Tracking news stories in short messages in the era of infodemic

Guillaume Bernard, Cyrille Suire, Cyril Faucher, Antoine Doucet, Paolo Rosso

Tracking news stories in short messages in the era of infodemic

Guillaume Bernard[1][0000−0001−5945−4865], Cyrille Suire[1], Cyril Faucher[1], Antoine Doucet[1][0000−0001−6160−3356], and Paolo Rosso[2][0000−0002−8922−1242]

[1] Université de La Rochelle, Laboratoire L3i, 17000 La Rochelle, France
{guillaume.bernard,cyrille.suire,cyril.faucher,antoine.doucet}@univ-lr.fr
https://l3i.univ-larochelle.fr/
[2] Universitat Politècnica de València, València, Spain
prosso@dsic.upv.es

Abstract. Tracking news stories in documents is a way to deal with the large amount of information that surrounds us everyday, to reduce the noise and to detect emergent topics in news. Since the Covid-19 outbreak, the world has known a new problem: infodemic. News article titles are massively shared on social networks and the analysis of trends and growing topics is complex. Grouping documents in news stories lowers the number of topics to analyse and the information to ingest and/or evaluate. Our study proposes to analyse news tracking with little information provided by titles on social networks. In this paper, we take advantage of datasets of public news article titles to experiment news tracking algorithms on short messages. We evaluate the clustering performance with little amount of data per document. We deal with the document representation (sparse with TF-IDF and dense using Transformers [26]), its impact on the results and why it is key to this type of work. We used a supervised algorithm proposed by Miranda et al. [22] and K-Means to provide evaluations for different use cases. We found that TF-IDF vectors are not always the best ones to group documents, and that algorithms are sensitive to the type of representation. Knowing this, we recommend taking both aspects into account while tracking news stories in short messages. With this paper, we share all the source code and resources we handled.

Keywords: Text Classification and Clustering · News · Social data

1 Introduction

Tracking emergent topics from news is a long-standing task in natural language processing (NLP), investigated since the last century [2]. In many fields, from politics [25], IT services [28,22] to banking [20], the purpose of discovering and tracking related news stories is an important application. It helps taking better, faster decisions than one's competitors.

Furthermore, the emergence of news sources, from official agencies to institutional blog posts, including social networks, restructured the information sector.

Since the 2010's, social networks have become a main source of information for a huge part of the population [30]. Consequently, official media relay their articles on Facebook or Twitter to draw audience to their websites. As a consequence, more and more people read social networks to inform themselves. On another hand, on-line social networks allowed non-institutional parties to publish and promote information and create communities.

Tracking news stories has been attempted with some success in recent years [20,22,32]. However, for most of experiments with short messages neither datasets nor implementations are shared [13,27,33,23,4]. With Covid-19, the world health organisation (WHO) introduced the concept of *'infodemic'* as *'too much information including false or misleading information in digital and physical environments during a disease outbreak'* [34]. We take benefit of this situation and of Covid-19 Twitter News datasets to conduct comparative experiments with different corpora.

In this paper, we address a few research issues and we propose a framework to track news stories in short messages. We focus on news article titles as they are shared on social networks, and aim to discover coherent clusters of events. We first address the problem of document representation and look into how algorithms interact with document features, formulating the hypothesis that they impact the results. Next, we experiment with news story tracking with article titles using two algorithms, one of them supervised and the other unsupervised. The latter is made relevant by the lack of annotated datasets in this research field. We also release the implementation of our tracking algorithms in Python Packages, as well as all the datasets and resources we used [3].

2 Related Work

The task of tracking news stories generally consists in ordering and clustering together documents reporting the same news story, written in identical or different languages [2]. A news story is an ordered collection of documents that relate a specific topic and all its subsequent developments [2]. The final football match of 2018 FIFA World cup is, for instance, the seminal event that is the root of a story. All the articles related to the preparation of the match, betting and editorials about the results are all related to the same event. It is part of a wider topic: 2018 FIFA World Cup, or more generally sports.

The Topic Detection and Tracking (TDT) project in 2002 [2] addressed the question of tracking news stories. Documents were grouped on macro topics: finance, sports, health, etc. In 2005, the Europe Media Monitor project [25] enriched the field of study with new results and strategies to identify emergent topics from press articles. They proposed an approach to track real-world events, not only macro-topics. The newsBrief system is still running [4] and gives a view of trending topics mentioned in news articles.

[3] Links to be added if the paper is accepted.

[4] `https://emm.newsbrief.eu/`

Later on, after 2010, the Event Registry project [19,28] published a multilingual dataset of recent press articles. They used TF-IDF vectors with unsupervised algorithms to group articles related the same news events. The newsLens news tracking system [17] also benefits from TF-IDF vectors to cluster documents. The passage of time is materialised by time buckets: it assumes that articles close in time may relate to the same events [31]. Hence they discovered that buckets of 6 days, with 50% overlap between them are the most suitable parameters to treat group of news articles. Some time after, Miranda et al. [22] introduced and described a supervised and streaming algorithm able to cluster press articles into coherent mono-lingual and multi-lingual news stories. A more recent study [32] analysed the impact of vectorisation for news tracking algorithms and concluded that TF-IDF vectors provide competitive results and outperform dense vectors computed with doc2vec [18].

