D2.4 Linguistic feature extraction tool

Report: Linguistic feature extraction tool

<table>
<thead>
<tr>
<th>Project acronym:</th>
<th>RED-Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project full title:</td>
<td>Real-time Early Detection and Alert System for Online Terrorist Content based on Natural Language Processing, Social Network Analysis, Artificial Intelligence and Complex Event Processing</td>
</tr>
<tr>
<td>Grant agreement no.:</td>
<td>740688</td>
</tr>
<tr>
<td>Responsible:</td>
<td>INSIKT</td>
</tr>
<tr>
<td>Contributors:</td>
<td>MAV, BCU</td>
</tr>
<tr>
<td>Document Reference:</td>
<td>D2.4</td>
</tr>
<tr>
<td>Dissemination Level:</td>
<td>CO</td>
</tr>
<tr>
<td>Version:</td>
<td>FINAL</td>
</tr>
<tr>
<td>Date:</td>
<td>31.07.2018</td>
</tr>
</tbody>
</table>
## History of Changes

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification reason</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01.06.2018</td>
<td>Document structure</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Yesika Laplaza (INSKT)</td>
</tr>
<tr>
<td>0.2</td>
<td>04.07.2018</td>
<td>First draft</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Berta Biescas (INSKT)</td>
</tr>
<tr>
<td>0.3</td>
<td>18.07.2018</td>
<td>First draft review</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Jennifer Woodard (INSKT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cristi Potlog (SIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dániel Abel (MAV)</td>
</tr>
<tr>
<td>0.4</td>
<td>18.07.2018</td>
<td>First draft review</td>
<td>Dániel Abel (MAV)</td>
</tr>
<tr>
<td>0.5</td>
<td>23.07.2018</td>
<td>Second draft</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Megan Lawrence (INSKT)</td>
</tr>
<tr>
<td>0.6</td>
<td>24.07.2018</td>
<td>Review of the content according to DoA</td>
<td>Adriana Prodan (SIV)</td>
</tr>
<tr>
<td>0.7</td>
<td>27.07.2018</td>
<td>Third draft</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
<tr>
<td>0.8</td>
<td>27.07.2018</td>
<td>Third draft review</td>
<td>Cristi Potlog (SIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adriana Prodan (SIV)</td>
</tr>
<tr>
<td>1.0</td>
<td>31.07.2018</td>
<td>Final updates</td>
<td>Guillem Garcia (INSKT)</td>
</tr>
</tbody>
</table>
# Table of contents

1. Executive Summary ................................................................. 7

2. Introduction ............................................................................. 9
   2.1. Document Scope .................................................................. 9
   2.2. Methodology ...................................................................... 9
   2.3. Document Structure .......................................................... 9

3. Overview of the task ............................................................... 11
   3.1. Role in the project ............................................................. 11
   3.2. Relation with other tasks of WP2 ...................................... 12
   3.3. Relation with WP4 (Complex Event Processing) ............... 13
   3.4. Relation with WP5 (Privacy, Visualization and Learning) .. 13
   3.5. Role in the project ............................................................. 13

4. State of the Art ......................................................................... 15
   4.1. Introduction ...................................................................... 15
   4.2. Feature Extraction in NLP .................................................. 15
   4.3. Deep Learning and Feature Extraction ......................... 16
   4.4. Relation with the methodology used in the project .......... 20

5. Summary of the Functionalities ................................................ 21

6. Linguistic Features Extraction Tool presentation ..................... 22
   6.1. Introduction ...................................................................... 22
   6.2. Objectives ........................................................................ 22
   6.3. Type of linguistic features .................................................. 22
   6.4. Relevant features for detecting terrorism-related content .. 24
   6.5. Feature description ............................................................ 24
   6.6. Deployment ...................................................................... 27

7. Demonstrator ........................................................................... 29
   7.1. Overview .......................................................................... 29
   7.2. Interface - WEB UI .......................................................... 30
   7.3. Interface - API ................................................................. 32
   7.4. Results .............................................................................. 33

8. Next steps ................................................................................. 35
   8.1. Next iterations ................................................................. 35

9. Conclusions .............................................................................. 37

10. References ............................................................................... 38
List of figures

Figure 1. Main modules of RED-Alert solution. ................................................................. 7
Figure 2. Functional objective of this task. ........................................................................ 8
Figure 3. Project Overview PERT Chart. ........................................................................... 11
Figure 4. Relation between WP2 and WP4. ...................................................................... 12
Figure 5. Scheme of an autoencoder. Image extracted from UFLDL Tutorial............... 17
Figure 6. Diagram of the RBM with 3 visible nodes and 4 hidden nodes....................... 18
Figure 7. Greedy learning for DBN. ................................................................................ 18
Figure 8. Figure: A common Convolutional Neural Network design............................. 19
Figure 9. Figure Representation of a Recurrent Neural Network. ................................ 20
Figure 11. Main components of task coding. ................................................................... 27
Figure 12. Components of the feature extractor. .............................................................. 28
Figure 13. Mock-up of the demo UI. ................................................................................ 29
Figure 14. Example of feature extractor output. ............................................................... 30
Figure 15. Main UI of the demo. ...................................................................................... 30
Figure 16. Demo UI. ........................................................................................................ 31
Figure 17. Example of output obtained in the demo. ........................................................ 31
Figure 18. Diagram of features ....................................................................................... 32
Figure 19. JSON output. .................................................................................................. 33
Figure 20. Accuracy improvement in English and Arabic............................................... 33
Figure 22. Planning of other languages........................................................................... 36
List of tables

