Chapter

Combined Deep Learning and Traditional NLP Approaches for Fire Burst Detection Based on Twitter Posts

Konstantinos-George Thanos, Andrianna Polydouri, Antonios Danelakis, Dimitris Kyriazanos and Stelios C.A. Thomopoulos

Abstract

The current chapter introduces a procedure that aims at determining regions that are on fire, based on Twitter posts, as soon as possible. The proposed scheme utilizes a deep learning approach for analyzing the text of Twitter posts announcing fire bursts. Deep learning is becoming very popular within different text applications involving text generalization, text summarization, and extracting text information. A deep learning network is to be trained so as to distinguish valid Twitter fire-announcing posts from junk posts. Next, the posts labeled as valid by the network have undergone traditional NLP-based information extraction where the initial unstructured text is converted into a structured one, from which potential location and timestamp of the incident for further exploitation are derived. Analytic processing is then implemented in order to output aggregated reports which are used to finally detect potential geographical areas that are probably threatened by fire. So far, the part that has been implemented is the traditional NLP-based and has already derived promising results under real-world conditions' testing. The deep learning enrichment is to be implemented and expected to build upon the performance of the existing architecture and further improve it.

Keywords: deep learning, NLP procedure, fire burst detection, twitter posts, valid posts

1. Introduction

Due to their cost and easy access, social media and Twitter, among them, are widely used as sources of news and means of information spreading. Among others, fire bursts are such breaking news that can be initially made known through Twitter posts.

Mega fires often result in significant environmental destructions, major damages on infrastructures, and economic loss. Most importantly, they put at stake the lives, not only of the civilians but also of the forest fire personnel. Thus, technologies that

facilitate early fire detection are important for reducing fires and their negative effects.

Our approach proposes the combination of a deep learning architecture along with a more traditional natural language processing (NLP) one. The deep learning component of the system is responsible for filtering out the fake from the valid fire-related posts, so that only posts containing true fire-related information are retained. For this part of the system, we refer to current state-of-the-art systems for detecting fake news and adopt the one that suits the needs of our problem best. Once the fake posts are filtered out, each valid post is afterward fed into the NLP-based subsystem. By converting the unstructured, raw text into a structured one, the NLP-based subsystem is able to extract information, such as the geographical area of the fire reported in the post. In order to draw final conclusions about the possible fire sources, aggregation statistics over the posts containing similar fire-related information are computed, and probability values for each potential fire source are given as output.

The rest of the chapter is organized as follows: Section 2 describes and analyzes the deep learning-based architecture to be utilized for detecting valid Twitter posts regarding fire bursts. Section 3 illustrates the typical NLP-based architecture for extracting meaningful information from the unstructured text of a valid Twitter post. Section 4 presents the overall scheme and its final output. Finally, before the conclusions, Section 5 highlights the results of the up-to-date validated part of the overall proposed scheme.

2. Deep learning-based architecture for false Twitter post detection

2.1 Introduction

Social media are low-cost and easy-to-access means of information sharing and, thus, nowadays are widely used as source of news and information. However, getting informed from social media is not always safe, as posts expressing fake news (i.e., news containing false information) are exponentially widespread, simultaneous to the boosting development of online social networks. In fact, fake news tends to outperform the valid ones in the near future [1].

In case of fire burst news, deciding whether a Twitter post is fake or not can be proven of crucial twofold importance. On the one hand, the required time and money for purposeless activation of the firefighting mechanisms are saved. On the other hand, timely confrontation of mega fires is facilitated. This will, in turn, make it less likely for human lives, the environment, and infrastructures to be jeopardized.

Thus, before extracting the crucial fire burst information at a later NLP-based stage, a preprocessing step, deciding whether a Twitter post that declares a fire burst is fake or not, is necessary. To this end, a deep learning-based architecture is to be implemented. The purpose of this architecture is to filter out the posts that will be characterized as "fake" and provide the sequential NLP procedure only with the "valid" posts.

2.2 Candidate deep learning architectures

In this subsection, the candidate state-of-the-art deep learning architectures for the detection of fake posts are described. The purpose of the subsection is to illustrate the most recent and modern approaches that have appeared from 2017 onward and have been examined. The input data, used by all architectures, is the

text provided by social media posts, especially focused on Twitter. The output is the decision whether the input text corresponds to a "valid" or "fake" post.

The 3HAN architecture [2] utilizes a three-level hierarchical attention network. Each of the three levels corresponds to words, sentences, and headline analysis. The three-edge analysis results in the construction of a news vector which represents the input post. The latter vector is used for classifying the reliability of the post.

The architecture presented in [3], namely, ConvNet, uses a convolutional layer to capture the dependency between the text and its metadata. For the case of the metadata, a standard max pooling and a bidirectional Long Short-Term Memory (LSTM) auto-encoder layer follow. For the case of the text, only a max-pooling layer is implemented. Finally, the max-pooled text representations are concatenated with the metadata representation from the bidirectional LSTM. The merged concatenations are fed to a fully connected layer with a softmax activation function. This generates the final prediction.