The code of implementations are rarely released but the algorithms are well described and datasets shared with the community to simplify the reproduction of experiments [17,22]. Miranda et al. [22] shared their implementation of their algorithm. In this paper, we will enhance this implementation with a new API and we will implement a baseline proposed in other research articles [21,32,28].

About the experiments performed on short messages, the propagation of the information on Twitter is studied since 2010 [13,27,33,23]. Researchers focused on tweets to track events discussed on the network. Tweets are short messages published on this social network, originally 140 characters long, 280 since 2017 [29]. Most of them used vectorisation to represent documents [24] while others used Twitter specific features, such as hashtags, internal links, followers or retweets to characterise tweets [4]. In this paper, we analyse two datasets from which we only keep news claims, that is to say, news article titles shared on the network, ignoring users reactions.

The scientific literature lacks news articles annotated in clusters of topics or stories. In the context of the pandemic, we took advantage of news article titles published on Twitter that are linked to fact checking services. It allows to know which articles are connected to the same event.

3 Datasets

Our task consists of building stories from documents written in natural language. We deal with article titles, which are short in size and contain a little amount of information. A suitable dataset for the task of tracking news stories has to provide events or clusters identifiers. In addition to generally used datasets [28] in this field, the emergence of Covid-19 datasets from Twitter with references to fact checking services such as PolitiFact[5] is the opportunity to carry out experiments with publicly available resources.

Our experiments focus on three available datasets, upon which we present relevant statistics in Table 1:

[5] `https://www.politifact.com/`

Table 1: Statistics about the datasets chosen for the experiments.

Dataset	Language	Partition	Documents	Tokens in documents		Nb. of clusters	Cluster size	
				Avg.	Std.		Avg.	Std.
Event Registry	eng	Train	12,233	56	19	593	21	32
		Test	8,726	58	19	222	39	89
CoAID	eng	Train	72,045	179	82	375	192	146
		Test	32,100	214	79	125	257	163
FibVid	eng	Train	988	206	77	51	19	7
		Test	402	201	79	52	8	2

- **Event Registry** [28,10]: a widely known news tracking dataset that has been used in various recent researches [22,32,20] to tackle the issue of discovering news stories in press articles. It comprises events reported in multiple languages: English, German and Spanish and was collected in 2014 and 2015. It is composed of full article texts and titles. In our experiments, we only keep the title of each article.
- **CoAID** [14,9]: a Twitter dataset with Covid-19 related tweets written in English. It has been gathered during the first months of the Covid-19 pandemic, from January to May 2020. We keep only the 500 biggest clusters, as they capture 77% of all documents (104.145 tweets). It comprises news claims and user reactions. The first are, as authors describe, links to news websites and the tweet text is the title of the article. We ignore user reactions.
- **FibVid** [16,11]: another Twitter dataset with Covid-19 related tweets. It focuses more on users reactions, but similarly to CoAID, tweets reporting news are connected to news claims identifiers. It was built in 2020. Similarly to CoAID, news claims are retained, user reactions are ignored.

In the Event Registry dataset, each document is associated with a ground truth cluster identifier. Not the two others. For them, each tweet is connected to a news claim URI and we use it as a label for clustering analysis. This way, tweets connected to the same URI are considered within the same cluster, so within the same news story. The number of clusters given in Table 1 is the number of distinct clusters, so news stories, given by the labels in the datasets. The issue with Covid-19 outbreak and the struggle against the *infodemic* resides in detecting false or misleading information in news as they emerge. We twist the purpose of these corpora to apply them to news story tracking.

3.1 Represent Document with Multiple Features

Documents are represented with vectors of numbers in order to be compared and processed by computers. In most cases, vectors are computed using the TF-IDF weighting scheme. Sparse vectors of numbers are a strong baseline [32], compared to dense representations, to track documents that report similar stories. However, these conclusions are valid for full articles - not only titles - and use a doc2vec [18] dense vectorisation. Recent advances, with the introduction of the

Transformer architectures provided new methods to represent documents, such as BERT [15,1] or XL-NET [35]. On an attempt to focus on sentences rather than tokens and to capture sentence information, the Sentence-BERT representation has been introduced [26]. In this paper, we propose both to compare algorithms and the relative impact of document representation. To that extent, we use four representations to encode documents, two of them are sparse while the other two are Transformer-based dense vector representations.