Table 1. Supported languages. ............................................................................................................14
Table 2. Functional requirements. .......................................................................................................21
Table 3. List of linguistic features already extracted. .........................................................................25
Table 4. Description of all the linguistic features. ..............................................................................26
Table 5. Relation with terrorism propaganda....................................................................................27
Table 6. Demo parameters. ..................................................................................................................31
Table 7. Accuracy with and without linguistic features. .................................................................34
Table 8. Summary of the Project iterations. .......................................................................................35
Table 9. Main Methodology and Technology of each iteration..........................................................35
## List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag-of-Words model</td>
</tr>
<tr>
<td>CEP</td>
<td>Complex Event Processing</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>DBN</td>
<td>Deep Belief Network</td>
</tr>
<tr>
<td>IG</td>
<td>Information gain</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>KNN</td>
<td>K nearest neighbors</td>
</tr>
<tr>
<td>LEA</td>
<td>Law Enforcement Agencies</td>
</tr>
<tr>
<td>LSI</td>
<td>Latent Semantic Index</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NMT</td>
<td>Neural Machine Translation</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>RBM</td>
<td>Restricted Boltzmann Machine</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>SMA</td>
<td>Semantic Multimedia Analysis</td>
</tr>
<tr>
<td>SNA</td>
<td>Social Network Analysis</td>
</tr>
<tr>
<td>UFLDL</td>
<td>Unsupervised Feature Learning and Deep Learning</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
1. Executive Summary

The present document, Deliverable 2.4 “Linguistic Features Report” (henceforth referred to as D2.4), is a deliverable related to Task 2.3 “Implement linguistic feature identification and extraction” (henceforth referred to as T2.3). The implementation of T2.3 assumes to integrate a set of methods for extracting features or attributes from social media text contents. With this purpose, this task will include selected Natural Language Processing methods, from low-level methods (if the message includes hashtags, exclamations marks, etc.) to higher-level ones (entity extraction, topic classification, etc.).

Mainly, the objective of the Project is to understand and detect how terrorism-related content is spread across social media platforms and one of the main features is to analyze text content. The figure below represents the diagram of the whole process (see Deliverable 1.7-System architecture for a more detailed description of the modules).

![Diagram of the RED-Alert solution](image)

**Figure 1. Main modules of RED-Alert solution.**

The result of the Task 2.3 is a set of new features extracted from the text in order to improve the Deep Learning methodology that classifies the text into its level of suspiciousness (included in the NLP module, Task 2.2). These new features will enrich the text analysis by identifying text properties that will help detection of suspicious messages.

For example, in the messages for ISIS support, they usually use the pronouns to confront “we” (as the real Muslims) against “they” (the other). So, using these pronouns like that can be an “indicator” of pertaining to this group.

Similarly, other properties like text length, readability, use of articles, etc. allows us to understand and identify radical messages. In terms of technical development, this task will be integrated into the NLP module, in the following way:
The objective of the T2.3 is to improve the accuracy of the Machine Learning model, by adding new features that give more information about the text content for the message that will be analyzed.

In the context of the project, T2.3 is closely related to the Task 2.2 (Deep Learning methodologies). Both will be part of the NLP module to analyze text, extracting the threat score, a score related to the suspiciousness of the text.
2. Introduction

This section presents the scope of the current document, the methodology used to produce it and the structure of the following sections, in order to provide an overview for reviewers.

2.1. Document Scope

The purpose of this document is to present the work accomplished within Task 2.3 “Implement linguistic feature identification and extraction”. The objective of this task is to improve the NLP classifier that detects the “suspicousness” of the messages. This classifier is extensively explained in the D2.2- Deep learning algorithms.

2.2. Methodology

To guarantee the quality of the work done for this deliverable (D2.4), the following steps have been carried out:

1. Task planning. Understanding the task objectives and the possible methods to achieve them.
2. State of the Art. Investigation about the state of the art related to text feature extraction.
3. List of potential text features.
5. Development of code in Java.
6. Testing with real data.
7. Evaluation of results.
8. Selection of the most relevant features related to the topic of radical content detection.
9. Second Test with real data.
10. Adaptation to other languages (Arabic, Spanish and French).
11. Planning the next iterations.

To test the tool from a functional and performance point of view, the same data collected to build the Deep Learning models has been used (see D2.2 for a detailed description of this data). All this data has been collected from Twitter by means of the Twitter API, and anonymized by means of the tools developed during the project. Because all these tasks only need the text content of the messages, the anonymization doesn’t affect the quality of the data.

2.3. Document Structure

The document is structured into 10 sections:

1. Executive summary. To explain the main goals of the deliverable.
2. Introduction. It includes a description of the work done in the deliverable.
3. Overview of the task. Brief explanation of the deliverable task.
4. State of the art.
5. Summary of the Functionalities.
6. Linguistic features. Description of the methodology to extract the features.
7. Demo. Description of the demo for model testing.
8. Next steps. Definition of the roadmap for the next iterations.
10. References.
3. Overview of the task

3.1. Role in the project

Linguistic Features Extraction Tool belongs to WP2 components as depicted in the below picture.