The work in [4] presents the FakeNewsTracker architecture. This is a deep learning architecture which is divided into two sub-schemes. The first sub-scheme uses an LSTM deep network [5] in order for the system to be trained on the post representation context. The second sub-scheme utilizes a recursive neural network (RNN) in order to be trained on the context of social engagements. The output features of the aforementioned sub-schemes are fused together to perform a binary classification procedure which labels the input news as "fake" or "valid."

The DeClarE architecture [6] is based on bidirectional LSTMs in order to result in a credibility score related to the input post. The scheme also considers post source and claims information, which is processed within the bidirectional LSTM dense layers. The concatenated output is also processed by two dense layers and a softmax layer before the prediction of the credibility score.

The work in [7] introduces a hybrid architecture approach which combines an LSTM and a convolutional neural network (CNN) model. Throughout this chapter, the aforementioned architecture will be called Hybrid LSTM-CNN. The LSTM was adopted for the sequence classification of the data. The 1D CNN was added immediately after the word embedding layer of the LSTM model. A max-pooling layer is also recruited to reduce dimensionality of the input layer, thus avoiding training over-fitting of the training data. This also helps in reducing the resources for the training of the model.

The FakeDetector architecture [8] relates post creators to posts and subjects. It contains a Hybrid Feature Vector Unit (HFLU) which extracts the feature vector based on a specific input. The feature vector is fed to the gated diffusive unit (GDU) model for effective relationship modeling among news articles, creators, and subjects. Formally, the GDU model accepts multiple inputs from different sources simultaneously. The GDU applies softmax operation on the output vector before assigning a credibility label. For a more explicit sight on deep learning architectures, the reader is referred to [9].

2.3 Procedural requirements

Before deciding which architecture fits best in our specific case, the direct requirements of the overall procedure should be recorded.

To begin with, detecting fake posts in real time is an essential requirement of the process. Rapid decision whether a fire-bursting declaration post is fake or not leads to fast implementation of the NLP procedure (as described in Section 3). The latter, in turn, facilitates the timely detection of the geographical areas threatened by fire as soon as possible which helps toward the prevention of the majority of negative

effects caused by mega fires. Therefore, the proposed architecture of [3] is not suitable for our use case, as it is not implemented in a fully automated manner.

Fake news detection accuracy is very important. High detection accuracy guarantees that the great majority of the posts that fed to be processed in the sequential NLP phase (see Section 3) express sincere fire burst claims. Thus, the final resulting fire-threatened geographical areas are much more likely to be actually threatened. Furthermore, the aforementioned accuracy needs to have been achieved in publicly available datasets and benchmarks. This windows the performance of the architecture much more reliable than others, tested on proprietary datasets. To this end, the FakeNewsTracker architecture [4] is not suitable for our use case, as it is tested on a proprietary dataset.

Last but not least, the architecture needs to be domain invariant. In other words, it needs to be generally applicable to any domain, other than the one(s) used for conducting training and testing procedures. More precisely, the accuracy of a system, detecting fake post that deal with fire burst, should not be significantly altered in the case of post that deal with any other domain (politics, sports, etc.). This makes the system architecture much more flexible and adoptable. From the remaining architectures analyzed in this section, only DeClarE [6] and the Hybrid LSTM-CNN [7] claim to be domain invariant. DeClarE has been tested on PolitiFact dataset [10] achieving accuracy 67.32%, while the Hybrid LSTM-CNN has been tested on PHEME dataset [11, 12] achieving 82.00% accuracy. Both datasets are publicly available. PolitiFact is a respected fact-checking website releasing a list of sites manually investigated and labeled. It mostly contains posts of political content. PHEME is also another EU-funded project whose results include collecting and annotating rumor tweets which are associated with nine different breaking news contents. Therefore, PHEME is a richer dataset with a wider variety of themes that makes the Hybrid LSTM-CNN system architecture [7] it has been tested on more suitable for our use case.

The procedural requirements for the fake post detection scheme with respect to the architectures analyzed in Section 2 are summarized in **Table 1**.

2.4 Implementation architecture

Based on the aforementioned requirements, the baseline of the architecture selected to be implemented follows the Hybrid LSTM-CNN architecture [7]. The overall architecture is illustrated in **Figure 1**. The *input layer* consists of Twitter posts which are, in fact, unstructured raw texts. A *word embedding layer* follows, within which the input text is parsed and is divided into a series of words and, consequently, into a series of sentences.

Architectures	Procedural requirements			
	Real time	Accuracy	Public dataset	Domain invariance
3HAN [2]	✓	✓	✓	×
ConvNet [3]	×	×	✓	×
FakeNewsTracker [4]	×	✓	×	×
DeClarE [6]	i	✓	✓	✓
Hybrid LSTM-CNN [7]	✓	✓	✓	✓
FakeDetector [8]	✓	✓	✓	×

Table 1.
Procedural requirements for fake post detection part.

