data

MDPI

Article

News Monitor: A Framework for Exploring News in Real-Time †

Nikolaos Panagiotou *, Antonia Saravanou and Dimitrios Gunopulos

Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Panepistimioupolis, Ilissia, 15784 Athens, Greece; antoniasar@di.uoa.gr (A.S.), dg@di.uoa.gr (D.G.)

* Correspondence: npanagio@di.uoa.gr

† This paper is an extended vesion of "Saravanou, A.; Panagiotou, N.; Gunopulos, D. News Monitor: A Framework for Querying News in Real Time. In Proceedings of 43rd European Conference on IR Research, ECIR 2021, Virtual Event, 28 March–1 April 2021; pp. 543–548".

Abstract: News articles generated by online media are a major source of information. In this work, we present News Monitor, a framework that automatically collects news articles from a wide variety of online news portals and performs various analysis tasks. The framework initially identifies fresh news (first stories) and clusters articles about the same incidents. For every story, at first, it extracts all of the corresponding triples and, then, it creates a knowledge base (KB) using open information extraction techniques. This knowledge base is then used to create a summary for the user. News Monitor allows for the users to use it as a search engine, ask their questions in their natural language and receive answers that have been created by the state-of-the-art framework BERT. In addition, News Monitor crawls the Twitter stream using a dynamic set of "trending" keywords in order to retrieve all messages relevant to the news. The framework is distributed, online and performs analysis in real-time. According to the evaluation results, the fake news detection techniques utilized by News Monitor allow for a F-measure of 82% in the rumor identification task and an accuracy of 92% in the stance detection tasks. The major contribution of this work can be summarized as a novel real-time and scalable architecture that combines various effective techniques under a news analysis framework.

Keywords: news monitoring; news articles analysis; event detection; question answering

Citation: Panagiotou, N.; Saravanou, A.; Gunopulos, D. News Monitor: A Framework for Exploring News in Real Time. *Data* **2022**, *7*, 3. https://doi.org/10.3390/data7010003

Academic Editor: Giuseppe Ciaburro

Received: 8 November 2021
Accepted: 25 November 2021
Published: 27 December 2021

1. Introduction

There is an increasing amount of news agencies that provide news articles on a daily basis that cover events happening around the world. Nowadays, people are searching, consuming, sharing and commenting news in online news portals and social networks. Some of the events described in the news articles and posts are evolving over a large period of time, some of them have a short duration, others are localized and interest a smaller crowd, whereas others are more popular and interest a large crowd globally, such as the COVID-19 pandemic. The vast amount of news stories that get posted every hours from news portals make it challenging to the users to be able to identify and choose the latest stories and avoid reading about the same events. The readers struggle to learn about the daily news in a short period of time because of the vast increase in journalism from official news agencies and citizens who post in social networks. The above difficulties increase the need for automated systems that are able to identify, analyze, filter and recommend fresh news. This problem is widely studied in the literature and expands to multiple categories under the news systems implementation. This implementation is in the intersection of news collection, first story detection [1], events detection [2] and sub-event detection [3], news articles summarization [4], opinion mining [5], trends detection [6] and fake news detection [7].

The users, due to their limited time and to the large amount of available news articles, are sometimes only interested in the summary of an article. That is, a brief version of the

original article that is able to describe the event, as well as the sub-events (*highlights*) that reside within the main event. The problem is known as summarization in the literature, and its goal is to provide short document versions from the originals. In this work, we address the problem of summarization from two different perspectives; the first one is "extractive" summarization in the form of a graph, whereas the second approach is a generative approach utilizing existing "abstractive" summarization models, such as T5 [8]. At the same time, users use social media, such as Twitter, to actively share their views and opinion on the various topics that appear around news stories. Following the social media discussions can give more insights on the users' pulse, different views and details about each story line. As a result, it is critical for tools to be developed that can provide the following functionalities: (i) summary of the news stories that talk about the same event, and (ii) analysis of the related social media streams.

Furthermore, social media users commonly tend to share news as they happen. At other times, however, users may share rumors and fake news. Whereas major news agencies, such as CNN and BBC, have well-established editorial teams that verify the news, the content on social media is completely unverified. Thus, automated systems for rumor detection and verification are very important for social media platforms.

All of the above issues provide a strong motivation to utilize the recent breakthroughs and techniques in natural language processing in order to provide an automated tool that helps the readers to quickly navigate through the articles and, at the same time, provides them with insights on the opinions on social media, such as Twitter. The long-term vision of this automated tool, however, is also to organize events as they occur in order to automatically maintain an events database, such as Wikipedia Events Portal.

However, there are plenty of challenges that need to be addressed in order to effectively design such a tool. The major challenge is the volume of data and the need for real-time analysis. This is even more challenging given the velocity of the data. News agencies all over the world constantly generate new content for a variety of topics, ranging from local social incidents to global events. Recent algorithms have significantly improved the accuracy of the text-mining techniques, but, given the volume of the data, a trade-off between accuracy and performance is necessary. Finally, news content is also generated from users (e.g., in Twitter) and, thus, the veracity challenge arises. Notably, due to the veracity challenge, the fake news detection research field significantly increased in popularity in recent years. As a result, all of the major challenges, also known as the five Vs of analyzing big data, must be addressed in order to implement such a tool.

In this paper, we describe our proposed system, News Monitor[1] [9], that works in a distributed setting. The main contribution of this work is the proposal of a scalable and distributed architecture that combines state-of-the-art techniques for analyzing news, while, at the same time, an events knowledge base is automatically created and maintained. The novel aspects of this work can be categorized into the following aspects:

- Scalability: Our system collects news stories from over 500 RSS streams in real-time. We analyse the collected news stories corpus using limited hardware and randomized data structures in constant space and time complexity. Our system architecture is elastic and distributed.
- Usability: Our proposed system allows the users to quickly explore a news story via a user friendly interface. In addition, our system constructs a knowledge base (KB) from all the news articles and they users can search directly in the KB.
- Novelty: Our system is implemented using a variety of state-of-the-art techniques under a unified solution.
- Comparison: Our proposed system, News Monitor, extends the functionality of the existing frameworks and provides extra tools for analysis such as online question answering, graph summarization, sentiment analysis and fake news detection.

2. Related Work

In this section, we present the advancements in research areas related the components of News Monitor. The corresponding components are relevant to: (i) first story detection, (ii) question answering, (iii) relation extraction, (iv) summarization and (v) fake news detection. A table summarizing the main related works can be found in Table 1. Moreover, in this Section, we present the relevant platforms to News Monitor.

Table 1. An overview of the existing literature and our proposed methodology.

Mining Task	Relevant Methods	News Monitor
First Story Detection	UMASS [10], LSH-UMASS [11], PAR-UMASS [12], K-Term [13], Rel-EFSD [1]	Rel-EFSD [1]
Question Answering	PARALEX [14], SEMPRE [15], ParSEMPRE MemNets [14], BERT [16], T5 [8], ALBERT [17], DistiBERT [18], GPT-3 [19]	BERT [16]
Information Extraction	Ollie [20], ReVerb [21], Open-IE [22], TextRunner [23], RelNoun [24], CALMIE [25]	ReVerb [21]
Summarization	T-BERTSum [26], PEGASUS [27], T5 [8]	T5 [8]
Fake News Detection	TriFN [28], FakeNewsTracker [29], Tree CRF [30]	BERT [16]

2.1. First Story Detection

Focusing on the aspect of first story detection (FSD), a variety of techniques have been proposed in the literature. The baseline technique is the system UMASS [10], where the authors proposed a technique that solves the problem of nearest neighbor identification using the cosine similarity distance. This work has been extended by [31], where a system that combines the terms' similarity along with the similarity of the named entities and the topics yielded better results. Another work that utilizes a nearest neighbor variation combined with emotion analysis methods to detect specific events of disastrous weather conditions (i.e., floods) and how those have affected different regions of a map [32]. The work of [10] has been revised in [11], where the authors propose the usage of a locality-sensitive hashing (LSH) index in order to speed up the technique. Other works related to social media data have used LSH variations in order to reduce their runtime [33–35]. Later, in the work of Petrovic et al. [12], it was found that the usage of paraphrases when searching for the nearest neighbor increased the detection accuracy. From a different perspective, Karkali et al. [36] solved the problem in an online fashion by simply using the IDF scores of the terms, avoiding the costly nearest neighbor identification. Wurzer et al. [13] followed a similar approach to [36] and suggested a technique where the k-term hashing algorithm is used in order to online detect first stories. Moran et al. [37] extended the work of Petrovic et al. [12] by using Word2Vec [38] embeddings instead of syntactic paraphrases. Saravanou et al. [39] proposes a method that utilizes LSH and Word2Vec to reveal hidden links of text similarity in a social graph and detect events by tracking the very large connected components in this graph. Later, they extended this work in [3] to delineate the events to sub-events and provide a timeline of the highlights for each event for better exploration and understanding of the individual events and first stories. Finally, in our previous work [1], we described a general framework that (i) combines a variety of FSD techniques, (ii) uses sophisticated linguistic features and (iii) is scalable.