Sparse document representation Usually, TF-IDF weights are computed with the train part of the datasets. Here, we consider documents are handled in a stream and they are unknown before being processed. TF-IDF weighting models then have to be trained before processing data. We build them with huge sets of documents, independent from the data to weight. Different sizes of input documents used to feed the TF-IDF models give different weights. Hence, logically, they will produce different algorithm results.

To evaluate the impact of vectorisation, we propose to use different sets of documents to fit TF-IDF models and compute document vectors. We collected two: one with news articles, the other one with tweets [12]. News articles come from the Deutsche Welle (DW) website[6], which is scrapped to extract the title and body. DW is one of the only website that provides content in multiple languages and that is free to query and download. The other one is a collection of tweets, published in English from institutional press accounts. We manually chose press agencies or newspaper that publish on Twitter. There is an API limitation and we can only download 3200 tweets per source[7]. To overcome this problem, we selected a high number of press accounts. With this paper, we share the news articles and tweets identifiers with the code that weights the documents of each dataset listed in Section 3.

In addition to computing our own vectors, we use the pre-computed ones published by Miranda et al. [22] for the Event Registry dataset. We use them to have results comparable to previous works. In their paper, each document is characterised by several vectors of features associated to the text. This means for each document there are several TF-IDF vectors: one for the tokens, one for the lemmas and one for the named entities. For a news article which has a title and a body, there are at least six vectors: three for the title and three for the body. With titles we deal with only three vectors: the title tokens, lemmas and entities. To compute the respective weightings, we fit three different TF-IDF models. One will weight the tokens, another one the lemmas and the third one the entities. For all datasets of Section 3, we compute TF-IDF vectors using both sets of documents we collected: news and tweets.

To extract tokens, lemmas and entities from the titles, we use the spaCy software[8]. We only keep `GPE`, `ORG` and `PER` entities, as in Miranda et al. works [22]. To give an idea of the sizes of the TF-IDF datasets we used to weight

[6] `https://www.dw.com`
[7] Twitter API v1: `get-statuses/user-timeline`
[8] `https://spaCy.io` v3.2.1 with medium size models in English

Table 2: Statistics on the content of the sets of documents used to fit TF-IDF weighting models.

Dataset	Language	Documents	Number of Unique Features in the Sets		
			Tokens	Lemmas	Entities
News	eng	79,856	13,135,162	12,205,181	881,298
Tweets	eng	55,792	546,625	544,538	49,540

the document features, some statistics are shown in Table 2. The software to compute weights is freely shared over the Internet [6].

Dense Document Representation We use the Transformer architecture, especially the S-BERT [26] algorithms and models to encode the title texts into dense vectors. We select pre-trained models that focus on semantic similarity to compute title vectors. Among all the proposed models, we retained multilingual models with the highest scores in semantic search, available at the time of the experiments. They are `distiluse-base-multilingual-cased-v1` and `paraphrase-multilingual-mpnet-base-v2`. To simplify the remainder of the paper, we will respectively name them USE and MPNet.

With the two models, we encode the title texts into vectors of different sizes, 512 logits for USE, 768 for MPNet. While the cardinality of TF-IDF vectors is equal to the number of unique tokens, lemmas or entities found in the text, dense vector representation encodes documents in vectors of a fixed size. Contrary to the sparse TF-IDF document representation, we do not tokenize or extract entities from the text and encode the full sentence without any kind of pre-training. There remains a unique vector that encapsulates the whole text, instead of three with TF-IDF. Refer to [5] to encode texts with dense models.

4 Tracking Documents Reporting the Same Stories

To build news stories with short documents, we use the publicly available tracking algorithm proposed by Miranda et al. [22]. This is a supervised algorithm that dynamically creates clusters from incoming documents. In case there is no training data to create a clustering model, we propose another implementation of a news tracking algorithm based on K-Means.

4.1 Streaming Algorithm to Build News Stories

In this section, we provide more detailed explanations about the Miranda et al. algorithm we use in this article. This latter handles the documents of the dataset as a stream and each incoming document is compared to every existing cluster in the pool of already known clusters. Each existing cluster is a candidate in whom the document might be added if its similarity with the cluster is over a specific threshold T_1. If multiple candidates exist, the one with the highest

similarity wins. On the contrary, if no candidate exists, a new cluster is created accordingly. The algorithm handles heterogeneous data with vectorised texts and with timestamps of documents and clusters. The similarity measure $sim(d, C)$ between a document d and a cluster C is detailed below.