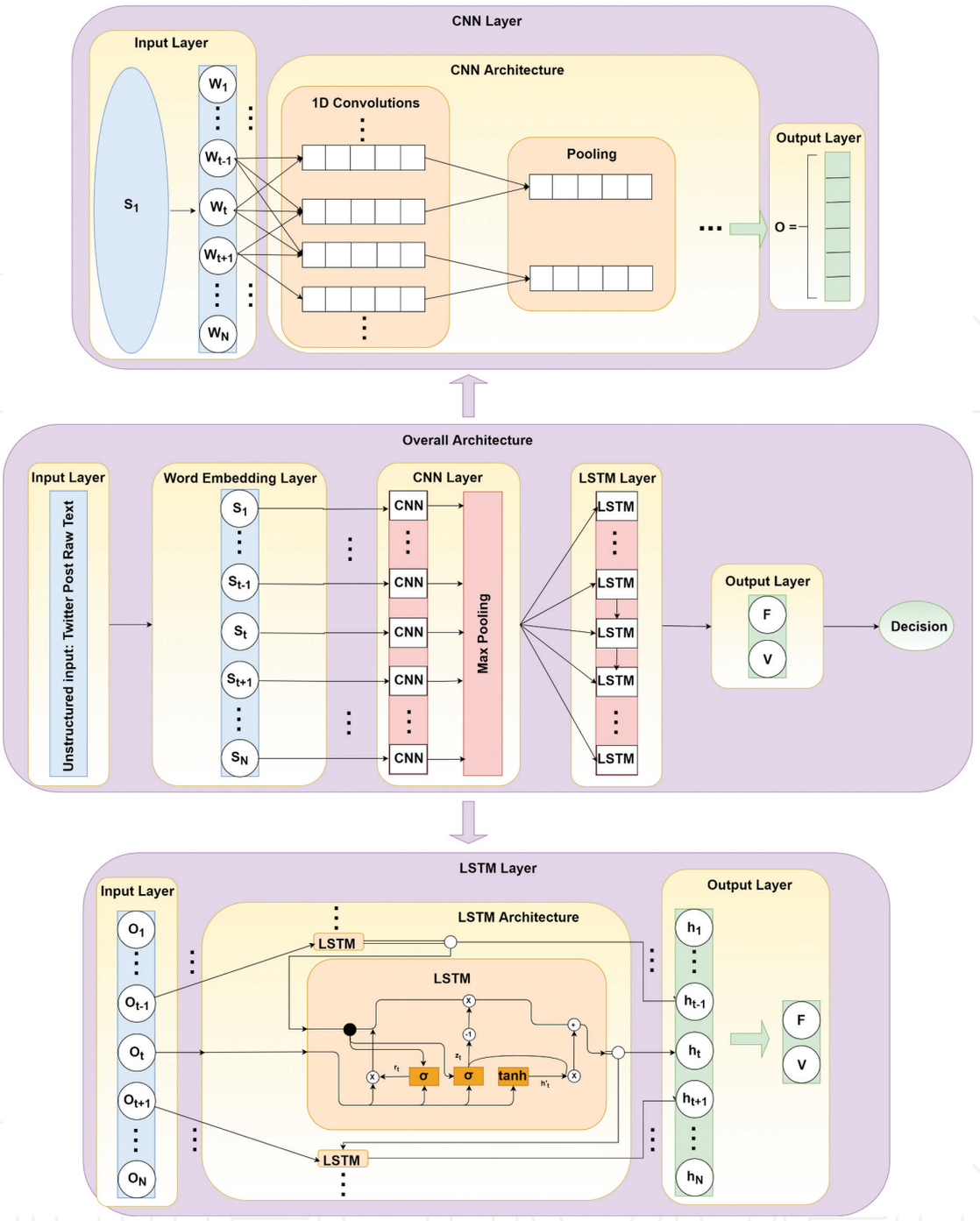

Figure 1.
Suggested fake post detection architecture.

Each sentence is then consumed by the *CNN layer* of the architecture which is made up of a set of 1D CNNs based on the work presented in [13]. The CNNs of this layer are structured as illustrated in the upper part of **Figure 1**. The 1D convolutions, taking place within the CNNs (as defined by Eq. (10) of the Appendix), operate on sliding windows of the words of the sentence. Before outputting the outcome of the layer, max pooling is performed to reduce dimensionality and avoid over-fitting of the training data. This also helps toward reducing computational complexity of the training process. The output of each CNN is a fixed length vector, acting as a digital signature of the corresponding sentence and describing the nature of the sentence. Thus, a set of such description vectors (descriptors) are fed forward for further process.

The *LSTM layer* follows, which is the core of the architecture. This layer consists of a set of LSTMs. It uses as input the sentence descriptors resulting from the CNN layer and outputs the final decision vector indicating whether the claim of the post is fake (F) or valid (V). Each LSTM of the layer is structured as presented in the lower part of **Figure 1**. LSTMs are chosen because they are proven to be robust for representing a series of data, such as the one we are dealing with here (i.e., series of words or sentences), as they are capable of capturing their internal temporal dependencies [9]. The LSTM layer is very interesting in terms of mathematics. For more information the reader is referred to Appendix.

3. NLP-based architecture for Twitter post information extraction

3.1 Introduction

This component consists of two sub-modules: (a) the fire incident report detection sub-module and (b) the fire incident report analytic sub-module. The first one is responsible for acquiring reports made by civilians on the Twitter platform and detects reports that refer to a potential fire incident. These reports are stored in a structured way. The fire incident report analytic sub-module is responsible for aggregating the detected fire incident reports, and based on the number of these reports and the location these reports refer to, it concludes to a probability that there was a significant amount of people that reported a fire incident at a specific location. The final output is the result along with a geographic area and a reliability score of each location and the coordinates of each location.