2.2. Question Answering

Another aspect that News Monitor aims to integrate is that of question answering. The question is written in the natural language, but, regarding the answer, two variants of the problem exist: (a) the answer originates from structured knowledge in the form of a knowledge graph and (b) the answer originates from a document (e.g., a document

span). For the first direction, the system PARALEX [14] aims to utilize a set of question paraphrases in order to link a question to a query. Then, the most relevant fact, extracted using ReVerb [21], is provided as an answer. Similarly, the authors in [15] aim to utilize large amounts of text in order to design the semantic parser SEMPRE, which maps questions to queries. A similar approach that, in addition, uses paraphrases is used in ParSEMPRE [40]. Finally, other approaches, such as [41] and MemNets [14], try to map both the question as well as the answer fact in an embedding space, and answer a query utilizing the distance of the embeddings.

The other direction where the answer is in the form of a span in the text is highly dominated by deep neural network approaches that utilize attention mechanisms and recurrent neural networks. These techniques include BERT, where a transformer is fine-tuned in order to select the appropriate start and end token of a document. ALBERT [17] is a more light version of BERT and DistiBERT [18] is a BERT version with significantly fewer parameters. T5 [8] addresses the task as a text-to-text problem, where the answer consists of a generated text. Finally, GPT-3 [19] is a language model with a significantly increased number of parameters that achieves a state-of-the-art performance in a large number of NLP tasks.

2.3. Information Extraction

Our proposed framework, News Monitor, depends on a robust relation extraction mechanism that is used in order to construct the knowledge base. Since we are interested in relations that are independent of a predefined taxonomy, we rely on open information extraction. These systems detect open domain relations by self-training over a massive corpus [23] or heuristic rules [20–22]. They allow for the development of very scalable systems. The relations extracted often have a generic format of two arguments that are connected by a verb. In our work, we use ReVerb [21] due to its efficiency and simplicity. Other related systems to ReVerb include OpenIE 4.1 [22], Ollie [20], RelNoun [24], SRLIE [42] and TextRunner [23]. Recent techniques, such as [43], address the problem using deep learning methods.

2.4. Summarization

News Monitor also depends on a robust summarization system. The summarization systems are classified into two categories: (i) extractive and (ii) abstractive. The former approaches extract chunks of text, whereas the latter are able to generate new text. The approach proposed in [44] describes a reinforcement learning summarization approach that optimizes the ROUGE metric. Zhang et al. [45] describe a latent approach for extractive text summarization, whereas [46] describe an approach based on BERT that performs both abstractive and extractive summarization. Srikanth et al. [47] use the existing BERT model to produce an extractive summarization by clustering the embeddings of sentences by K-means clustering. Ma et al. [26] propose a topic-aware extractive and abstractive summarization model named T-BERTSum. It focuses on pretrained external knowledge and topic mining to capture more accurate contextual representations. Finally, the PEGASUS [27] and T5 [8] transformer models are capable of performing abstractive summarization.

2.5. Fake News Detection

Fake news is one of the key problems that our proposed framework, News Monitor, addresses. Fake news detection techniques have become crucial to fight the extensive spread of harmful or negative effects on individuals and society, as fake news persuades people to accept biased or false beliefs. Shu et al. [29] have conducted a survey to present a comprehensive review of method for fakes news detection on social media, including characterizations on psychology and social theories and existing algorithms from a data mining perspective. Oshikawa et al. published a survey to summarize the techniques that focus and utilize natural language processing to detect fake news. The original RSS sources are high-quality news agencies. However, the Twitter stream that is available

to the system is of questionable quality. A fake news detection system, as suggested by the works of Zubiaga et al. [48] and Kochkina et al. [49], follows the following pipeline: (i) rumor detection, (ii) topic tracking, (iii) stance classification and, finally, (iv) verification. For rumor detection, a classification approach is described in [48]. The majority of works for fake news detection address the challenge of stance detection, including the works of [30,49,50]. Another work focused on the last aspect of topic verification is described in [51]. Shu et al. [28] propose a tri-relationship embedding framework (TriFN) that models publisher–news relations and user–news interactions simultaneously for fake news classification. Shu et al. [35] study the problem of understanding and exploiting user profiles on social media for fake news detection. Reis et al. [52] present a new set of features and measure the prediction performance of current approaches and features for the automatic detection of fake news. An overview of the related literature is available in Table 1.

2.6. Relevant Platforms

Similar systems that analyze document streams include a variety of works that focus on social media, such as Twitter. These include TwitterMonitor [6], Jasmine [53] and TwitterStand [54], which focus on detecting trending keywords and clusters of messages that correspond to global or local events. A detailed description of these systems is included in our previous work [2]. Focusing on commercial news monitoring platforms, Google News and Yahoo News are some of the leaders. Specifically, Google announced[2] that Google News already uses advanced language models, such as BERT, for purposes of fake news detection. Finally, a commercial platform, very relevant to News Monitor, that includes the tasks of event detection, among many other services using news, is Event Registry [55]. However, since this is a commercial product, we do not have access to all of its features, and a direct comparison is difficult.

3. Architecture

We have designed the News Monitor architecture with a distributed microservices perspective, specifically each module in our proposed architecture runs on a different machine. All components are integrated using the Apache Kafka message queue which allows the individual components to communicate with each other by subscribing to one or more topics in an efficient and fault-tolerant way.

Each component consists of a thread that subscribes to a Kafka topic and a thread that publishes to a Kafka topic. Due to their nature, some components, such as the Twitter fetcher, do not subscribe to any Kafka topic. In addition, each component periodically monitors and logs its state in order to inform News Monitor about the memory usage, and for issues processing the data.

We store the knowledge base (KB) and the analysis in a MongoDB instance. In order to ensure an optimal performance, the appropriate indexes are used in the MongoDB database. In addition, the raw text data, both for Twitter as well as for the news, are stored in an elastic search instance in order to effectively search using free text. The architecture of News Monitor is shown in Figure 1.

In Figure 1, we show two of the main components of our system the News Fetcher and the Twitter Fetcher which are responsible for crawling the news articles from the web. Notably, the news fetcher tracks a set of static RSS sources. On the other hand, the Twitter fetcher has an input from the trend detection module. That is, it dynamically tracks keywords relevant to trending entities from the news. The next component is the extractor, which receives HTML data and is responsible for extracting the main content. More specifically, the extractor transforms the HTML document into single text. The output of the extractor transfers to the pre-processor module. This module performs natural language processing on the input data, including tasks such as tokenization, named entity recognition and information extraction. The reader will notice that the open information extraction is actually a separate service that is called by the pre-processor module.

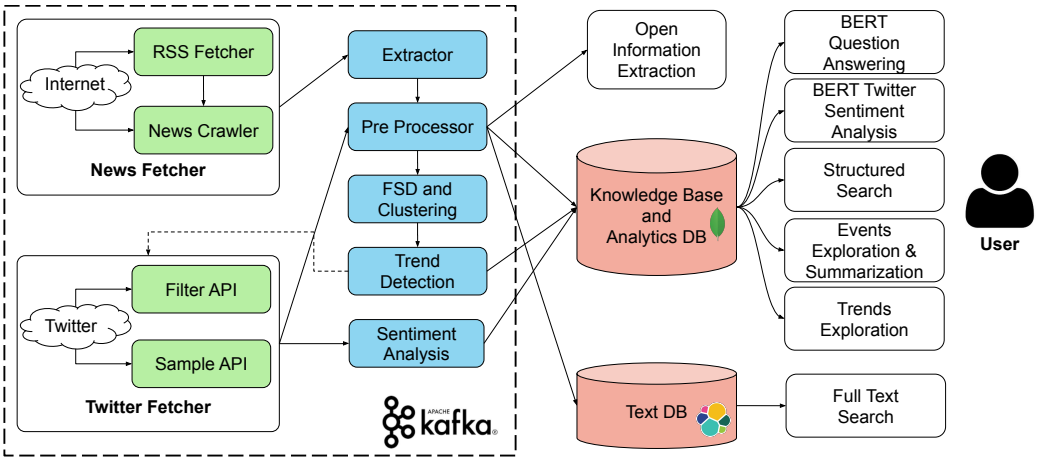


Figure 1. The architecture of News Monitor.