![Project Overview PERT Chart](image)

Figure 3. Project Overview PERT Chart.

T2.3 presented in this deliverable is part of WP2 (Social Language Processing) composed by 3 different components:

1. NLP module based on Ontologies. An NLP engine to extract the most relevant features (concepts, topics, entities, etc.) by means on an Ontology-based methodology (See deliverable 2.1).
2. NLP module based on Deep Learning methods (See deliverable 2.2). This module also includes a NLP engine, but based on Deep Learning methods, complementary to the previous one, most focused on training models to detect suspicious text content.
3. Semantic Multimedia Analysis. A set of methods to extract information from the audio, video and images (Task 2.4).

In the context of the project, WP2 is the part that analyzes the content of messages, looking for traces to detect radical propaganda.
This WP2 is also related to WP4, in which the development and implementations of a CEP engine that can process events from NLP and SNA is carried out:

![Diagram showing the relation between WP2 and WP4]

**Figure 4. Relation between WP2 and WP4.**

The output of WP2 (a JSON containing all the information about the analysis of the messages) is the input of WP4- Complex Event Processing.

In that sense, the output of the task described in this deliverable (D2.4) will be part of this JSON.

### 3.2. Relation with other tasks of WP2

- The other tasks of WP 2 are: Task 2.1: Develop terrorism-specific classifiers
  Obviously, both tasks are related to NLP technologies and so they are related in the sense that they share some of the methods applied, like text pre-processing.

- Task 2.2: Implement deep learning algorithms
  The classification of the messages as suspicious or not suspicious, in order to detect radical content is calculated for each of them, and represents a previous and fundamental step for Task 2.2, where NLP techniques will be applied to extract quantitative information from the texts, converting messages into a vector in the Word Embedding Vector Space.

  In this respect, the work associated with D2.4 enriches this information by describing other properties of the text that are not currently taken into account by the Deep Learning Model.

- Task 2.4: Implement audio, image and video feature extraction
  The main relation between Task 2.4 and Task 2.3 is that, in some cases, the output of the task 2.4 is the input of the Task 2.3. For example, the ‘audio to text’ technology converts the conversation from a video into text, to be then analyzed by all the NLP modules (from task 2.1 to 2.3).
3.3. Relation with WP4 (Complex Event Processing)

Complex Event Processing (CEP) engine will ingest the result of all the other modules, including the NLP results. In that sense, the output of Task 2.3 will become the input of this CEP engine.

So this task is also related to WP4, where the development and implementation of a CEP engine that can process events from NLP analysis, including the extraction of linguistic features, will be performed. The classification model should serve as a pre-filter of the messages, videos or photos obtained from the different social networks selected for this project, which allows focusing on the analysis of those whose probability of being suspicious is higher.

In the case of multimedia data (audio, video or photography), the model will work with all the related textual data (author, comments, title of the publication, etc.) to perform a first filtering between suspects and non-suspects.

3.4. Relation with WP5 (Privacy, Visualization and Learning)

This task is related to WP5 in 2 ways:

A. To the privacy tool, because prior to extracting the linguistic features, the anonymization tool will remove all sensitive data in the message (including all the metadata related to personal information).

B. Learning process. The linguistic feature will be a part of the Meta-Learning process that will improve the accuracy of the model by:
   a. Including more annotated data into the model.
   b. Improve the results by correcting some model errors.

3.5. Role in the project

3.5.1. Milestones

If we take a closer look at the project milestones included in DoA, the main milestone related to Task 2.3 is Milestone MS2, Core Modules ready. This task will improve the module of NLP analysis, one of the main methodologies included as a core module.

3.5.2. Objectives

Regarding the main objectives of the project, Task 2.3 is aligned with the objectives: NLP building block. This task will improve the modules that detect radical messages by analyzing their content.

3.5.3. KPIs

If we go into detail on the project KPIs, this component needs to respect the KPI related to languages supported:

- At the end of M12, will be supported 4 languages
● At the end of M24, will be supported 6 languages
● At the end of M36, will be supported 10 languages

The list of languages that needs to be followed is detailed in D1.3 Technical Requirement Specification (see note in section 6.1.11. Language support).

At this moment, the Linguistic Features Extraction Tool supports the following languages marked as supported and the expected support of the other ones (in project month):

<table>
<thead>
<tr>
<th>Language</th>
<th>Supported</th>
<th>Expected date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>French</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Romanian</td>
<td></td>
<td>October 2018</td>
</tr>
<tr>
<td>Turkish</td>
<td></td>
<td>October 2018</td>
</tr>
<tr>
<td>German</td>
<td></td>
<td>January 2019</td>
</tr>
<tr>
<td>Russian</td>
<td></td>
<td>January 2019</td>
</tr>
<tr>
<td>Hebrew</td>
<td></td>
<td>June 2019</td>
</tr>
<tr>
<td>Ukrainian</td>
<td></td>
<td>June 2019</td>
</tr>
</tbody>
</table>

*Table 1. Supported languages.*

### 3.5.4. Technical risks

In relation to the project’s technical risks, Task 2.3 is closely related to the second risk from DoA, namely: Insufficient corpus size of real terrorism content to create models and ontologies.