3.2. Fire incident detection

3.2.1 Introduction to information extraction

Natural language processing (NLP) is a field of computer science responsible for the study and analysis of raw text. The purpose of this field is to enhance human-computer communication by constructing systems that are capable of understanding raw text and incorporate interaction interfaces based on textual messages. Some of the main topics of NLP are learning syntactic and semantic rules and determining concept, topics, and sentiment from a document, automatic summarization, machine translation, natural language generation, information extraction, etc. [14].

Information extraction corresponds to the section of NLP which is responsible for the analysis of unstructured textual pieces and conversion to a structured form. For example, the conversion of the following unstructured text (raw text):

> "Yesterday, New York based Foo Inc. announced their acquisition of Bar Corp."

to the structured form:

```
MergerBetween('Foo Inc', 'Bar Corp', date ...)
```

The above-structured form corresponds to a relation of various entities that were embedded in the initial unstructured raw text. The benefit of this conversion is that structured relations can be manipulated by computer algorithms and finally

be exploited by computer algorithms. Apparently, for a given unstructured text, many structured forms correspond each one holding different knowledge and representing different relations. As a result, the algorithm designer has the responsibility of selecting the appropriate structured form.

The information extraction procedure consists of the following steps:

1. *Sentence segmentation*: the procedure of distinguishing different sentences.

2. *Tokenization*: the procedure of splitting each sentence to structural components (words and punctuations).

3. *Part of speech tagging*: the procedure of characterizing each token of each sentence to the corresponding part of speech.

4. *Entity recognition*: the procedure of characterizing tokens or set of tokens of each sentence based on previous knowledge. For example, characterize words referring to geographic locations as "city," "country," "mountain," etc.

5. *Relation recognition*: the procedure of detecting specific combination of tokens that corresponds to a specific meaning relation among them. For example, the following segmented tagged sentence **'George' (SUBJECT, NAME)** ↔ **'lives' (VERB, RELEVANT TO LOCATION)** ↔ **'in'** ↔ **'Athens' (OBJECT, LOCATION)** leads to the relation **lives('George', 'Athens')**.

The above procedures make use of text processing algorithms, knowledge representation, and information retrieval algorithms. In order to achieve text segmentation (sentence segmentation or tokenization), each text should be treated as an array of characters. Segmentation is based on the a priori knowledge of special characters that in most cases are used for splitting. For example, sentences usually end with a period mark "." or exclamation mark "!" or question mark "?" and begin with a capital letter. As a result a general rule for segmenting sentences would be to search for pairs: (special character ↔ capital letter) or (special character ↔ end of text).

Tagging procedures are usually based on knowledge databases and information retrieval algorithms. For this task, there is a need of having a lexical and syntactic and semantic database, which we call a corpora (of course different for each language!), which holds characterizations of several words to conceptual entities, and their relations in a structural way. Consequently, segmented texts (tokenized texts) are used as key vectors in order to retrieve from the corpora the corresponding characterization set. The most common approaches for this task are:

- *Sequential classification algorithms*: Hidden Markov Models (HMM) and Conditional Random Fields (CRF).

- *Classification algorithms*: Support Vector Machines (SVM) and Artificial Neural Networks (ANN).

Finally, relation extraction procedures demand from the algorithm designer to predefine either directly by specifying relation rules and use matching algorithms in order to detect word patterns corresponding to specific rule or indirectly by providing to the system several examples of annotated tagged sentences and then use classification algorithms in order to specify the corresponding relations.

3.2.2 Information extraction from Twitter

In this section, a real-case scenario of a system that was realized and evaluated for the purposes of real-time automatic fire detection as demanded by the EU-funded research project "AF3" is presented [15]. The suggested solution comprises a training phase where, via surveys, a variety of tweet samples for various predetermined occasions were collected. These samples were used in order to create a language model (template) that refers to fire incident report.

Training phase: The system presented here is responsible for acquiring reports and comments made by civilians about fire incidents at specific locations. In order to define the algorithms to be used, first it is needed to determine the requirements of these algorithms, the desired performance, and efficiency [16]. Consequently, as a first step, a training comment platform was constructed where users were asked to make some comments about a fire incident that they were witnessed hypothetically (see **Figure 2**). Moreover, they were asked to make some comments that use phrases that refer to fire reports, but the comment *should not* refer to a fire incident but to something else (see **Figure 3**). For example, "John has a burning desire to succeed in his new business" (here "burning" means "very strong").