The next component is the first story detection (FSD) module, which is able to identify fresh news in order to form news clusters. The input for this module are the preprocessed documents. The documents include metadata, such as named entities, that are detected by the preprocessor module. The cluster information, together with documents, are forwarded to the trend detection module. The trend detection component monitors the clusters created by the FSD module and identifies trending terms and entities in real-time. Those terms are used by the Twitter fetcher in order to crawl relevant content to the news from the Twitter. The output of the Twitter fetcher is provided to the sentiment analysis module. Finally, the sentiment analysis component uses a deep learning model in order to identify the sentiment of the Twitter data. Then, it stores the tweets in the MongoDB database.

The stored data are used in order to provide to the end user a variety of services. The end user is able to view in real-time news clusters, as well as first stories. In addition, the user is able to view a summary for each of the articles and is also able to inspect the knowledge base extracted from each of the articles. Furthermore, the user can explore the news knowledge base constructed from the articles and, finally, the user can query an article in order to retrieve the answer using the BERT engine. All of the modules process data in real time and all of the services provided to the user retrieve the data from the databases. The question-answering service, however, apart from the databases, also uses a deep learning model that uses the user input (e.g., the questions).

A general view of the News Monitor interface is illustrated in Figure 2. The output provided by the different components such as first Story Detection is provided in Figures 3–5.

Hardware Requirements: News Monitor currently runs on two servers with 16 threads, and 32 GB of RAM. The various services are split into the two machines. In addition, the deep learning tecniques run on an NVIDIA 2070 SUPER GPU in one of the two machines. The web server layer is implemented in the Python Django framework.

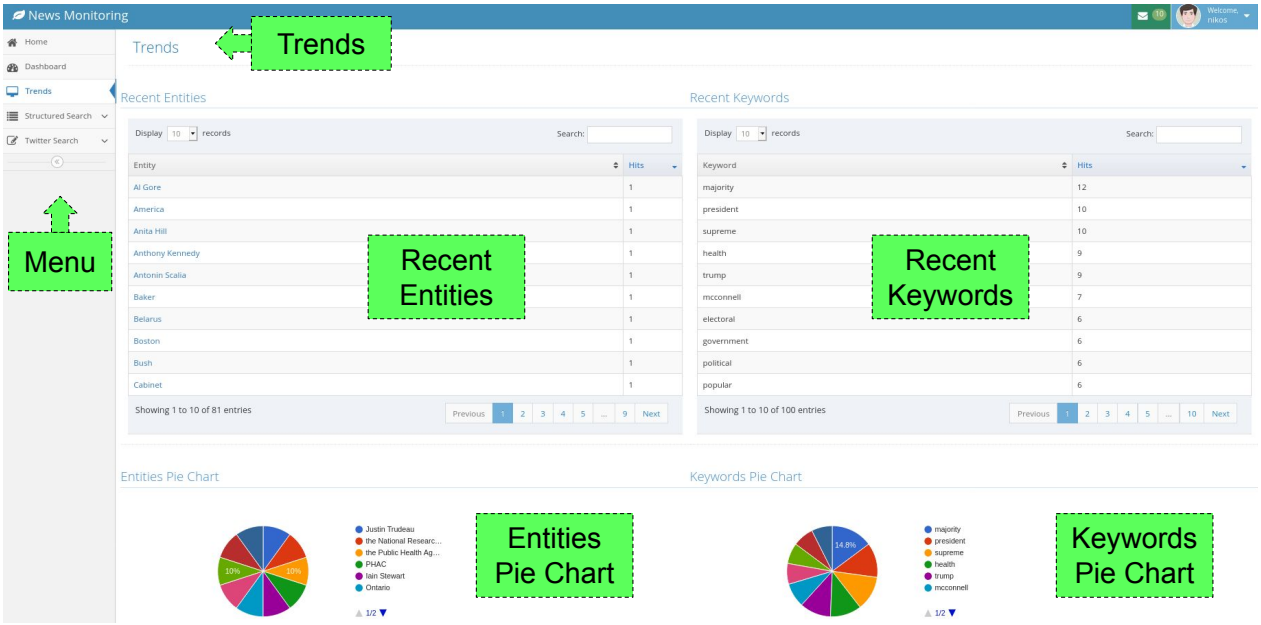
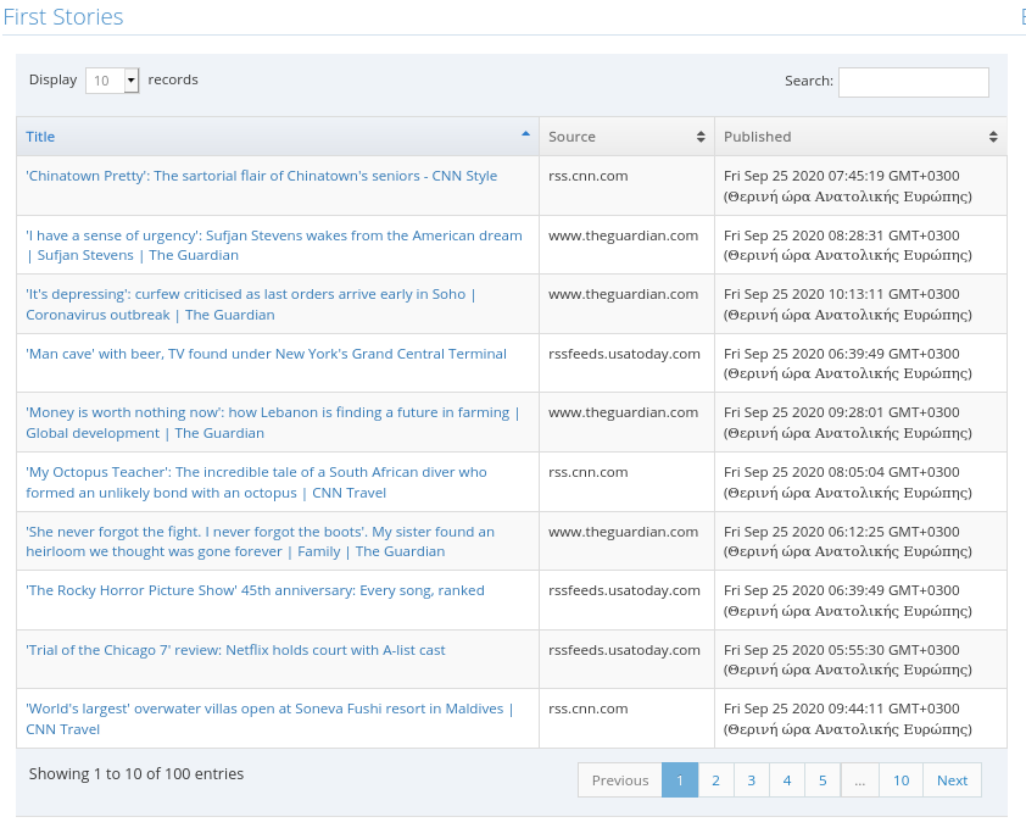


Figure 2. The News Monitor interface. The trends detection page is illustrated, which gives to the reader the general appearance of the system.



First Stories

Display 10 records Search:

Title	Source	Published
'Chinatown Pretty': The sartorial flair of Chinatown's seniors - CNN Style	rss.cnn.com	Fri Sep 25 2020 07:45:19 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'I have a sense of urgency': Sufjan Stevens wakes from the American dream | Sufjan Stevens | The Guardian	www.theguardian.com	Fri Sep 25 2020 08:28:31 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'It's depressing': curfew criticised as last orders arrive early in Soho | Coronavirus outbreak | The Guardian	www.theguardian.com	Fri Sep 25 2020 10:13:11 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'Man cave' with beer, TV found under New York's Grand Central Terminal	rssfeeds.usatoday.com	Fri Sep 25 2020 06:39:49 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'Money is worth nothing now': how Lebanon is finding a future in farming | Global development | The Guardian	www.theguardian.com	Fri Sep 25 2020 09:28:01 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'My Octopus Teacher': The incredible tale of a South African diver who formed an unlikely bond with an octopus | CNN Travel	rss.cnn.com	Fri Sep 25 2020 08:05:04 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'She never forgot the fight. I never forgot the boots'. My sister found an heirloom we thought was gone forever | Family | The Guardian	www.theguardian.com	Fri Sep 25 2020 06:12:25 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'The Rocky Horror Picture Show' 45th anniversary: Every song, ranked	rssfeeds.usatoday.com	Fri Sep 25 2020 06:39:49 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'Trial of the Chicago 7' review: Netflix holds court with A-list cast	rssfeeds.usatoday.com	Fri Sep 25 2020 05:55:30 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)
'World's largest' overwater villas open at Soneva Fushi resort in Maldives | CNN Travel	rss.cnn.com	Fri Sep 25 2020 09:44:11 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

Showing 1 to 10 of 100 entries Previous 1 2 3 4 5 ... 10 Next

Figure 3. First stories identified by News Monitor.