First of all, for dates comparison, clusters keep track of two dates: a lower bound, with the oldest document date in the cluster, and a higher bound, with the most recent one. To compute time similarities, a Gaussian distribution is used with $\mu = 0$ and $\sigma = 3$. This latter parameter is to be seen as a number of days after which the similarity falls dramatically.

$$f(d_{date}, C_{date}) = \phi_{\mu,\sigma^2}(|d_{date} - C_{date}|) \tag{1}$$

After that, we compute text similarities. The cosine measure $(\theta(d^k, C^k))$ computes the similarity between the representative vectors (TF-IDF or dense) of the document and the cluster. The cluster features are the average of all the documents vectors it is composed of. $K = 3$ stands for the tokens, entities and lemmas vectors, as described in Subsection 3.1 for TF-IDF. With dense representations, $K = 1$. There are two time similarities with the lower and upper bounds. In the Equation 2, β acts as a logistic regression coefficient, α as the intercept. β_k balance the importance of features in the final similarity score.

$$g(d, C) = \sum_{k=0}^{K} \beta_k \times \theta(d^k, C^k) + \sum_{k=0}^{K=2} \beta_k \times f(d_i^k, C_i^k) + \alpha \tag{2}$$

To flatten the similarity scores within the $[0 : 1]$ interval, we put $g(d, C)$ into a sigmoid function, in compliance with the logistic regression. The final similarity is given by Equation 3.

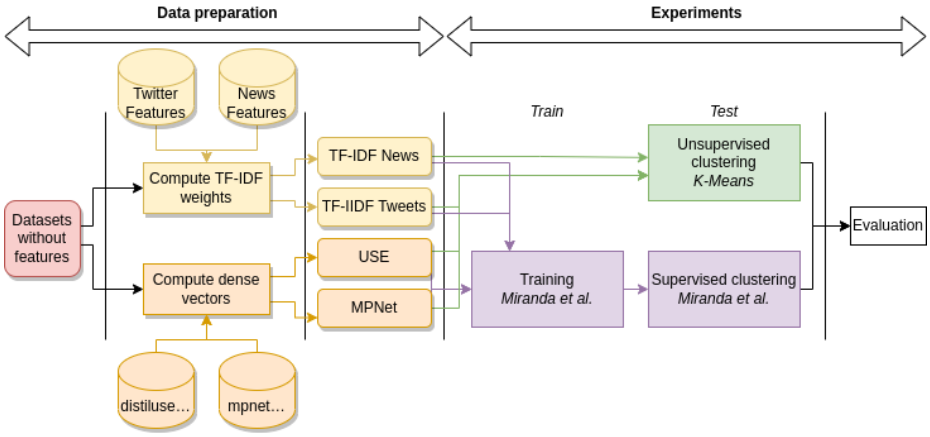
$$sim(d, C) = \frac{1}{1 + \mathrm{e}^{-g(d,C)}} \tag{3}$$

Model coefficients β and threshold T_1 are trained with a logistic regression on the train part of the corpus. True label clustering is computed on the train part from which we keep all document - cluster similarity scores. To lower the number of negative examples, we keep, for each document - cluster comparison, the twenty highest negative similarity scores. A grid search gives the best model and the decision threshold T_1 is the one that maximizes F1 on the train set.

4.2 Unsupervised News Tracking with K-Means

In accordance with suggestions of previous authors [21,32,28] we use the K-Means algorithm as an unsupervised method to create news stories, with cosine similarity as the distance measure. We propose this algorithm to counter the lack of training data, which are rare in this field of research. To simulate the time that passes, the dataset is split into buckets of sliding windows [21,17]. We use the *newsLens* optimal parameters given in Section 2 with a window of 6 days. We try different configurations, as K-Means is an unsupervised algorithm. An

Fig. 1: Description of the whole process, from the documents in the datasets without features to the clustering evaluation. Two key parts are noticeable: dataset features computation and algorithm evaluation.



optimal one for which we give the algorithm the true number of clusters for each window. The other one uses the Silhouette score to identify a coherent number of clusters. Early experiments implemented the *elbow method* that generated a too high number of cluster per window, providing unusable results. As it is necessary to compute numerous configurations in order to select the *optimal* number of clusters k, the time to compute clusterings may be very high for windows with lots of documents.

5 Experiments

With this paper, we release our implementation of the algorithms [7] and the training software [8] written in Python. It is a package with its own API. We intend to fulfil the lack of an end-to-end tool that builds news stories from documents. To the best of our knowledge, it does not exist yet.

The experiments consist of applying the algorithms mentioned in Subsection 4.1 with the datasets of Section 3, as described in Figure 1. To evaluate the cluster results and coherence, standard and BCubed [3] evaluation metrics are computed. The latter is a more accurate evaluation method of clustering performance.

First, we run experiments with the algorithm developed by Miranda et al. [22] and the results are shown in Table 3. The three datasets are tested with document vectors computed with the sparse and dense models described in Subsection 3.1. CoAID and FibVid are not multilingual, so we only focus on the English language in the Event Registry dataset. For the news story tracking results, the precision is good with Event Registry and CoAID. The low recall is correlated with a high number of clusters. By creating more clusters, the system focuses on precision,

Table 3: Experimental results of the Miranda et al. [22] algorithm on all the datasets described in Section 3. Sorted according to F1 BCubed score.