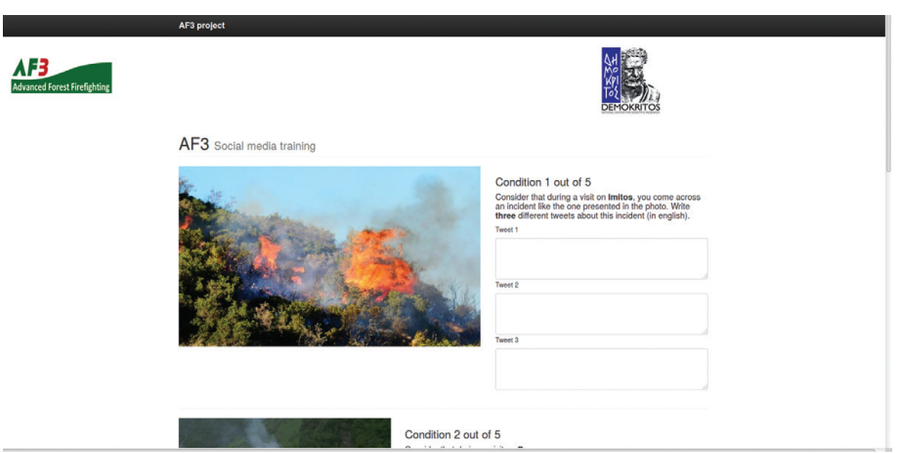

Figure 2.
Training comment platform: declaration of fire burst.

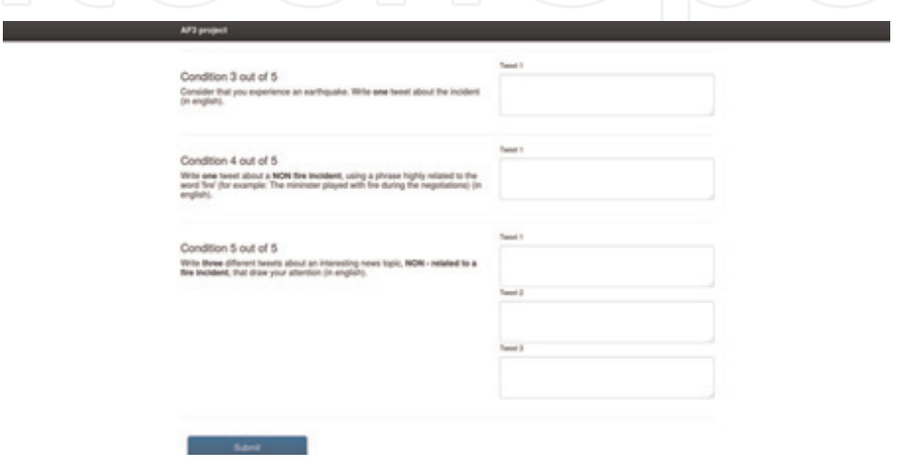

Figure 3.
Training comment platform: tricky "fire" word usage.

3.2.3 Figure training comment platform interface

The results of the training phase were passed through (a) sentence segmentation, (b) tokenization, (c) part of speech tagging, and (d) name entity detection algorithms, so consequently each report was converted to a tagged sentence form:

> E.g. <'I'> <'think'> <'there'> <'is', DEFINING VERB> <'fire', FIRE RELATED WORD> <'at'><'Immitos', LOCATION>

As a result, this procedure concluded to a set of tagged sentences that we know that they refer to fire incident report. Next, these reports were aggregated based on their similarity. Finally, the most common aggregated ones were kept in a regular expression form in order to represent the variations. These aggregated rules correspond to the relation rules that will be used by the relation recognition step of the information extraction module. The selected rules are the following:

> 1. <FIRE RELATED WORD> <EXCLAMATION MARK> * <TIME> + EXCLAMATION MARK> * <VERB LOCATION DEFINITION> + <PREPOSITION> + < HASHTAG> + <LOCATION>

> 2. <FIRE RELATED NOUN> <EXCLAMATION MARK> * <FIRE RELATED VERB>

> 3. <FIRE RELATED NOUN> <EXCLAMATION MARK> *<VERB RELATED TO SMOKE> + <PREPOSITION> + <HASHTAG> + <LOCATION>

> 4. <LOCATION> <EXCLAMATION MARK> * <SENSITIVE AREA> + <FIRE EXPRESSION>

> 5. <LOCATION> <EXCLAMATION MARK> * <SENSITIVE AREA> + < EXCLAMATION MARK> * <HASHTAG> + <FIRE RELATED NOUN>

> 6. <LOCATION> <EXCLAMATION MARK> * <FOREST> + <EXCLAMATION> * < FIRE RELATED VERB> <EXCLAMATION MARK> * <HASHTAG> + <FIRE RELATED NOUN>

> 7. <SENSITIVE AREA> + <EXCLAMATION MARK> * <FIRE EXPRESSION> < EXCLAMATION MARK> * <HASHTAG> + <LOCATION>

where

FIRE-RELATED NOUN: 'fire', 'flames', 'smoke', etc.

VERB LOCATION DEFINITION: verbs that define location ('exists', 'is located', 'is', etc.)

FIRE LOCATION VERB: ‘burn’, ‘fire’, etc.
VERB-RELATED TO SMOKE: ‘covering’, ‘smoke’, etc.
SENSITIVE AREA: forest, trees, park, etc.