Records

Display 10 records Search:

User	Document	Sentiment	Timestamp
_charlesperry	RT @Kasparov63: The original version of a March 2017 article I wrote about the dangers of a Trump presidency contained the line, "Without f...	NEGATIVE	Thu Mar 11 52715 07:52:13 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
ABayer42	RT @briantylercohen: Funny how Trump is so against defunding the police, considering his taxes prove he's never actually funded them.	POSITIVE	Thu Mar 11 52715 07:38:33 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
AGCalhau	RT @RBReich: -The pandemic has killed 203,000 Americans so far. -Trump blew it. -At least the A...	NEGATIVE	Thu Mar 11 52715 08:29:33 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
ailsdive	RT @carolecadwalla: We knew it. We said it. Now @Channel4News has amazing cache of hard evidence. Facebook is a tool that Trump uses to...	POSITIVE	Thu Mar 11 52715 07:06:32 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
AlexanderTMWK	RT @ericgarland: New NYT story: The Apprentice helped Trump line deals up to save his bankrupt business...WITH THE RUSSIAN MOB "There were...	POSITIVE	Thu Mar 11 52715 06:52:00 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
amikegreen2	Michael Cohen says behind closed doors Trump is probably panicking: 'He's lost, he's confused, he's dazed' https://t.co/r9NOZmowT4	NEGATIVE	Thu Mar 11 52715 06:49:25 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)
amjohnson53	RT @RBReich: Undocumented people paid upwards of $20,000,000,000 in federal taxes in 2015. Donald Trump paid $0.	NEGATIVE	Thu Mar 11 52715 07:17:39 GMT+0200 (Χειμερινή ώρα Ανατολικής Ευρώπης)

Figure 4. Tweets identified by News Monitor. For each tweet, the sentiment is detected.

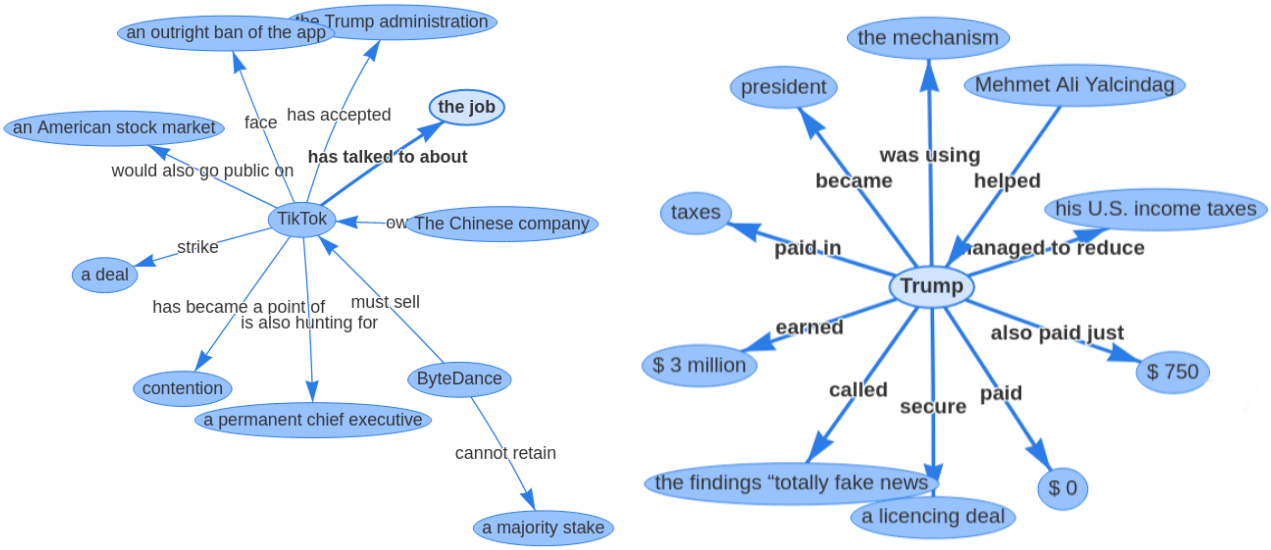


Figure 5. The summary of articles provided as a graph. The links represent interactions between entities.

4. News Monitor's Characteristics

In this section, we describe the characteristics of our proposed framework, News Monitor, to analyse and explore news articles. A user can view the first stories as a list and can also explore the event clusters. For each article and for each cluster, the user is able to view the content, the extracted knowledge base and the extracted summary as a graph. Furthermore, for each article the user can make a quick question in their natural language. If the answer exists in the document, the News Monitor will provide it to the user. Finally, the user can perform a search in the extracted knowledge base and in the relevant tweets.

Specifically, for the Twitter data, News Monitor performs a sentiment analysis using BERT. A detailed overview of the components follows in this Section.

4.1. News Fetcher and Twitter Fetcher

Two of the most important components of our system are the News and the Twitter Fetchers, they are responsible for collecting the news articles and social network's streams (input data). The news fetcher periodically monitors a list of RSS feeds. In order to effectively crawl the RSS feeds, the component uses the Python Scrapy framework. If a new RSS feed is detected, the news fetcher crawls the article, receives the status code and downloads the HTML document, as well as links to the article images.

The Twitter fetcher collects the posts (tweets) using the streaming Twitter API and tracks a dynamically selected list of keywords that is based on our trend detection component. The list of keywords is not known in advance. The keywords to be tracked are dynamically selected using the most referred named entities in the most recent articles. This is carried out in order to retrieve tweets relevant to the news. The two fetchers run in parallel. The Twitter fetcher starts with predefined keywords and is periodically updated.

4.2. Extractor

The output of the news fetcher component, as illustrated in Figure 1, is provided to the extractor component. This component is responsible for processing the HTML data and extracting the main content of the article automatically. It does so by using the python readability[3].

4.3. Preprocessing, Graph Summary and Knowledge Base Construction

For each news article that has been collected by the News Fetcher and has been processed by the Extractor module, we have an extracted content, which is then used by the Pre Processor module. The Pre Processor module performs various NLP tasks that include tokenization, dependency parsing, sentence splitting and named entity recognition. For these NLP tasks, we use the open source library Spacy[4].

For the open information extraction, we use the web scale library ReVerb[5] to get a list of triples of the form (subject, predicate, object) for each news article. The triples also have metadata that describe the type of entities that participate. For example, the metadata may suggest that the first argument ("subject") of the triple is a person, whereas the second argument ("object") is a geopolitical entity. Each triple with the metadata information is stored in the MongoDB Knowledge Base with the appropriate indexes in order to provide fast queries.

Then, we use these tuples to create a graph structure that represents the summary of a news article, an example is shown in Figure 5. In this graph, multiple references to the same entity are merged to the same node. The edges of the graph represent the actions ("predicates") and the nodes correspond to the actors ("subject" or "object") that participate. Finally, the connected components of the graph typically correspond to the sub-events of the article. We store all the pre-processed news articles and the relevant tweets in an ElasticSearch[6] database.

4.4. First Story Detection and Clustering

The First Story Detection component uses the data from the Pre Processor module. This module is based on the work of Panagiotou et al. [1] and detects for each story it is a first story or not. We have implemented multiple LSH indexes with capacity of 2K documents. This way we ensure the scalability of our system and that the processing time is constant. In addition, the module is implemented using vectorized operations in order to achieve a high performance.

In a nutshell, the module works as follows. For each new document, the nearest neighbor is identified. Then, similarity features are calculated from NLP elements, such as named entities and relations. Each similarity feature captures a different aspect, such as

the topic of the document, the entities that participate and the entities that also interact. These similarity features are provided to a classifier that decides if the document is novel. The supervised approach employed, referred to as REL-EFSD, according to the original publication [1], achieved significant improvement in two datasets, with respect to the state-of-the-art.

If a document is detected as a first story, a new event cluster is created. On the other hand, if a document is detected as a non-first story, it is assigned to the nearest neighbor cluster. A document is assigned to the cluster with the nearest centroid using the cosine similarity of the TF-IDF weighted vectors as a distance metric. We also used the aggregated Word2Vec vector. However, the usage of the average Word2Vec document vector did not improve the results. Thus, we used the TF-IDF vectors due to their good performance and simplicity.

4.5. Trends Detection

All of the pre processed news articles are delivered to the Trends Detection module to get the trending keywords and named entities (and also n-grams), that will later be used from the Twitter Fetcher to crawl all related tweets. For each time window, we count and store the number of appearance of each named entity and keyword in the news articles. The number of entities, keywords and counts is limited and as a result we store those in hash tables. In case where we need to extend the vocabulary of entities and keywords, we could use a randomized approach as for example count-min sketch.