Deep Learning methodology to classify messages according to their threat score requires annotated data to build the model. Thanks to adding more features, the models will be improved without the need to add more annotated data of suspicious messages. This is important when data is not easy to acquire, for example, for minority languages.
4. State of the Art

4.1. Introduction

The feature extraction of a text is one of the tasks that help in the classification of texts. Traditionally, it has been approached from the NLP point of view, where an expert person created a series of rules based on features of the corpus that allowed such classification. The results obtained with these techniques are good, but they require time and supervision to generate optimal rules.

In recent years, the use of Deep Learning has revolutionized this area of NLP. Neural networks have been used to address the task of classifying texts through feature extraction. This technique allows extracting patterns for the classification of texts in an unsupervised way, considerably reducing the training time and obtaining in most cases significantly better results compared to more traditional techniques.

The following shows how this task has been approached from the NLP point of view as the Deep Learning.

4.2. Feature Extraction in NLP

Feature extraction is a task that can be approached from different points of view. Please find below the methods most used by the scientific community to carry out the feature extraction based on NLP techniques.

- **Filtering method.** It is a method that allows extracting features of large corpora quickly. For text filtering, different processes can be used, and the following are the three most frequent:
  - **Word frequency:** These methods take into account the frequency of appearance of each of the words in the text, and eliminate those whose frequency is low, since the elimination of these terms reduces the spatial dimensionality of the features. On the contrary, it has been observed, in information retrieval studies, that these rare words can provide important information. For this reason, it is not appropriate to eliminate or discard certain words based on the number of occurrences.
  - **Mutual information:** It is used to measure the differences in the characteristics of the subjects. It is applied to represent the relationships between the information and the statistical measurement of the correlation of two random variables. These models are based on the hypothesis that words have a frequency of appearance according to the subject being treated, that is, words like ‘bomb’, ‘assault’ or ‘ammunition’ will recurrently appear in texts that deal with war, and will practically not appear in other types of texts, such as those on medicine.
○ **Information gain (IG):** It is a method widely used in text filtering. Measures the information obtained to predict the category by means of the presence or absence of a word in the document⁴.

- **Fusion method.** This method uses a weighting for each of the characteristics observed in the corpus. This weighting can vary during the training until it adjusts. The main drawback of this method is that it requires considerable time to extract the textual characteristics, so they are not convenient for dealing with large corpora.
  ○ **Weighted KNN (K nearest neighbors):** It is used for the classification of continuous cumulative values. It is based on the statistical performance of pattern recognition, obtaining good results in both accuracy and recovery⁵.

- **Mapping method:** It has been used mainly to work with latent semantic index (LSI) and principal component analysis (PCA), obtaining good results to text classification⁶.
  ○ **Latent semantic analysis:** It is a method that uses a statistical approach to improve the search for information. It consists of reducing the dimensionality of the problem of finding information to overcome the difficulties related to synonymy and polysemy⁷ (multiple meanings). This method, apart from the extraction of information, has also been used in other areas such as filtering information, classification and grouping of texts, or indexes of documents, among others.
  ○ **Principal component analysis:** It searches for the projection according to which the data is best represented in terms of least square, converting a set of observations of variables possibly correlated in a set of variables values without linear correlation called principal components.

- **Clustering method:** This method compares the features of a text with the text features of the cluster. The main drawback shown by this method is the high temporal complexity they need.
  ○ **Chi-Square clustering method:** Each word is assigned a CHI value for each class. By grouping these words by their CHI value, a classification model is created in which the pattern of each word of the corpus corresponds to a dimension in the conventional algorithm. This particular method requires a relatively low temporal complexity⁸.
  ○ **Concept indexing:** The base for this model is the center of each class as a vector subspace, to further map each text vector with the subspace created previously, thus acquiring a representation of text vectors with respect to this subspace. The algorithm manages to reduce the vector space, since the subspace of the concept indexation is generally smaller than the vector space of the text. Each class center is considered as a concept, and the process of mapping the text vector is considered an indexation to this vector space⁹.

### 4.3. Deep Learning and Feature Extraction

The more traditional extraction methods, based on NLP, require a long process in the design of functions, since they must be performed manually from a previous knowledge acquired
when studying and analyzing the work corpus. On the other hand, the use of Deep Learning methods allows for a more efficient and faster way from a corpus of training (Liang et al., 2017).

The most remarkable aspect of the methods based on Deep Learning is that the big data features are automatically learnt, without the need of human supervision, which considerably reduces the time needed to create them.

The most popular Deep Learning methods used for text feature extraction are shown in the successive sub-sections.

- **Autoencoder:**
  It is one of the most used tools to implement Deep Learning. They are often used as 3-layer neural networks, in which only one is hidden. These systems learn to produce in the output exactly the same information they receive at the entrance, with the peculiarity that they use a reduced number of neurons to encode this information, which forces the system to look for an intermediate representation of the information. This process is carried out in a completely automatic way, without the need for supervision, nor of previous examples to learn.

  ![Figure 5. Scheme of an autoencoder. Image extracted from UFLDL Tutorial](image)

  The neural network is forced to generalize, and find patterns. These patterns being the most frequent features observed in the corpus. This method, as compared to others, has improved both recognition accuracy and generalization, providing greater stability in the results.