3.3 Fire incident aggregation and potential fire incident prediction

3.3.1 Overview

In the previous section, the procedure of fire incident report acquisition was presented. The result is the gathering of various fire incident reports on different locations with different timestamps. Despite the fact that these reports may seem reliable, due to the severity of the situation, there would be cases, however, that a report may indeed refer to a false fire incident, either because of false fire incident detection from the information extraction component or because of a false report by a civilian [17]. It should be highlighted here that a false report is not made intentionally (like fake news, e.g., as examined in Section 2), but it is an outcome of misunderstanding or a tricky usage of the word fire and its derivatives (i.e., pants on fire). In order to ensure that fire incident notification alerts correspond to a noteworthy event, such reports should be checked of their validity before they are reported to the ingestion server. Consequently, the system consists of an analytic process responsible for the confirmation of the reports based on the number and the location of them. The analytic process implements a reliability model which aggregates the reports and concludes to a fire incident event report along with a reliability score. The reliability score corresponds to the level of how many trustful reports of fire incidents refer to a specific location. The reliability model is presented in more detail in the next section.

3.3.2 Implementation

Initially, the analytic process clusters incident reports based on their geo-coordinates (longitude, latitude). Due to the fact that fire incident reports usually are distributed densely along the fire locations, DBSCAN algorithm [18] was used for report clustering, which is a very efficient dense-based unsupervised classification algorithm for two-dimensional spaces and Euclidean distance as proximity measure and is able to detect accurately various cluster shapes. Then, for each cluster, the reliability model is applied where, finally, a geographical area that it is suspected of being threatened by fire incident is estimated, along with a reliability score.

3.3.3 Reliability model

The reliability model was designed by assuming that very few reports for specific location probably would mean that these reports are probably false alarms, but above a specific threshold, it is almost clear that there is a significant number of people reported a fire incident. In other words if, for example, there emerges one tweet referring to a great fire at the center of Athens, apparently there would be doubts about the validity of this report. Probably, we would say that either this report was a joke or the author of this comment might mean something different that of the literal meaning of a fire incident. On the other hand, if 100 tweets reported a fire incident, probably a real fire incident in the center of Athens is very likely. Apparently, some more tweets would not do the difference. As a result, an exponential model was selected which is parameterized by:

- *Low threshold*: The bottom threshold of the number or reports, where below of it these reports are considered unreliable
- *High threshold*: The upper threshold of the number of reports, where above of it these reports are considered very reliable
- *Low threshold probability (Pl)*: reliability corresponding to the low threshold
- *High threshold probability (Ph)*: reliability corresponding to the high threshold

The reliability score is given by

$$\text{Reliability score} = 1 - b \cdot e^{a \cdot NoR}$$

The term *NoR* stands for the number of results. In case of

$$\text{NoR} = \text{low threshold then we set reliability score} \leftarrow \text{Ph} \tag{1}$$

$$\text{NoR} = \text{high threshold then we set reliability score} \leftarrow \text{Pl} \tag{2}$$

Thus:

$$\text{Eq.}(2) \leftrightarrow 1 - b \cdot e^{a \cdot low \text{ threshold}} = Ph \leftrightarrow \ln((1 - Pl))/b) = a \cdot low\ threshold \tag{3}$$

Similarly:

$$Eq.(3) \leftrightarrow \ln((1 - Ph))/b) = a \cdot high\ threshold \tag{4}$$

$$Eq.(4), Eq.(5) \rightarrow \frac{\ln((1 - Pl))/b)}{\ln((1 - Ph))/b)} = \frac{(\ln(1 - Pl) - \ln(b))}{(\ln(1 - Ph) - \ln(b))} = \frac{Low\ threshold}{High\ threshold} \tag{5}$$

Let:

$$\text{c} = \frac{Low\ threshold}{High\ threshold} \tag{6}$$

Then:

$$Eq.(6), Eq.(7) \rightarrow \ln(1 - Pl) - \ln(b) = c \cdot \ln(1 - Ph) - c \cdot \ln(b) \leftrightarrow b = e^{\frac{(c \cdot \ln(1 - ph) - \ln(1 - Pl))}{c - 1}} \tag{7}$$

Moreover:

$$\text{Eq.}(5), \text{Eq.}(8) \rightarrow a = \frac{1}{High\ threshold} \cdot \ln((1 - Ph))/b) \tag{8}$$

4. Overall proposed scheme for Twitter post-based fire burst detection

Based on the system architectures presented in Sections 2 and 3, we propose a hybrid architecture for detecting fire bursts in real time based on Twitter posts. The proposed architecture can be divided into two parts: a deep learning scheme for distinguishing false from valid Twitter posts and a typical NLP scheme for

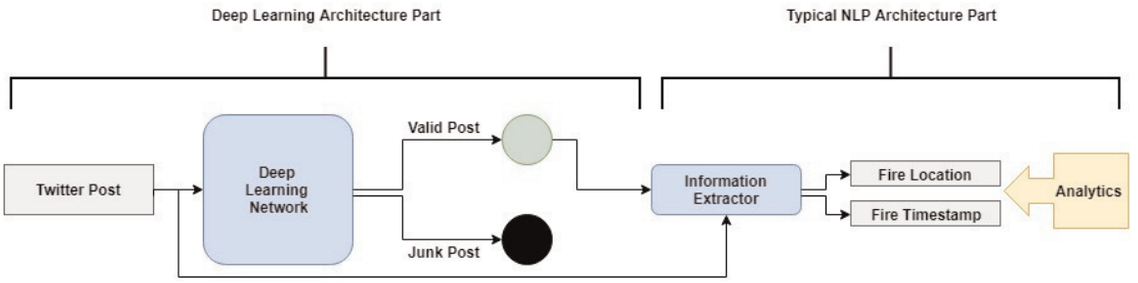

Figure 4.
Proposed overall architecture.

extracting the crucial information with respect to the declared fire burst post. The overall combined scheme is illustrated in **Figure 4**. The deep learning network part represents the scheme presented in Section 2, while the information extractor of the typical NLP part represents the scheme presented in Section 3.