Corpus	Vectors	Standard			BCubed			Clusters		Time
		F1	P	R	F1	P	R	Real	Predicted	
Event Registry	**MPNet**	**54.50**	**90.00**	**39.00**	**74.30**	**85.60**	**65.70**	222	**362**	**00:05:17**
	News	74.80	86.10	66.10	72.30	72.10	72.60		206	00:01:45
	Miranda	61.80	98.20	45.10	73.00	95.90	59.00		902	00:02:33
	USE	46.50	91.00	31.20	68.80	89.70	55.90		644	00:08:10
	Tweets	52.90	96.90	36.30	65.50	93.30	50.50		1,154	00:03:10
CoAID	**Tweets**	**54.70**	**65.90**	**46.70**	**61.70**	**80.60**	**50.00**	125	**6,356**	**01:04:32**
	News	50.50	64.00	41.70	57.60	77.90	45.70		6,621	01:09:38
	USE	12.50	23.50	8.50	20.90	64.00	12.50		13,628	14:29:02
	MPNet	3.20	34.70	1.70	6.90	80.40	3.60		21,826	22:57:34
FibVid	**Tweets**	**30.40**	**33.40**	**28.00**	**41.20**	**48.20**	**36.00**	52	**98**	**00:00:02**
	USE	29.10	31.50	27.10	41.20	49.30	35.90		116	00:00:04
	MPNet	24.70	20.20	31.80	39.60	39.40	39.80		85	00:00:03
	News	19.40	26.70	15.20	37.80	71.70	25.60		207	00:00:02

hence decreases the recall. Results are constantly bad with Fibvid, which seems not to be a very suitable dataset for this task: it comprises a low number of news claims over the number of user reactions (220 K) we eliminated. We also tracked the necessary time to process the datasets that depends on two factors: the number of documents and the number of clusters found by the algorithm. As we previously mentioned, each incoming document is compared to every existing cluster and as a consequence, a high number of clusters increases the processing time.

When analysing the results of Miranda et al. algorithm on the three datasets, we first notice the F1 scores are all below 75% and the precision is always very high compared to the recall. There is no clear trend in favour of a specific document representation. For Event Registry, MPNet representation is way higher than its competitors, the second one being the TF-IDF document representation based on the News TF-IDF corpus. With CoAID, we have the right opposite, MPNet is the worst one while the TF-IDF representation based on the Tweets TF-IDF corpus, then on the News one give close results. In addition to this clear distinction between Event Registry and CoAID vectors, the algorithm on FibVid behaves differently. The dense and Tweets TF-IDF representations give very close results in terms of precision and recall, while the News TF-IDF vectorisation is low.

On another hand, in Table 4 we report the results computed with the unsupervised algorithm. We notice the documents are clustered together with a high precision. The low recall, so low harmonic mean is explained by the high number of clusters found, a consequence of time windows. We encounter a similar but lesser phenomena with the other algorithm. With K-Means, the document vectors that produce the best clustering scores are not the same as in the other experiment. A noticeable point is that the unsupervised method is not able to cluster documents as well as the Miranda et al. [22] algorithm in all situations. Furthermore, the processing time with K-Means is incredibly high. The Silhou-

Table 4: Experimental results running the K-Means baseline on all the datasets described in Section 3. Method **T** stands for true number of clusters, **S** for Silhouette. Sorted according to F1 BCubed Silhouette score.

Corpus	Vectors	Method	Standard			BCubed			Clusters		Time
			F1	P	R	F1	P	R	Real	Predicted	
Event Registry	**MPNet**	**T**	**71.00**	**97.70**	**55.70**	**70.40**	**88.80**	**58.30**	**301**	**301**	**00:33:31**
		S	**62.40**	**78.10**	**51.90**	**74.00**	**80.00**	**68.80**	**301**	**228**	**18:18:53**
	USE	T	70.80	97.00	55.80	70.40	88.40	58.50	301	301	00:34:28
		S	54.40	68.30	45.20	71.40	78.80	65.20	301	196	10:45:14
	News	T	66.50	87.20	53.80	63.60	81.60	52.00	301	301	00:02:14
		S	31.40	57.30	21.60	57.90	76.40	46.70	301	259	02:22:49
	Miranda	T	66.70	86.90	54.20	63.50	81.30	52.10	301	301	00:02:13
		S	31.90	58.50	22.00	57.90	76.60	46.50	301	291	02:22:51
	Tweets	T	67.20	87.60	54.50	63.70	81.40	52.30	301	301	00:02:14
		S	32.10	55.90	22.50	57.40	75.00	46.50	301	234	02:21:46
CoAID	**News**	**T**	**34.10**	**49.70**	**26.00**	**35.10**	**66.40**	**23.80**	**965**	**965**	**00:31:49**
		S	**33.70**	**46.70**	**26.30**	**35.10**	**63.90**	**24.20**	**965**	**655**	**105:22:08**
	Tweets	T	34.00	45.40	27.20	36.60	68.80	24.90	965	965	00:28:46
		S	30.90	59.30	20.90	30.60	67.90	19.70	965	768	88:07:17
	MPNet	T	17.80	42.00	11.30	19.40	53.80	11.80	965	965	00:39:11
		S	12.70	24.90	8.60	16.00	43.40	9.80	965	688	01:37:36
	USE	T	22.10	48.70	14.30	23.20	60.10	14.40	965	965	00:32:52
		S	11.80	43.50	6.80	14.50	55.10	8.40	965	1,410	04:39:42
FibVid	**MPNet**	**T**	**24.70**	**62.20**	**15.40**	**38.40**	**79.70**	**25.30**	**224**	**224**	**00:05:46**
		S	**23.90**	**24.20**	**23.60**	**35.00**	**38.00**	**32.40**	**224**	**65**	**00:27:41**
	USE	T	25.30	61.00	16.00	38.70	78.80	25.60	224	244	00:05:32
		S	23.50	23.50	23.40	34.20	36.00	32.60	224	62	00:26:51
	News	T	21.70	54,20	13.60	36.10	76.40	23.60	224	224	00:00:54
		S	19.60	19.60	19.60	32.80	37.20	29.30	224	68	00:03:15
	Tweets	T	19.80	55.00	12.10	34.70	75.90	22.50	224	224	00:00:54
		S	19.30	19.00	19.60	32.80	37.20	29.40	224	68	00:03:13