We then use these counts to calculate the z-scores for all elements. The z-scores are defined as in Equation (1). $Counts_w(Entity)$ refers to the number of times an entity is referred to in the time window w. $Mean(Counts(Entity))$ and $Std(Counts(Entity))$ refers to the mean and the standard deviation of the entity's appearances in the last windows.

The most trending entities and keywords (those that appear more frequently according to their average activity) are selected and provided to the user. As already mentioned, the most popular entities are used by the Twitter fetcher in order to crawl Twitter.

$$Z_{score}(Entity) = \frac{Counts_w(Entity) - Mean(Counts(Entity))}{Std(Counts(Entity))} \quad (1)$$

4.6. Question Answering

As mentioned in the functionality of our system, the users can query an article using the English language and the Question Answering module will calculate the answer to each user generated query. News Monitor utilizes the recent language model BERT for the task. More specifically, this model is trained to receive, as an input, a query in the natural language, along with a document, and provides, as an output, the document span that corresponds to the answer. More specifically, the BERT "bert-large-uncased" pre-trained model[7] in the SQuAD dataset is used.

According to the original BERT [16] publication, for the task of question answering, the model "bert-large" surpassed all of the competitor methods in the datasets SQuAD 1.1 and SQuAD 2.0 and, specifically, for the SQuAD 1.1 dataset, the performance was similar to human annotators.

4.7. Abstractive Summarization

News Monitor also contains a module that performs abstractive text summarization. While this summarizing version provides only a small chunk of text describing the main event, it is very useful for users with very limited time. News Monitor utilizes the state-of-the-art model text-to-text transfer transformer [8] (T5) and, using the library Hugging Face[8], the transformer is able to provide a summary for every one of the news articles.

The text-to-text transfer transformer is an encoder–decoder deep architecture that is able to transform any natural language processing tasks to a sequence-to-sequence task. That is, the input is a document and the output is, again, a document. For a classification

task, the T5 model learns to generate the label "Text" given as an input to the document text. For a semantic similarity task, the T5 model is given two sentences as an input and is able to provide a text with the semantic similarity of the sentences. Finally, for the abstractive summarization task, the T5 is trained by providing documents and also short versions of these documents as an input. The T5 model actually learns to re-write the input document; this is why it is referred to as abstractive summarization. In contrast to extractive summarization, in abstractive summarization, the summary is not restricted to containing only chunks from the document.

According to the original publication, T5 achieved a state-of-the-art performance in the CNN/DailyMail[9] dataset on all of the evaluation metrics, including ROUGE-1, ROUGE-2 and ROUGE-L.

4.8. Fake News Detection

The Fake News Detection module retrieves the tweets and classifies a tweet as a rumor or not. If a rumor is detected, it forms a cluster. The cluster remains alive for a predefined amount of time and the tweets of the cluster are classified for stance detection. In summary, in the first step, we identify the rumors, and, in the next step, we gather relevant tweets in order to identify the stance.

5. Evaluation

News Monitor is a framework that is built on top of a variety of state-of-the-art techniques. In this work, we evaluate the system against baseline solutions in terms of:

- Fake Tweets Detection: News Monitor collects tweets that are relevant to the news articles. Since the tweets contain unverified data, we filter them in terms of fake news. Specifically, we include the tasks of (a) rumor detection and (b) stance detection;
- Extractive Summarization: News Monitor, after extracting the major relations of a news article, selects the most appropriate of these. We evaluate this functionality as an instance of extractive summarization.

For each evaluation task, we later describe its experimental setup.

5.1. Datasets

For the purposes of evaluation, various public datasets were used, along with some datasets created from data collecting from the framework. The datasets can be summarized below:

The PHEME Dataset. The extended version of the PHEME dataset was proposed in the work of Kochkina et al. [49]. This dataset contains Twitter messages that are relevant to nine events. The dataset is annotated in three levels according to the authors. In the first level, each tweet is classified as a rumor or not. In the second level, the rumor tweets are categorized as true, false or unverified;

The Relations (REL) Dataset. In order to evaluate the task of extractive document summarization, we built an annotator interface on top of News Monitor. The user is able to view the graph of an article and is able to annotate the relations as useful and non-useful using the mouse buttons. The relations dataset contains 500 relations that belong to two classes: relations that are useful for summarizing the article and relations that are not useful for summarizing the article;

The FNC1 Dataset. For estimating the performance in terms of stance detection, we relied on the FNC1 dataset[10]. The dataset was released during the fake news detection challenge. The dataset contains (a) headlines and (b) bodies. The bodies correspond to some reaction to the headline. The task is to predict the correct stance given the head and the body. The available stances include (a) unrelated, (b) agree, (c) disagree and (d) discuss.

Evaluation Metrics

All of the tasks of News Monitor can be seen as classification tasks. Thus, standard metrics used in information retrieval can be used. The evaluation metrics we use include:

accuracy, precision, recall and F-measure. Since accuracy is sensitive to imbalanced classes, we decided to report the precision and recall of the rumor class, as well as their harmonic mean F-measure. The definitions for the evaluation metrics are described below. In all of the experiments, the micro (weighted) precision, recall and F-measure are reported.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

5.2. Methods

We experimented with both traditional machine learning techniques, as well as with recent deep learning state-of-the-art techniques, such as BERT, which is utilized by News Monitor. The methods we included in the evaluation are described below:

- Random: A random guess is provided, taking into account the class balance;
- Naive Bayes: A multinomial naive Bayes classifier with the default parameters;
- Logistic Regression: A logistic regression classifier is fitted into the TF-IDF-weighted vectors. The number of iterations is set to 100;
- Random Forest: A random forest with 100 estimators is fitted into the TF-IDF-weighted vectors. The Gini split criterion is selected;
- BERT: We used BERT embeddings with a classification head in order to classify a sequence of tokens as rumor and non-rumor. We used the Hugging Face transformers 4.0 implementation. The same model is also used for selecting a relation as useful or not.

All of the linear models and the random forest implementation were obtained from the library Scikit-Learn 0.24.2. The default hyperparameters were used. The BERT implementation was provided from Hugging Face transformers 4.0. For the naive Bayes, logistic regression and random forest models, the documents were converted to TF-IDF vectors. For the BERT model, the word-piece tokenizer is used. The parameters of the models are described in Table 2.

Table 2. The parameters for the models used.

Method	Major Parameters	Library
Random	method = stratified random predictions	Sk-Learn 0.24.2
Logistic Regression	penalty = L2 norm, tolerance = 1×10^{-4}	Sk-Learn 0.24.2
Random Forest	num_trees = 100, split criterion = gini	Sk-Learn 0.24.2
BERT	hidden_size = 768, hidden_layers = 12, attention_heads = 12, model = "bert-base-uncased"	Hugging Face Transformers 4.0

5.3. Rumor Detection Results

The first evaluation task in terms of fake news detection is that of rumor detection. That is, from a list of documents, the final goal is to identify which of these are rumors and which are not.

Experimental Setup: We use the PHEME dataset and the first level of annotation that contains tweets belonging to the categories (a) rumor and (b) non-rumor. The dataset is split randomly into a training set (80%) and a testing set (20%).

According to the results, the random baseline that exploits the class balance does not perform well, with an accuracy of 54%. As expected, the first supervised system that uses the naive Bayes classifiers scores an accuracy of 75%. The logistic regression baseline scores an accuracy of 83% and the random forest classifier scores an accuracy of 85%. The most complex model we tested, the BERT model, scored an accuracy of 87%, with the highest F-measure score of 82%. The detailed results are described in Table 3 and are illustrated in Figure 6.

Table 3. Results for the rumor identification task.