- **Restricted Boltzmann Machine (RBM)**
  It is a neuronal network formed of two layers, where one is visible while the other remains hidden. Each visible node is connecting to all hidden nodes, this connection being reciprocal. To make learning easier, this model presents a restriction: no visible node can be connected to another visible node, just as a hidden node cannot be connected to another hidden node.

  The visible layer is the input corpus to be analyzed, while the hidden layer represents the latent factors that must be learned. To achieve this learning, each entry node has a weight and a bias, and these are adjusted until the reconstruction and input are as similar as possible.
This method is mainly used to reduce the dimensionality of vectors, feature learning, classification tasks and topic modeling among other aspects related to the NLP. This neural network represents a new type of tool to work with machine learning, a great power of representation, which is why it has been used as a feature extractor in a great variety of classification problems, obtaining very good results, being the best obtained through advanced models of superficial learning.

- **Deep Belief Network (DBN)**

It emerged from the possibility of stacking the RBMs and the fact that these could be trained in a greedy way.

It is a type of deep neural network composed of multiple layers of latent variables, and similar to the manner in which RBM have connections between the layers, but not between units within each layer.

The training system of this method is divided into three steps. In the first step, thanks to a Contrastive divergence algorithm, the neural network is able to learn a layer of characteristics of the visible unit. During the second step, from the functions previously trained, it learns the
characteristics of the functions of a second hidden layer. Finally, the entire DBM is trained to achieve learning for the final hidden layer.\(^{14}\)

Some studies that have used this model for classification tasks, such as Reuters-21578 and 20 Newsgroup corpus, show significantly better results than those obtained with other traditional classifiers.\(^{15}^{16}\)

- **Convolutional Neural Network (CNN)**

Convolutional neural networks are a variation of a multilayer perceptron applied in two-dimensional matrices.

As classification networks, in case of CNN there is a first phase of feature extraction, composed of convolutional neurons and sampling reduction, where as the data progresses along this phase, its dimensionality/volume is diminished. This means that the farther layers are less sensitive to possible disturbances of the input data, and, at the same time, they are activated by more complex characteristics. In the end, the network consists of the simple perceptron neurons, which perform the final classification based on the characteristics extracted in the initial phase.

The CNNs are able to learn to classify all types of data distributed continuously along the input map, and at the same time, be statistically similar anywhere in the input map. For this reason, CNNs are especially efficient in classified and segmented images.

---

![Convolutional Neural Network design](image)

**Figure 8. Figure: A common Convolutional Neural Network design.**\(^{17}\)

- **Recurrent Neural Network (RNN)**

Recurrent nets are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or numerical time series data emanating from sensors, stock markets and government agencies (Eclipse Deeplearning4j Development Team).

This algorithm takes as input not only the current input but also the one that has been previously perceived, that is, the recurrent networks have two input sources: the present and the recent past, and the combination of both determines how to respond to the new data. The sequential information of each comment is stored in a hidden state, which allows for covering many steps in the process for each example. Because of this feedback, each hidden state contains information about the previous hidden state, and all previous ones, while the memory lasts. The fact that it can use its internal memory to process arbitrary input sequences, makes it an algorithm especially appropriate for tasks in which the data to be...
predicted or analyzed can be arranged in the form of a signal (temporal variable). In a series of letters, a recurring algorithm will use the first character to determine which will be the second.

Figure 9. Figure Representation of a Recurrent Neural Network

4.4. Relation with the methodology used in the project

Some parts of these methodologies already explained in the State of the Art are used in the tool developed for the RED-Alert solution, specifically:

- Methods for carrying out the feature extraction based on NLP techniques.

During the tasks, a filtering method has been used because it has some advantages, in comparison with the other methods:

- The main reason is being the faster method. This is critical for real-time processing, one of the main goals of the RED-Alert solution.
- Also it is very easy to apply the same methodology to other languages, logically, changing some of the parameters of the model.

- Deep Learning feature extraction methods

The Autoencoder methodology is used in the project, due to the following:

- It is fully unsupervised, meaning there is no need of labeled data.
- It is the simpler methodology to apply.
5. Summary of the Functionalities

This section gives a brief summary of the functionalities covered by the Linguistic Features Extraction tool. These functionalities are referred to according to the Requirements Specification from “D1.3 – Technical Requirements Specification”.

<table>
<thead>
<tr>
<th>Code</th>
<th>Functional requirement description</th>
<th>Comments</th>
</tr>
</thead>
</table>
| FR13 | Domain and topic categorization, clustering and scoring  
Categorizes any given text according to its topics, type, ideological stream and sect (in Islamic documents) and other customizable domains. | The tool will help to categorize the text content of the document into 6 levels of “suspiciousness” by means of a threat score. |
| FR16 | Short message analysis:  
Ability of the system to analyse short messages (e.g. twitter posts) | The extraction of these features is especially relevant for the analysis of short text, because in this type of texts, the more information you can get from the text, the better. The tool will be addressed specifically for this type of texts. |

*Table 2. Functional requirements.*
6. Linguistic Features Extraction Tool presentation

6.1. Introduction

This chapter will first explain the objectives to be fulfilled when working with linguistic features, then will present a list with the linguistic features most used over the years in different projects and investigations, and finally all these linguistic features, will be detailed. It will indicate which of them are the most relevant and could be used in this Project to detect terrorist content in text messages from different social networks.

