science*, 44(4):443–463.

Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. 2019. Real-time event detection from the twitter data stream using the twitternews+ framework. *Information Processing & Management*, 56(3):1146–1165.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):1–38.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Jooho Kim and Makarand Hastak. 2018. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, 38(1):86–96.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.

Quanzhi Li, Armineh Nourbakhsh, Sameena Shah, and Xiaomo Liu. 2017. Real-time novel event detection from social media. In *2017 IEEE 33Rd international conference on data engineering (ICDE)*, pages 1129–1139. IEEE.

Béatrice Mazoyer, Julia Cagé, Nicolas Hervé, and Céline Hudelot. 2020. A french corpus for event detection on twitter. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6220–6227.

Andrew J McMinn and Joemon M Jose. 2015. Real-time entity-based event detection for twitter. In *International conference of the cross-language evaluation forum for european languages*, pages 65–77. Springer.

Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 409–418.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mor Naaman, Hila Becker, and Luis Gravano. 2011. Hip and trendy: Characterizing emerging trends on twitter. *JASIST*, 62:902–918.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan Routledge, and Noah Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4.

Nuno Oliveira, Paulo Cortez, and Nelson Areal. 2017. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 181–189, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Øystein Repp and Heri Ramampiaro. 2018. Extracting news events from microblogs. *Journal of Statistics and Management Systems*, 21(4):695–723.

Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. 2012. Correlating financial time series with micro-blogging activity. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 513–522.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. *arXiv preprint cs/0008005*.

Deyu Zhou, Xuan Zhang, and Yulan He. 2017. Event extraction from twitter using non-parametric bayesian mixture model with word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 808–817.