Method	Accuracy %	Precision %	Recall %	F1 %
Random	54	38	39	38
Naive Bayes	75	82	64	71
Logistic Regression	83	67	86	76
Random Forest	85	73	87	79
BERT	87	83	82	82

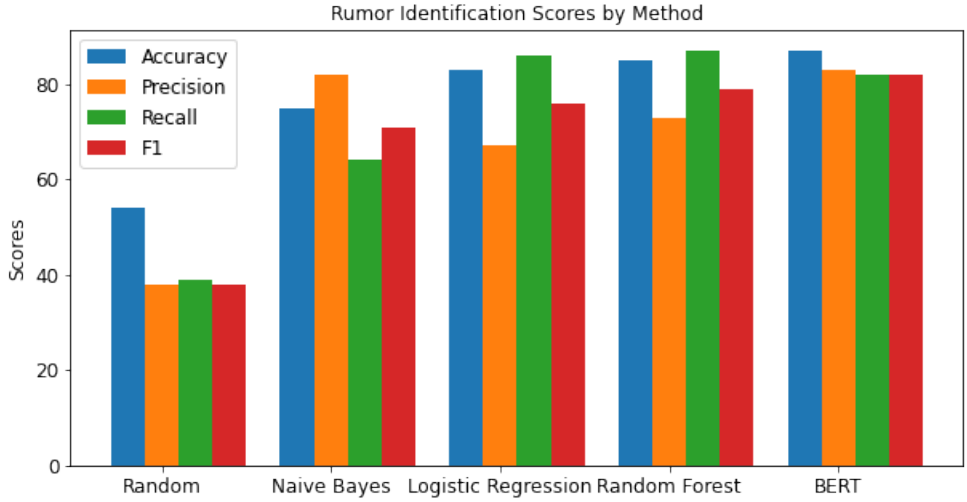


Figure 6. Results for the rumor detection tasks for different baselines.

5.4. Stance Detection Results

The second evaluation task that is relevant to the fake news detection problem is the problem of stance detection. In this problem, a pair is given as an input that contains a pair that consists of two parts. The first part contains an original message (e.g., a news headline). The second part contains a user response to the first part (e.g., a user tweet). The final goal is to detect if the response (second part) agrees with the first part.

Experimental Setup: For this task, the dataset FNC-1 is used, which provides such pairs and, for each pair, an annotation. For this experiment, the classical machine learning models receive two different TF-IDF vectors as an input, one for the first part and one for the second part. The linear models are expected to handle the task as sentiment classification, where the two input vectors are treated as a single vector. The unrelated class cannot be effectively treated by these models since a comparison between the two vectors (e.g., cosine similarity) is necessary. The random forest baseline, however, has the ability to compare the vectors. Finally, the BERT model receives the head and the body as two separate sentences. The special token "[SEP" is used by the encoder. The model is then fine-tuned to optimize the classification task. Since labels only used for the training set are available, we split the training set into two sets: (a) training (80%) and (b) testing (20%).

The best results (92% accuracy) are obtained by the random forest baseline, while BERT scored the second best accuracy score (82%). The results for all of the baselines are illustrated in Figure 7 and Table 4. Notably, even the random model has a great performance.

This is explained by the fact that the classes are highly unbalanced, with the vast majority of the stances "Unrelated". Since the micro precision, recall and F-measure are used, the metrics are dominated by the majority class. For instance, the random model scores an F-measure of 0.0 in all of the other classes, and thus a macro F-measure score of ≈0.25. However, the random forest model has a macro F-measure score of 0.75, whereas naive Bayes and logistic regression have macro F-measure scores of 0.33 and 0.56, respectively.

Table 4. Results for the stance identification task.

Method	Accuracy %	Precision %	Recall %	F1 %
Random	58	73	73	73
Naive Bayes	77	78	97	86
Logistic Regression	80	81	82	80
Random Forest	92	92	93	92
BERT	82	93	82	86

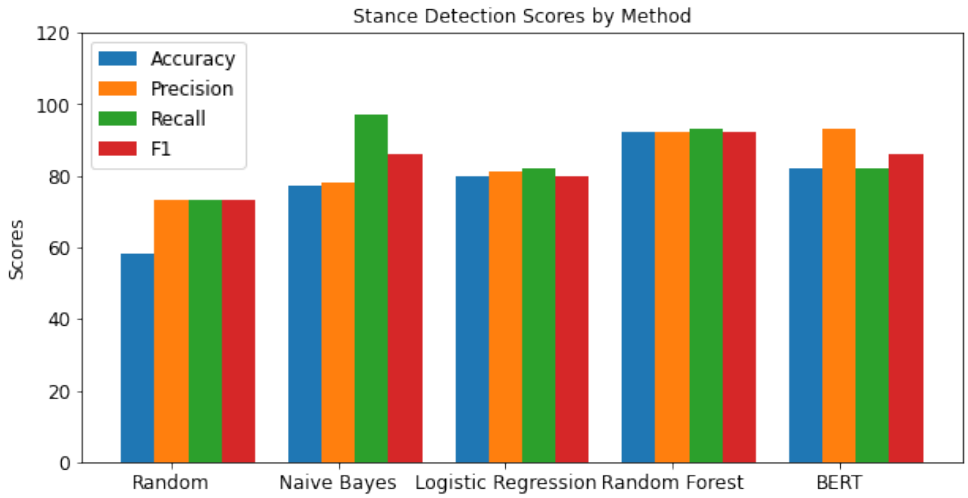


Figure 7. Results for the stance detection task for different baselines.

5.5. Extractive Summarization Results

The last evaluation task is for the extractive summarization feature. As a reminder, in the first step, an open information extraction system extracts some relations, using syntactic information and part of speech information, which correspond to the summary. In the second step, we classify these relations as useful or not.

Experimental Setup: For the extractive summarization task, we used the relations dataset (REL). This dataset contains relations extracted from News Monitor and are annotated as (a) useful and (b) non-useful. The dataset is split again into a training set (80%) and a testing set (20%).

Since the second task is a fairly simple text classification instance, even simple models perform extremely well. For instance, logistic regression scores an F1 of 83%. On the other hand, due to the limited data for fine-tuning the model, BERT is not an appropriate choice, with an F1 of 62%. The detailed results for the task are illustrated in Figure 8 and Table 5. According to these result, a simple and extremely computational inexpensive model, such as naive Bayes, is enough.

Table 5. Results for the Extractive Summarization task.

Method	Accuracy %	Precision %	Recall %	F1 %
Random	50	66	57	61
Naive Bayes	72	78	83	81
Logistic Regression	73	74	94	83
Random Forest	76	76	97	85
BERT	60	94	46	62

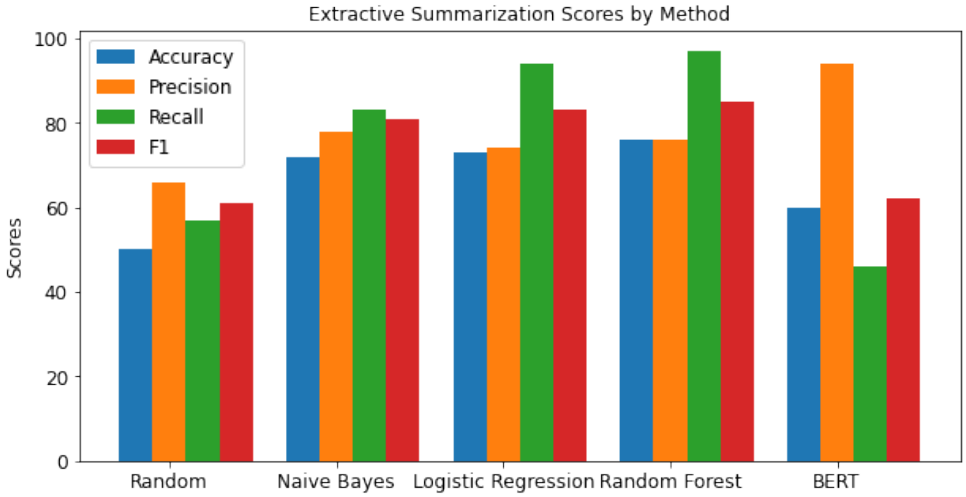


Figure 8. Results for the extractive summarization tasks for different baselines.

5.6. System Scalability

The News Monitor framework has constant space and time requirements through the usage of efficient randomized data structures. More specifically, the most expensive component in terms of complexity is the first story detection. This is because this component needs to calculate the nearest neighbor for every new document in the stream. However, as already mentioned, the component uses a bounded LSH index in order to speed up the queries, and the implementation is vectorized in order to benefit from implicit parallelization. On average, the system requires 10 ms to identify a first story. In Figure 9, the processing time (y-axis), with respect to the time (x-axis), is illustrated. As shown in the figure, the processing time remains stable over time, verifying the constant space and time requirements statement. Some random spikes in the figure are explained as a result of the sudden load in the machine.

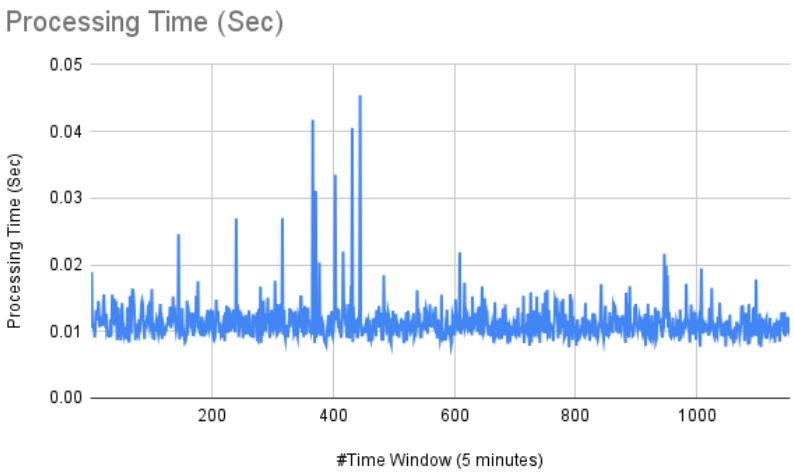


Figure 9. Processing time per document.