6.2. Objectives

The texts that can be obtained from social networks containing relevant information about the world, concrete actions, or more mundane aspects about preferences and tastes in society. Knowing how to extract all that information and understand it correctly is vital for understanding society and transcendent facts that affect you or not.

Based on the linguistic characteristics observed in a training corpus, a series of patterns can be extracted, from which rules are created that allow the content to be first understood, and later to classify the statements into the different groups or subgroups that have been previously established.

In this Project, the main objective of the linguistic characteristics is to contribute to the classification of the input statements, helping in the identification of terrorist content, and in the assignment of a threat score, according to the degree of radicalization that they present.

6.3. Type of linguistic features

The linguistic characteristics to be extracted from a corpus can be many and of diverse nature, depending on the objective for which the analysis and subsequent classification will be destined. Below is a detailed list of the most used features.

- **Word frequency**: As already mentioned in section 4.2 Feature Extraction in NLP, the measurement of frequent words in a text is a feature that can help in the classification of statements. Knowing the frequency of the words in a text allows you to decode it in a faster way.

- **Lexicological correlation**: There are 4 levels of correlation between sentences:
  - **Noun overlap**: it measures the frequency with which the same noun is shared between 2 sentences.
  - **Argument overlap**: it measures how often two sentences share nouns with common stems.
  - **Stem overlap**: Measures how often a noun in a sentence shares common root with other words from other sentences in the text.
  - **Content word overlap**: Measures how often sentences share content words. Greater word overlap helps build large units of meaning in a text, and is an indicator of paragraph boundaries.
- **Total number of words in the text:** Although it is not strictly a linguistic feature, it tends to be considered that the most extensive texts, with a greater number of words, tend to have a higher quality than shorter texts.

- **Bag-of-words model:** This method is used for the purpose of extracting functions from text documents, representing each statement or document in a bag of words. The algorithm only takes into account the frequency of words’ appearance, without paying attention to the order, in which they appear, nor to the grammar, or to the possible semantic relationship that may exist between them.

- **Topic model:** It is widely used in data mining and network analysis since it allows, through a statistical model, to identify the topic that a document presents. Starting from the premise that each text deals with a particular topic, the words will appear more frequently in each text depending on the topic that it deals with. The mathematical model, based on the statistics of words and the relations of similarity between words of each document analyzed, is able to extract the topics that are treated and the balance of topics of each document.

- **Latent Semantic Analysis:** This method is used for extracting and representing the contextual-usage meaning of the words by statistical computations applied to a large corpus of text. It is based on the idea that the contexts of words, in which a word can appear or not, provide sets of mutual constraints, which allow determining the similarities between a word and a set of words. LSA is closely related to neural net models, but is based on a mathematical matrix decomposition technique closely akin to factor analysis.

- **Part-of-Speech Tagger:** Each word of the text is assigned to a label according to its morphosyntactic function (name, verb, preposition, adverb, etc.) within the sentence. This technique can be applied with different objectives, on the one hand it allows to disambiguate the polysemic words that can appear in the text and, on the other hand, it can be applied to language models based on n-grams. Besides taking into account the context in which the words are found, it also considers their grammatical category to make a better prediction.

- **Decision tree:** It is a prediction model used in different fields, not only in linguistics. A series of diagrams of logical constructions are created from a data set, similar to prediction systems based on rules, with which a series of conditions that take place successively can be represented and categorized.

- **Sentiment analysis:** The basic task in feeling analysis is to classify a text according to the polarity it presents: positive, negative or neutral. Apart from polarity, it can also be classified according to the emotions that the text presents: the 5 basic emotions that are usually treated more frequently being joy, anger, sadness, surprise, and fear. Normally, sentiment analysis approaches are usually grouped into 4 main categories: keyword location, lexicon affinity, statistical methods and concept-level techniques. This technique shows a probability of success around 79%.
6.4. Relevant features for detecting terrorism-related content

The **number of words** a text can contain cannot be considered a decisive factor in this project, since it works with different social networks, some of which have strict restrictions on the amount of characters that users can use in their writings. Despite this, this factor can be useful in order to know what type of text is going to be analyzed (post of a blog, Facebook, Twitter, etc.). In the case of Twitter messages, although they have few words, the fact of knowing how many exact words the publication presents can also be a factor to be taken into account, since in many publications only mentions appear to other users, or mentions and URLs, as for example ‘@lamAmmarY*** @Alishbabukh*** @sakina*** @Komi*** @sunnishar*** @farahsy*** @DrSaqla*** @JN*** @SNA11*** https://t.co/zXCNCRhtSz’. Publications that meet these characteristics could be directly labeled as non-suspect, and not analyzed by the system.

**Emotions** play an important role in DAESH’s propaganda, since the use of these allows them to sympathize with social movements and recruit new members\(^{24}\). Depending on the topic they deal with, they use some emotions or other, for example when they talk about the Islamic State and what life is like there, they do it in a very idyllic way, with feelings of love and hope. On the other hand, when they talk about other countries, the society in these countries, their rulers, religion, the emotion that most predominates is anger. During the Project, based on the topics that are identified in each publication and what feelings do prevail in them, the correct classification of the messages can be refined.