ette coefficient process computes every possible clustering from 1 to the number of documents in the window.

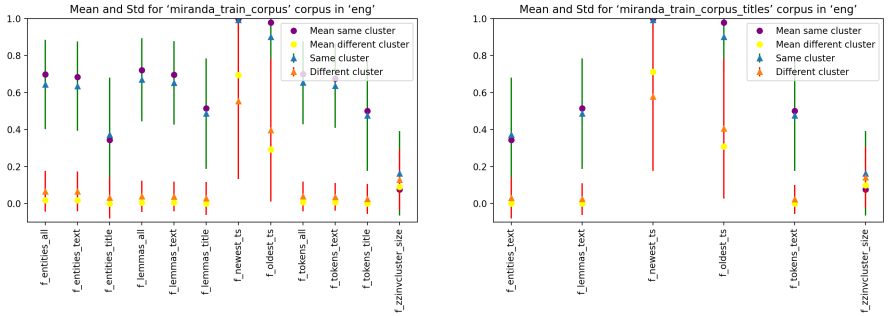
With this algorithm and Event Registry, the dense MPNet representation over-performs the sparse TF-IDF vectorisations (with between 9 and 17 points less of F1 compared to News TF-IDF, between 4 and 11 points for Tweets TF-IDF). The difference between the gold clustering results and those given with Silhouette are closer with dense vectors. With CoAID, sparse TF-IDF vectors are better for clustering the dataset, in this case also, the results are close between the gold and silhouette method. With this dataset, dense vectorisation is not an option. Finally, with FibVid, the algorithm behaves similarily to the other one, and the results are close to each other with a F1 score of about 35%. With this last dataset, there is a huge gap between the precision obtained using the gold number of clusters and when discovering a number of clusters k with the Silhouette score.

In all cases, the Miranda et al. [22] algorithm proceeds better and faster than the baseline, even if we include the training time, that is, in this scenario, four times the testing duration.

6 The Issue of Short Message Similarities

Results published in Table 3 and Table 4 are low in comparison to other studies that processed the Event Registry dataset for the same task [22,32,20]. In these works, authors use the whole text and title of the article. Our approach focuses on news article titles, and we wonder how well it is possible to apply news tracking algorithms on this specific type of documents.

We compare the document - cluster similarities described in Subsection 4.1 with two types of datasets: one with titles (the one we are using in this paper) and one that also includes article content text. We display in Figure 2, the cosine similarities mean and standard deviation for document - cluster pairs belonging to the same news story (in green) or not (in red). This means for the feature *f_entities_all* in Figure 2a, the mean similarity for this feature is around 0.65 for documents and clusters related to the same story, and almost 0 for documents and clusters that belong to different stories. The bigger the separation between green and red is, the more efficient the algorithm can be.



(a) With full articles, the article text gives a better separation of true and false clustering.

(b) With only titles, the dimensions that best discriminate documents and clusters are absent. There are less features.

Fig. 2: Document - cluster similarities on the training set with gold labels.

Independently of the algorithm itself, whether Miranda et al. (Table 3) or K-Means (Table 4), the very nature of data is at stake. Considering only news article titles does not permit to separate well documents that are dissimilar; and to well cluster ones that report a same news story.