For the fake post detection part, we are to recruit the aforementioned deep learning scheme as it performs twice as good as the related NLP-based methods [19]. Thus, Twitter post processing is expected to work much faster than in the case of implementing a typical NLP-based procedure of the state of the art. In addition, the availability of large posts/news datasets [10–12] facilitates the reliable training of such systems.

Despite the current trend of massively turning to deep neural networks, we designed and constructed a rather typical NLP-based architecture for the information extraction part of our system. This is highly related to the prerequisites that the training procedure of a deep neural network sets, as well as the nature of the problem itself. To begin with, due to lack of a publicly available (i.e., dataset containing a large number of fire burst-related Twitter posts), appropriate dataset for this task, a deep learning approach would be one of only few chances of success. More importantly, the nature of the task itself points to the direction we followed; fire-related posts on a social media platform are reasonably expected to have some common characteristics that make it suitable for a human to model them in order to obtain the desired information. For example, such posts are expected to be short in length, declaring the area of the fire source while containing words and phrases from a fire-related expression set of manageable size. So, our NLP-based subsystem is human and not machine modeled, is proven to be efficient, and is human intuitive and understandable, something that makes it easier to manipulate and expand, if needed.

5. Validation

The system was tested during the AF3 pilot exercise in Skaramagas naval base in two scenarios: (a) fire incident indication based on reports coming from mobile app and (b) fire incident indication based on reports coming from Twitter posts (tweets) containing the hashtag #af3EUprojectFireDetection_TRIAL.

During the first scenario test, a controlled fire was set at an open area inside the naval base. After a while actors, members of the pilot exercise, pretending to be citizens passing by, started posting reports about the fire incident they witnessed. These posts were analyzed by the fire incident detection module and return a notification of a potential of fire incident along with the estimated location and a reliability score. The results were visualized by the public information channel, where fire incident notifications were presented on the map as an area that it was

Figure 5.
Validating the scenarios.

Source	Expected results	Validation
Fire incident indication based on reports coming from Twitter posts (tweets)	• Collect the post coming from Twitter and containing the hashtag #af3EUprojectFireDetection_TRIAL • Pass these posts through the information extraction sub-module in order to distinguish the tweets that referred to fire incidents from the ones that did not • Cluster the posts referring to fire incidents, and detect fire incident areas along with the corresponding reliability score • Send result to ingestion server via the REST API	Done successfully

Table 2.
Validation results of the NLP-based scheme.

estimated that the fire was located along with the post comments of the reports, photos attached with the reports, and the reliability score (see **Figure 5**).

During the second scenario test, a controlled fire was set at an open area near the military airport in Aktio [20]. After a while actors, members of the pilot exercise, similarly with the first scenario, committed posts about the fire incident on the Twitter instead of the mobile app. These tweets were collected by the fire incident detection component, analyzed, and distinguished the ones that refer to the fire incident. These reports were gathered by the analytic module and, as described above, clustered, and finally the corresponding notifications were sent to the ingestion server. The results, similar to the first case, were visualized by the public information channel and exploited by the data fusion component in order to enhance its estimation. **Table 2** illustrates the validation results.

6. Conclusions

Fire bursts are a dangerous problem of great importance worldwide. Mega fires often result in significant environmental destructions, major damages on infrastructures, and economic loss. Most importantly, they put at stake the lives, not only of the civilians but also of the forest fire personnel. Thus, technologies that facilitate early fire detection are important for reducing fires and their negative effects.

This chapter aims to provide an alternative view for early fire detection based on twitter posts, instead of expensive sensors and other infrastructures. A hybrid system architecture is introduced which combines a deep learning process for the detection of valid twitter posts regarding fire bursts and a NLP process which extracts the crucial information (place, time, etc.) from the valid tweets. Finally, risk assessment, based on analytics, is performed which derives the geographical places threatened by fire at the current time.

Part of the architecture is already validated under real-world conditions, and the results are promising. The overall system performance is expected to be further improved once the deep learning scheme is entirely utilized.

Acknowledgements

This work was performed within the AF3 Project (Advanced Forest Fire Fighting), with the support of the European Commission by means of the Seventh Framework Programme (FP7), under Grant Agreement No. 607276.

Conflict of interest

There are no "conflict of interest" issues regarding this chapter.