### 6.5. Feature description

#### 6.5.1. Description

The features already implemented in the NLP part of the system are:

<table>
<thead>
<tr>
<th>(N)</th>
<th><strong>Column Title</strong></th>
<th><strong>Data type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>First Singular Pronoun Score</td>
<td>pronounFirstSingularScore</td>
<td>From 0 to 1</td>
</tr>
<tr>
<td>Generality A Score</td>
<td>generalityAScore</td>
<td>From 0 to 1</td>
</tr>
<tr>
<td>Contains emoticons</td>
<td>containsEmoticons</td>
<td>yes/no</td>
</tr>
<tr>
<td>Generality The Score</td>
<td>generalityTheScore</td>
<td>From 0 to 1</td>
</tr>
<tr>
<td>Contains Mention</td>
<td>containsMention</td>
<td>yes/no</td>
</tr>
<tr>
<td>Third Plural Pronoun Score</td>
<td>pronounThirdPluralScore</td>
<td>From 0 to 1</td>
</tr>
<tr>
<td>Contains numeric</td>
<td>containsNumeric</td>
<td>yes/no</td>
</tr>
<tr>
<td>Contains Rt</td>
<td>wordRt</td>
<td>yes/no</td>
</tr>
<tr>
<td>Contains Intensifier</td>
<td>containsIntensifier</td>
<td>yes/no</td>
</tr>
<tr>
<td>Contains Spread</td>
<td>wordSpread</td>
<td>yes/no</td>
</tr>
<tr>
<td>Second Pronoun Score</td>
<td>pronounSecondScore</td>
<td>From 0 to 1</td>
</tr>
<tr>
<td>Contains Hashtags</td>
<td>containsHashtag</td>
<td>yes/no</td>
</tr>
<tr>
<td>Informative Noun Score</td>
<td>informNounScore</td>
<td>From 0 to 1</td>
</tr>
</tbody>
</table>
Table 3. List of linguistic features already extracted.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains please rt</td>
<td>wordPleaseRt yes/no</td>
</tr>
<tr>
<td>Flesch Reading Ease Readability</td>
<td>readabilityRE From 0 to 100</td>
</tr>
<tr>
<td>Text Length</td>
<td>textLength From 0 to inf</td>
</tr>
<tr>
<td>Contains pls</td>
<td>wordPls yes/no</td>
</tr>
<tr>
<td>First Plural Pronoun Score</td>
<td>pronounFirstPluralScore From 0 to 1</td>
</tr>
<tr>
<td>Informativeness Adverb Score</td>
<td>informAdvScore From 0 to 1</td>
</tr>
<tr>
<td>Informativeness Score</td>
<td>informativenessScore From 0 to 1</td>
</tr>
<tr>
<td>Third Singular Pronoun Score</td>
<td>pronounThirdSingularScore From 0 to 1</td>
</tr>
<tr>
<td>Informativeness Verb Score</td>
<td>informVerbScore From 0 to 1</td>
</tr>
<tr>
<td>Contains Slang</td>
<td>containsSlang yes/no</td>
</tr>
<tr>
<td>Contains Urls</td>
<td>containsUrl yes/no</td>
</tr>
<tr>
<td>Flesch-Kincaid Grade Level Readability</td>
<td>readabilityFKRA From 0 to inf</td>
</tr>
<tr>
<td>Informativeness Adjective Score</td>
<td>informAdjScore From 0 to 1</td>
</tr>
<tr>
<td>Contains retweet</td>
<td>wordRetweet yes/no</td>
</tr>
</tbody>
</table>

In terms of the description of each one:

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First Singular Pronoun Score Text containing more singular pronouns than the average.</td>
</tr>
<tr>
<td>2</td>
<td>Generality A Score Frequency of the “a” article in the text.</td>
</tr>
<tr>
<td>3</td>
<td>Contains emoticons If the text contains emoticons.</td>
</tr>
<tr>
<td>4</td>
<td>Generality The Score Frequency of the “the” article in the text.</td>
</tr>
<tr>
<td>5</td>
<td>Contains Mention If the text contains mentions to other authors.</td>
</tr>
<tr>
<td>6</td>
<td>Third Plural Pronoun Score Frequency of the third plural pronouns.</td>
</tr>
<tr>
<td>7</td>
<td>Contains numeric If the text contains numeric values.</td>
</tr>
<tr>
<td>8</td>
<td>Contains Rt If the text is a retweet (only for Twitter).</td>
</tr>
<tr>
<td>9</td>
<td>Contains Intensifier If the text contains intensifies (very, a lot of, huge, etc.).</td>
</tr>
<tr>
<td>10</td>
<td>Contains Spread The text contains some particles to spread the message.</td>
</tr>
<tr>
<td>11</td>
<td>Second Pronoun Score Frequency of the second pronouns.</td>
</tr>
<tr>
<td>12</td>
<td>Contains Hashtags If the text contains hashtags (only for Twitter, Facebook and Instagram).</td>
</tr>
<tr>
<td>13</td>
<td>Informative Noun Score Informativeness of the nouns included in the text, defined as their “rarity” (frequency in a general corpus).</td>
</tr>
<tr>
<td>14</td>
<td>Contains please rt If the text contains please rt (only for Twitter).</td>
</tr>
<tr>
<td>15</td>
<td>Flesch Reading Ease Readability Score Score designed to indicate how difficult a passage in English is to understand.</td>
</tr>
<tr>
<td>Text Length</td>
<td>Length of the text.</td>
</tr>
<tr>
<td>Contains pls</td>
<td>If the text contains pls (only for Twitter).</td>
</tr>
<tr>
<td>First Plural Pronoun Score</td>
<td>Frequency of the first plural pronouns.</td>
</tr>
<tr>
<td>Informativeness Adverb Score</td>
<td>Informativeness of the adverbs included in the text, defined as their “rarity” (frequency in a general corpus).</td>
</tr>
<tr>
<td>Informativeness Score</td>
<td>General informativeness of all the terms.</td>
</tr>
<tr>
<td>Third Singular Pronoun Score</td>
<td>Frequency of the third singular pronouns.</td>
</tr>
<tr>
<td>Informativeness Verb Score</td>
<td>Informativeness of the verbs included in the text.</td>
</tr>
<tr>
<td>Contains Slang</td>
<td>If the text contains slang.</td>
</tr>
<tr>
<td>Contains Urls</td>
<td>If the text contains urls.</td>
</tr>
<tr>
<td>Flesch-Kincaid Grade Level Readability</td>
<td>Score of the readability value of a text, expressed as (in English): $0.39 \left( \frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left( \frac{\text{total syllables}}{\text{total words}} \right) - 15.59$</td>
</tr>
<tr>
<td>Informativeness Adjective Score</td>
<td>Informativeness of the adjectives included in the text.</td>
</tr>
<tr>
<td>Contains retweet</td>
<td>If the text contains the word re-tweet.</td>
</tr>
</tbody>
</table>