7 Discussion

As we showed in our experiments, it is possible to obtain rather good clustering results with a very little of information contained in article titles. On the other hand, we wanted to evaluate the impact of the document representation, with sparse TF-IDF weightings and dense vectors. It is impossible to conclude on a

general trend that would allow us to give a recommendation on whether using one instead of the other. Our results may be considered as illogical, as it is reasonable to state that dense vectors should perform better in any circumstance as they better capture the context, over TF-IDF vectors that are necessarily shorter because of the limited number of tokens in article titles. Our suggestion, when tracking news stories in short messages, especially articles titles is to always test multiple document representations on the dataset in order to select the one that performs best. Even if the conclusion of a previous study [32] mentioned the pertinence of sparse vectors over dense ones, we assume this conclusion does not apply here. They handled full article content and similarly to the Transformer architecture, here only the 512 first tokens are used to compute the logits. In our case, the article title is always shorter than 512 tokens and the dense vectors represent the whole text. For full articles, longer than 512 tokens, it signifies deleting the rest of the text and removing pertinent information.

8 Conclusion

In this paper, we published an analysis of news propagation with tweets and articles titles coming from public sources. We tackled the issue of applying news tracking algorithms on short documents: article titles. We took advantage of the Covid-19 *infodemic* to twist the purpose of datasets dedicated to true and false news detection. We proposed to use the supervised algorithm released by Miranda et al. [22] to build stories from tweets when there are training data. For cases when they are missing, K-Means is a suitable unsupervised algorithm. We experimented the impact of document representation, with sparse TF-IDF vectorisations based on two corpora, and dense vectorisation with the Transformer architecture. We showed that the representation of a document is a major issue sometimes neglected in the literature. With this article, we release all resources: the code of the algorithms and the sets of data we collected to vectorise documents. We share with the community our implementations to let anyone reproduce our results and experiment with private datasets.

We showed one of the reason why clustering article titles works worse than when also taking the article content into account. Short messages do not contain enough discriminant data. We also lacked big datasets qualitatively annotated with events. By analysing Fibvid, we notice the quality of primary data could be the explanation of the rather bad results it produces.

On another hand, we computed the dense vectors with multilingual models [26]. These vectors are aligned in multiple languages, providing similar vectors for similar semantic in different languages. We will run new experiments on the Event Registry dataset, which has a set of similar events reported in multiple languages. We expect to notice new outcomes in multilingualism for this type of task.

Acknowledgments

This work has been supported by the European Union's Horizon 2020 research and innovation program under grants 770299 (NewsEye) and by the ANNA project funded by the Nouvelle-Aquitaine Region. The research work by Paolo Rosso was partially funded by the Generalitat Valenciana under DeepPattern (PROMETEO/2019/121). The authors would like to thank the Polytechnic University Of València (UPV)¨, Spain, which made this work possible, and its IT laboratory, DSIC.

References

1. Ai, M.: BERT for Russian news clustering. In: Proceedings of the International Conference "Dialogue 2021". p. 6. Moscow, Russia (Jun 2021)
2. Allan, J.: Introduction to Topic Detection and Tracking. In: Topic Detection And Tracking: Event-based Information Organization, pp. 1–16 (2002)
3. Amigo, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. Information Retrieval **12**(4), 461–486 (2009)
4. Atefeh, F., Khreich, W.: A Survey of Techniques for Event Detection in Twitter. Computational Intelligence **31**(1), 132–164 (Feb 2015). https://doi.org/10.1111/coin.12017
5. Bernard, G.: compute_dense_vectors, `https://archive.softwareheritage.org/swh:1:dir:7b4552980670d658ab07e5458d8f3ee1956aae4b`
6. Bernard, G.: compute_tf_idf_weights, `https://archive.softwareheritage.org/swh:1:dir:76f1022d1380e5f1d39ba02924e9f8eb9906dd95`
7. Bernard, G.: document_tracking, `https://archive.softwareheritage.org/swh:1:dir:e51ab63fd7dcfa830773c8cdfe40979d64a63133`
8. Bernard, G.: news_tracking, `https://archive.softwareheritage.org/swh:1:dir:efa67f09d67b843a1a2a6f3cdac5aac96a46da9a`
9. Bernard, G.: CoAID dataset with multiple extracted features (both sparse and dense) (Jun 2022). https://doi.org/10.5281/zenodo.6630405, `https://doi.org/10.5281/zenodo.6630405`
10. Bernard, G.: Event Registry dataset with multiple extracted features (both sparse and dense) (Jun 2022). https://doi.org/10.5281/zenodo.6630367, `https://doi.org/10.5281/zenodo.6630367`
11. Bernard, G.: Fibvid dataset with multiple extracted features (both sparse and dense) (Jun 2022). https://doi.org/10.5281/zenodo.6630409, `https://doi.org/10.5281/zenodo.6630409`
12. Bernard, G.: Resources to compute TF-IDF weightings on press articles and tweets (Jun 2022). https://doi.org/10.5281/zenodo.6610406
13. Brigadir, I., Greene, D., Cunningham, P.: Adaptive Representations for Tracking Breaking News on Twitter. arXiv:1403.2923 [cs] (Nov 2014), `http://arxiv.org/abs/1403.2923`
14. Cui, L., Lee, D.: CoAID: COVID-19 Healthcare Misinformation Dataset. arXiv:2006.00885 [cs] (Nov 2020), `http://arxiv.org/abs/2006.00885`
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] (May 2019), `http://arxiv.org/abs/1810.04805`