6. Demonstration

The interface of our proposed framework, News Monitor, is shown in Figure 3. Our system is built to help the readers analyze and explore the news stories and the different views and opinions in each topic. We collect news articles from more than 500 RSS feeds and we also collect all messages that are being posted in a social network based on the entity and keyword popularity from our Trend Detection module. Our framework detects the first stories that describe the latest news, and the users can read the first stories, or they can decide if they want to expand for more stories in the same main event. This gives the opportunity to the users to look at the same event from different angles and also know about the multiple views and various opinions on the same topic.

News Monitor uses the collected corpus of news articles to construct a knowledge base and a knowledge graph per article that forms a summary in a graph structure (an example of a summary appears in Figure 5). In addition, our framework allows users to ask questions in the English language relevant to each article and answers them in the form of a text chunk by using the state-of-the-art BERT [16] model. This feature is one of the novelties of our work, as it is something that does not exist in Yahoo News or Event Registry which are the most related systems to ours.

As mentioned above, News Monitor creates a knowledge graph and a knowledge base from the tuples that are being extracted from the collection of news articles. This gives one more the ability to users that are also able to search using a structured search option that is searching the knowledge base for answers. They can also read more in the article that the answer was found. In addition, they can search with more sophisticated queries using the advanced search option. This functionality is another novelty of our work, more specifically the ability to search the sub-events of an article directly.

Finally, our system allows the users to explore the people's views on each topic based on Twitter messages. The system based on the users' queries will provide a collection of messages from Twitter (tweets) that are the most relevant. Our system calculates the sentiment in each tweet using the BERT model. This feature is another novel feature that is added in News Monitor and does not exist in the related systems Event Registry and Yahoo News.

7. Discussion

In this work, we demonstrate News Monitor, a scalable framework for exploring news in real-time. News Monitor collects news from a vast amount of RSS news sources and automatically extracts the main content from the articles, along with other metadata, such as the images' URLs and the raw HTML content. For every article, it performs natural language processing in order to extract useful pieces of information. In addition, it performs open information extraction for every incoming article using the web scale algorithm ReVerb. Its is able to perform first story detection in real-time and uses online clustering in order to group together articles about the same topic. Finally, it collects tweets relevant to the trending entities in the news articles and performs rumor detection in real time.

Focusing on the article exploration aspect of News Monitor, it allows for the user to view a summary of the article in the form of a knowledge graph in order to visually explore the relationships present in the article. Furthermore, the framework provides an abstractive summary to the user using the transformers library and the T5 model, and allows for the user to query the article in their natural language using the BERT model. In addition, the user is able to explore the user knowledge base in order to find relevant articles, and, finally, the user is able to explore the trending entities and the trending keywords.

The architecture of News Monitor follows the distributed micro-services paradigm and is based on Apache Kafka technology. Each of the components are able to run independently on a different machine and each component periodically monitors its health, including its processing time and memory usage. Currently, News Monitor runs on two computer nodes, each with 16 threads and 32 GB of RAM.

From its instantiation, News Monitor has downloaded more than 500 K articles and has formed more than 300 K news clusters. In addition, News Monitor has collected more than four million tweets relevant to the trending entities that are present in the news articles. The News Monitor knowledge base consists of more than six million triples, which the user is able to query and filter based on the type of the entities.

Currently the knowledge base is stored in a MongoDB database and does not support complex graph queries. As a future work, we plan to utilize graph databases technology, such as Neo4J, in order to allow for the user to perform complex queries in the knowledge base using the Neo4J scripting language. In addition, News Monitor is able to answer natural language queries in a specific language. However, this is not possible for the knowledge base. As a future work, we plan to exploit recent transformer-embedding models, such as BERT, combined with semantic parsers, such as PARALEX [40] and Sempre [15], in order to also perform question answering in the knowledge base. In addition, News Monitor does not support a functionality to export data for researchers. A future work is to implement some functionality that is relevant to exporting aggregated data, such as n-grams, in order to allow other researchers to also use the data. Finally, News Monitor uses models that are periodically trained. However, feedback from users can be utilized for uncertain predictions in an active learning process.

8. Conclusions

We described News Monitor, a framework for analyzing news articles in real time. The systems uses a mixture of text mining and natural language processing techniques in order to allow for the end user to quickly explore the news articles. The end vision is that the system will automatically construct and maintain a knowledge graph of the various events and sub-events mentioned in the news. The system mines the news streams in real-time and is designed as a distributed architecture based on micro-services that allows for quick and also flexible scaling. It runs on minimal hardware and, when possible, exploits randomized solutions, such as locality-sensitive hashing, in order to reduce the computation requirements. Finally, according to the evaluation results described, the fake news detection techniques used by News Monitor are able to obtain a F-measure of 82% in rumor detection tasks, and an accuracy of 92% in stance detection tasks.

Author Contributions: Conceptualization, N.P., A.S. and D.G.; methodology, N.P., A.S. and D.G.; software, N.P. and A.S.; validation, N.P., A.S. and D.G.; formal analysis, N.P., A.S. and D.G.; investigation, N.P., A.S. and D.G.; resources, N.P., A.S. and D.G.; data curation, N.P., A.S. and D.G.; writing—original draft preparation, N.P., A.S. and D.G.; writing—review and editing, N.P., A.S. and D.G.; visualization, N.P., A.S. and D.G.; supervision, D.G.; project administration, D.G.; funding acquisition, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: The present work was co-funded by the European Union and Greek national funds through the Operational Program "Human Resources Development, Education and Lifelong Learning" (NSRF 2014–2020), under the call "Supporting Researchers with an Emphasis on Young Researchers—Cycle B" (MIS:5048149).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this work, we have experiment with two publicly available datasets that are also described in the paper. The first dataset is called the PHEME Dataset, the extended version of the PHEME dataset was first used in a paper by Kochkina et al. [49] and it is released publicly in the following link: https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078 (accessed on 12 December 2021). This is an annotated dataset that contains Twitter messages from nine events and it is used to evaluate for rumor detection. The second dataset that we have utilized is the FNC1 Dataset. This is a publicly available dataset, that has been released in the following link: https://github.com/FakeNewsChallenge/fnc-1 (accessed on 12 December 2021). This is an annotated dataset that is used to evaluate for stance detection.

Acknowledgments: The authors would like to thank the European Union and Greek National Funds for funding this project. The authors would like to thank the reviewers for providing constructive feedback to improve the quality of this work. Part of this work is reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, European Conference on Information Retrieval, Saravanou A., Panagiotou N., Gunopulos D. (2021) News Monitor: A Framework for Querying News in Real Time. In: Hiemstra D., Moens MF., Mothe J., Perego R., Potthast M., Sebastiani F. (eds) Advances in Information Retrieval. ECIR 2021. Lecture Notes in Computer Science, vol. 12657. Springer, Cham. doi:10.1007/978-3-030-72240-1_62, © Springer Nature Switzerland AG 2021.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Notes

1 System demonstration available in: http://195.134.67.89/news_monitor/ (accessed on 12 December 2021).
2 https://blog.google/products/search/how-google-delivers-reliable-information-search/ (accessed on 12 December 2021).
3 https://github.com/buriy/python-readability (accessed on 12 December 2021) package. This package is able to extract the main content by heuristically searching for the largest HTML " " tag.
4 https://spacy.io/ (accessed on 12 December 2021).
5 https://github.com/knowitall/reverb (accessed on 12 December 2021).
6 https://www.elastic.co/ (accessed on 12 December 2021).
7 https://huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad (accessed on 12 December 2021).
8 https://huggingface.co/ (accessed on 12 December 2021).
9 https://github.com/abisee/cnn-dailymail (accessed on 12 December 2021).
10 https://github.com/FakeNewsChallenge/fnc-1 (accessed on 12 December 2021).