A. Appendices and nomenclature

The mathematical definition of the convolution process between two one-dimensional signals $f(t)$ and $g(t)$ follows in Eq. (9). The mathematics behind LSTM layer architecture follows in Eqs. (10)–(13). Functions σ and *tanh* represent the sigmoid and hyperbolic tangent function, respectively. Parameter W corresponds to weighting matrices:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau) g(t - \tau) d\tau \tag{9}$$

$$z_t = \sigma(W_z \cdot [O_{t-1}, O_t]) \tag{10}$$

$$r_t = \sigma(W_r \cdot [O_{t-1}, O_t]) \tag{11}$$

$$h'_t = \tanh(W \cdot [r_t * O_{t-1}, O_t]) \tag{12}$$

$$O_t = (1 - z_t) * O_{t-1} + z_t * h'_t \tag{13}$$

Author details

Konstantinos-George Thanos*, Andrianna Polydouri, Antonios Danelakis, Dimitris Kyriazanos and Stelios C.A. Thomopoulos
Integrated Systems Laboratory (ISL), Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens, Greece

*Address all correspondence to: giorgos.thanos@iit.demokritos.gr

References

[1] Titcomb, J, Carson, J. Fake News: What Exactly Is It—And How Can You Spot It? [Internet]. 2018. Available from: https://www.telegraph.co.uk/technology/0/fake-news-exactly-has-really-had-influence/ [Accessed: 16 January 2018]

[2] Singhania S, Fernandez N, Rao S. 3HAN: A deep neural network for fake news detection. In: Proceedings of the International Conference on Neural Information Processing (ICONIP 2017); 14-18 November 2017; Guangzhou, China. New York: Springer; 2018. pp. 572-581

[3] Wang WY. Liar, liar pants on fire. A new benchmark dataset for fake news detection. Computation and Language. 2017, arXiv preprint: 1-5. Available from: arXiv:1705.00648

[4] Shu K, Mahudeswaran D, Liu H. FakeNewsTracker: A tool for fake news collection, detection, and visualization. Computational and Mathematical Organization Theory. 2018:1-12. https://link.springer.com/article/10.1007/s10588-018-09280-3

[5] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997;**9**(8):1735-1780

[6] Popat K, Mukherjee S, Yates A, Weikum G. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. Computation and Language. 2018, arXiv preprint: 1-11. Available from: arXiv:1809.06416

[7] Oluwaseun A, Deepayan B, Shahrzad Z. Fake news identification on Twitter with hybrid CNN and RNN models. In: Proceedings of the 9th International Conference on Social Media and Society (SMSociety '18); 18-20 July 2018; Copenhagen, Denmark. New York: ACM; 2018. pp. 226-230

[8] Zhang J, Cui L, Fu Y, Gouza FB. Fake news detection with deep diffusive network model. Social and Information Networks. 2018, arXiv preprint: 1-10. Available from: arXiv:1805.08751

[9] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, Massachusetts, USA: MIT Press; 2016. pp. 326-716. Available from: http://www.deeplearningbook.org/ [Accessed: 23 January 2019]

[10] Gillin J. Politifact's Guide to Fake News Websites and What They Peddle [Internet]. 2017. Available from: https://www.politifact.com/punditfact/article/2017/apr/20/politifacts-guide-fake-news-websites-and-what-they/ [Accessed: 18 January 2019]

[11] University of Warwick. PHEME rumor dataset: Support, certainty and evidentially [Internet]. 2016. Available from: https://www.pheme.eu/2016/06/13/pheme-rumour-dataset-support-certainty-and-evidentiality/ [Accessed: 18 January 2019]

[12] Zubiaga A, Liakata M, Procter P, 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**:1-29. DOI: https://doi.org/10.1371/journal.pone.0150989

[13] Hsu ST, Moon C, Jones P, Samatova N. A hybrid CNN-RNN alignment model for phrase-aware sentence classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017); 3-7 April 2017; Valencia, Spain, Short Papers. Vol. 2. 2017. pp. 443-449

[14] Bird S, Klein E, Loper Ed. Natural Language Processing with Python. 1st ed. Sebastopol, California, USA: O' Reilly Media; 2009. Available from: http://www.nltk.org/book [Accessed: 23 January 2019]

[15] AF3 EU-Project. Advanced Forest FireFighting [Internet]. Available from: http://af3project.eu/ [Accessed: 23 January 2019]

[16] Imran M, Castillo C, Diaz F, Vieweg S. Processing social media messages in mass emergency: A survey. Social and Information Networks, 2015, arXiv preprint: 1-37. Available from: arXiv: 1407.7071

[17] Mendoza M, Poblete B, Castillo C. Twitter unser crisis: Can we trust what we RT? In: Proceedings of the First Workshop on Social Media Analytics (SOMA 2010); 25 July 2010; Washington DC, USA. Pennsylvania Plaza, New York City, USA: ACM; 2010. pp. 71-79

[18] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996); 2-4 Aug 1996; Oregon, USA. Vol. 969(34). 1996. pp. 226-231

[19] Oshikawa R, Qian J, Yang Wang W. A survey on natural language processing for fake news detection. Computation and Language, 2018, arXiv preprint: 1-11. Available from: arXiv:1811.00770

[20] Thomopoulos CAT, Kyriazanos DM, Astyakopoulos A, Lampropoulos V, Dimitros K, Margonis C, et al. OCULUS fire: A control and command system for fire management with crowd sourcing and social media interconnectivity. In: Proceedings of SPIE Defence, Security and Sensing (SPIE DSS 2016); 17-21 April 2016; Baltimore, Meryland, USA. Vol. 9842. 2016. p. 98420U