16. Kim, J., Aum, J., Lee, S., Jang, Y., Park, E., Choi, D.: FibVID: Comprehensive fake news diffusion dataset during the COVID-19 period. Telematics and Informatics **64**, 101688 (Nov 2021). https://doi.org/10.1016/j.tele.2021.101688
17. Laban, P., Hearst, M.: newsLens: Building and visualizing long-ranging news stories. In: Proceedings of the Events and Stories in the News Workshop. pp. 1–9. Vancouver, Canada (2017). https://doi.org/10.18653/v1/W17-2701
18. Le, Q.V., Mikolov, T.: Distributed Representations of Sentences and Documents. In: arXiv:1405.4053 [Cs] (May 2014), http://arxiv.org/abs/1405.4053
19. Leban, G., Fortuna, B., Brank, J., Grobelnik, M.: Event registry: Learning about world events from news. In: Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion. pp. 107–110. Seoul, Korea (2014). https://doi.org/10.1145/2567948.2577024
20. Linger, M., Hajaiej, M.: Batch Clustering for Multilingual News Streaming. In: Proceedings of Text2Story Co-Located with 42nd ECIR. vol. 2593, pp. 55–61. Lisbon, Portugal (Apr 2020), http://ceur-ws.org/Vol-2593/paper7.pdf
21. Mele, I., Bahrainian, S.A., Crestani, F.: Event mining and timeliness analysis from heterogeneous news streams. Information Processing & Management **56**(3), 969–993 (May 2019). https://doi.org/10.1016/j.ipm.2019.02.003
22. Miranda, S., Znotiņš, A., Cohen, S.B., Barzdins, G.: Multilingual Clustering of Streaming News. In: 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4535–4544. Brussels, Belgium (Oct 2018), https://www.aclweb.org/anthology/D18-1483/
23. Petrovic, S., Osborne, M., Lavrenko, V.: Streaming First Story Detection with application to Twitter. In: NACL 2010. pp. 181–189. Los Angeles, Californi, USA (Jun 2010), https://dl.acm.org/citation.cfm?id=1858020
24. Phuvipadawat, S., Murata, T.: Breaking News Detection and Tracking in Twitter. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. pp. 120–123. Toronto, AB, Canada (Aug 2010). https://doi.org/10.1109/WI-IAT.2010.205
25. Pouliquen, B., Steinberger, R., Ignat, C., Käsper, E., Temnikova, I.: Multilingual and cross-lingual news topic tracking. In: Proceedings of the 20th International Conference on Computational Linguistics - COLING '04. pp. 959–es. Geneva, Switzerland (2004). https://doi.org/10.3115/1220355.1220493
26. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410
27. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open domain event extraction from Twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12. p. 1104. Beijing, China (2012). https://doi.org/10.1145/2339530.2339704
28. Rupnik, J., Muhic, A., Leban, G., Skraba, P., Fortuna, B., Grobelnik, M.: News Across Languages - Cross-Lingual Document Similarity and Event Tracking. Journal of Artificial Intelligence Research **55**, 283–316 (Jan 2016). https://doi.org/10.1613/jair.4780
29. Sasank Reddy: Now on Twitter: 140 characters for your replies (Mar 2017), https://blog.twitter.com/en_us/topics/product/2017/now-on-twitter-140-characters-for-your-replies

30. Shearer, E., Mitchell, A.: News Use Across Social Media Platforms in 2020 (Jan 2021), https://www.pewresearch.org/journalism/2021/01/12/news-use-across-social-media-platforms-in-2020/
31. Shinyama, Y., Sekine, S., Sudo, K.: Automatic paraphrase acquisition from news articles. In: Proceedings of the Second International Conference on Human Language Technology Research. pp. 313–318. San Diego, California (2002). https://doi.org/10.3115/1289189.1289218
32. Staykovski, T., Barron-Cedeno, A., da San Martino, G., Nakov, P.: Dense vs. Sparse Representations for News Stream Clustering. In: Proceedings of Text2Story Co-Located with the 41st ECIR. vol. 2342, pp. 47–52. Cologne, Germany (Apr 2019), https://ceur-ws.org/Vol-2342/paper6.pdf
33. Weng, J., Lee, B.S.: Event Detection in Twitter. In: Proceedings of the Fifth International Conference on Weblogs and Social Media. pp. 401–408. Barcelona, Catalonia, Spain (Jul 2011), http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2767
34. World Health Organisation: Infodemic (Jan 2022), https://www.who.int/westernpacific/health-topics/infodemic
35. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv:1906.08237 [cs] (Jan 2020), http://arxiv.org/abs/1906.08237