References

1. Panagiotou, N.E.; Akkaya, C.; Tsioutsiouliklis, K.; Kalogeraki, V.; Gunopulos, D. A General Framework for First Story Detection Utilizing Entities and their Relations. *IEEE Trans. Knowl. Data Eng.* **2020**, *33*, 3482–3493. [CrossRef]
2. Panagiotou, N.; Katakis, I.; Gunopulos, D. Detecting events in online social networks: Definitions, trends and challenges. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 42–84.
3. Saravanou, A.; Katakis, I.; Valkanas, G.; Gunopulos, D. Detection and Delineation of Events and Sub-Events in Social Networks. In IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018; pp. 1348–1351. [CrossRef]
4. Sethi, P.; Sonawane, S.; Khanwalker, S.; Keskar, R. Automatic text summarization of news articles. In Proceedings of the 2017 International Conference on Big Data, IoT and Data Science (BID), Pune, India, 20–22 December 2017; pp. 23–29.
5. Sun, C.; Huang, L.; Qiu, X. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In Proceedings of NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 380–385.
6. Mathioudakis, M.; Koudas, N. Twittermonitor: Trend detection over the twitter stream. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10), 6–10 June 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 1155–1158.
7. Helmstetter, S.; Paulheim, H. Weakly supervised learning for fake news detection on Twitter. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018; pp. 274–277.
8. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
9. Saravanou, A.; Panagiotou, N.; Gunopulos, D. News Monitor: A Framework for Querying News in Real Time. In Proceedings of 43rd European Conference on IR Research, ECIR 2021, Virtual Event, 28 March–1 April 2021; pp. 543–548.
10. Allan, J.; Lavrenko, V.; Malin, D.; Swan, R. Detections, Bounds, and Timelines: Umass and tdt-3. Available online: http://ciir.cs.umass.edu/pubfiles/ir-201.pdf (accessed on 20 December 2021).
11. Petrović, S.; Osborne, M.; Lavrenko, V. Streaming first story detection with application to twitter. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10), Los Angeles, CA, USA, 2–4 June 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 181–189.

12. Petrović, S.; Osborne, M.; Lavrenko, V. Using paraphrases for improving first story detection in news and Twitter. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT '12), Montreal, QC, Canada, 3–8 June 2012; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 338–346.
13. Wurzer, D.; Lavrenko, V.; Osborne, M. Twitter-scale new event detection via k-term hashing. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 2584–2589. [CrossRef]
14. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale Simple Question Answering with Memory Networks. *arXiv* **2015**, arXiv:1506.02075.
15. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1533–1544.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
17. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.
18. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
19. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; pp. 1877–1901.
20. Mausam; Schmitz, M.; Bart, R.; Soderland, S.; Etzioni, O. Open language learning for information extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 523–534.
21. Fader, A.; Soderland, S.; Etzioni, O. Identifying relations for open information extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11), Edinburgh, UK, 27–31 July 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 1535–1545.
22. Etzioni, O.; Fader, A.; Christensen, J.; Soderland, S.; Mausam, M. Open information extraction: The second generation. In Proceedings of the 22nd international joint conference on Artificial Intelligence—Volume One (IJCAI'11), Barcelona, Spain, 16–22 July 2011; pp. 3–10.
23. Banko, M.; Cafarella, M.J.; Soderland, S.; Broadhead, M.; Etzioni, O. Open information extraction for the web. In Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07), Hyderabad, India, 6–12 January 2007; pp. 2670–2676.
24. Pal, H. Demonyms and compound relational nouns in nominal open IE. In Proceedings of the 5th Workshop on Automated Knowledge Base Construction, San Diego, CA, USA, 17 June 2016; pp. 35–39.
25. Saha, S. Open information extraction from conjunctive sentences. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2288–2299.
26. Ma, T.; Pan, Q.; Rong, H.; Qian, Y.; Tian, Y.; Al-Nabhan, N. T-BERTSum: Topic-Aware Text Summarization Based on BERT. *IEEE Trans. Comput. Soc. Syst.* **2021**, 1–12,
27. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 11328–11339.
28. Shu, K.; Wang, S.; Liu, H. Beyond news contents: The role of social context for fake news detection. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), Melbourne, Australia, 11–15 February 2019; pp. 312–320. [CrossRef]
29. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explor. Newsl.* **2017**, *19*, 22–36. [CrossRef]
30. Lukasik, M.; Srijith, P.; Vu, D.; Bontcheva, K.; Zubiaga, A.; Cohn, T. Hawkes processes for continuous time sequence classification: An application to rumour stance classification in twitter. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 393–398.
31. Kumaran, G.; Allan, J. Using names and topics for new event detection. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 121–128.
32. Saravanou, A.; Valkanas, G.; Gunopulos, D.; Andrienko, G. Twitter floods when it rains: A case study of the UK floods in early 2014. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion), Florence, Italy, 18–22 May 2015; pp.1233–1238. [CrossRef]
33. Kaleel, S.B.; Abhari, A. Cluster-discovery of Twitter messages for event detection and trending. *J. Comput. Sci.* **2015**, *6*, 47–57. [CrossRef]
34. Nguyen Mau, T.; Inoguchi, Y. Locality-Sensitive Hashing for Information Retrieval System on Multiple GPGPU Devices. *Appl. Sci.* **2020**, *10*, 2539. [CrossRef]

35. Corizzo, R.; Pio, G.; Ceci, M.; Malerba, D. DENCAST: Distributed density-based clustering for multi-target regression. *J. Big Data* **2019**, *6*, 43. [CrossRef]
36. Karkali, M.; Rousseau, F.; Ntoulas, A.; Vazirgiannis, M. Efficient online novelty detection in news streams. In Proceedings of the Web Information Systems Engineering (WISE 2013), Nanjing, China, 13–15 October 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 57–71.
37. Moran, S.; McCreadie, R.; Macdonald, C.; Ounis, I. Enhancing first story detection using word embeddings. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16), Pisa, Italy, 17–21 July 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 821–824. [CrossRef]
38. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'13), Stateline, NV, USA, 5–10 December 2013; pp. 3111–3119.
39. Saravanou, A.; Katakis, I.; Valkanas, G.; Kalogeraki, V.; Gunopulos, D. Revealing the hidden links in content networks: An application to event discovery. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17), Singapore, 6–10 November 2017; pp. 2283–2286. [CrossRef]
40. Berant, J.; Liang, P. Semantic parsing via paraphrasing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, USA, 22–27 June 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1415–1425. [CrossRef]
41. Bordes, A.; Chopra, S.; Weston, J. Question answering with subgraph embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), October 2014, Doha, Qatar; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 615–620. [CrossRef]
42. Christensen, J.; Soderland, S.; Etzioni, O. An analysis of open information extraction based on semantic role labeling. In Proceedings of the Sixth International Conference on Knowledge Capture, Banff, AB, Canada, 25–29 June 2011; pp. 113–120.
43. Cui, L.; Wei, F.; Zhou, M. Neural open information extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 407–413. [CrossRef]
44. Narayan, S.; Cohen, S.B.; Lapata, M. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, USA, 1–6 June 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 1747–1759. [CrossRef]
45. Zhang, X.; Lapata, M.; Wei, F.; Zhou, M. Neural Latent Extractive Document Summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 779–784. [CrossRef]
46. Liu, Y.; Lapata, M. Text Summarization with Pretrained Encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3730–3740.
47. Srikanth, A.; Umasankar, A.S.; Thanu, S.; Nirmala, S.J. Extractive Text Summarization using Dynamic Clustering and Co-Reference on BERT. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Bihar, India, 14–16 October 2020; pp. 1–5. [CrossRef]
48. Zubiaga, A.; Liakata, M.; Procter, R.; Wong Sak Hoi, G.; Tolmie, P. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLoS ONE* **2016**, *11*, e0150989. [CrossRef] [PubMed]
49. Kochkina, E.; Liakata, M.; Zubiaga, A. All-in-one: Multi-task Learning for Rumour Verification. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–25 August 2018; pp. 3402–3413.
50. Kochkina, E.; Liakata, M.; Augenstein, I. Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm. *arXiv* **2017**, arXiv:1704.07221.
51. Wang, W.Y. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv* **2017**, arXiv:1705.00648.
52. Reis, J.C.; Correia, A.; Murai, F.; Veloso, A.; Benevenuto, F. Supervised learning for fake news detection. *IEEE Intell. Syst.* **2019**, *34*, 76–81. [CrossRef]
53. Watanabe, K.; Ochi, M.; Okabe, M.; Onai, R. Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs. In Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11), Glasgow, UK, 24–28 October 2011; pp. 2541–2544. [CrossRef]
54. Sankaranarayanan, J.; Samet, H.; Teitler, B.E.; Lieberman, M.D.; Sperling, J. Twitterstand: News in tweets. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09), Seattle, WA, USA, 4–6 November 2009; pp. 42–51. [CrossRef]
55. 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, Seoul, Korea, 7–11 April 2014; pp. 107–110.