Table 4. Description of all the linguistic features.

### 6.5.2. Languages

Some of the features are language dependent:

- A - the score has their equivalent in other languages, for example in Spanish, un – el.
- Flesch–Kincaid grade and Flesch Reading Ease Readability Score have calculation for different languages.\(^{25}\)

In the chapter 7.2 “Next iterations”, there is a description of the expected roadmap to implement the methods for each language defined as priorities.

### 6.5.3. Relation with terrorism related content

The extraction of these features will help to improve the accuracy of the threat score classification models.

<table>
<thead>
<tr>
<th>Type of feature</th>
<th>Relation with terrorism related content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of the Pronouns</td>
<td>Usually, in the texts related to radical groups, they use pronouns of “we” (as the “chosen”) in opposition to “they” (the enemy).</td>
</tr>
<tr>
<td>Emoticons</td>
<td>They almost never use emoticons.</td>
</tr>
<tr>
<td>Mentions</td>
<td>Some mentions to specific users are very representatives of these messages.</td>
</tr>
<tr>
<td>Contains intensifier</td>
<td>The average frequency of the use of intensifiers is higher in this type of messages.</td>
</tr>
<tr>
<td>Contains Spread</td>
<td>They almost never use this kind of expressions.</td>
</tr>
</tbody>
</table>
Contains Hashtags | Hashtags are very common, even including some general ones not related to the content of the message, with the aim to get a larger potential audience.
---|---
Informativeness | Usually the informativeness of the adjectives is higher than the average.
Contains Slang | They almost never use slang expressions.
Contains Urls | It is very common to include urls.
Flesch ReadingEase Readability Score | In general, the score is lower to the average (difficult to read).

Table 5. Relation with terrorism propaganda.

6.6. Deployment

6.6.1. Overview

The deployment has consisted in the integration of the classification model into the NLP analysis system, coded in Java.

So, all the process is coded as Java classes, in particular:

![Figure 10. Main components of task coding.](image)

**Text.** The deployment starts with a text and a set of parameters (type of text, language, method of preprocessing, etc.).

**Parameters.** Some parameters must be specified according to the type of input message (language, type of text, etc.).

**Text preprocessing.** All the text is pre-processed in order to improve the accuracy of the feature extraction.

**Linguistic Features.** Collection of information of the text, including all the linguistic features explained in the previous section.
**JSON output.** Format of the output, including information about the features and the likelihood of the result (a kind of level of confidence of the result).

### 6.6.2. Feature Extractor

Once the text is pre-processed, the Java class classifier extracts all the linguistic features from this text, through a model trained with annotated data.

![Figure 11. Components of the feature extractor.](image)

The output of the classifier is a Java object that represents the value of each of the linguistic features explained previously.
7. Demonstrator

7.1. Overview

The main features of the demonstrator are:

**Input:** Text

**Output:** Linguistic features + Threat Score

**Demo:** The user inputs a text in the box, presses the button and the results are viewed on the right.

![Mock-up of the demo UI.](image)

The script to extract these results from the text is coded in Java and shared on the internal Git repository of the project.

Example of output JSON

```json
{
    "pronounFirstSingularScore": 0.44,
    "generalityAScore": 0.87,
    "containsEmoticons": "yes",
    "generalityTheScore": 1.0,
    "containsMention": "no",
    "pronounThirdPluralScore": 0.0,
    "containsNumeric": "no",
    "wordRt": "no",
    "containsIntensifier": "no",
    "wordSpread": "no",
    "pronounSecondScore": 0.0,
    "containsHashtag": "no",
    "informNounScore": 1.0,
    "wordPleaseRt": "no",
    "readabilityRE": 34.09,
}
```
This JSON is represented in the demo as values of all the fields in a more visual way.

7.2. Interface - WEB UI

7.2.1. Main screen mock-up

The main screen mock-up of the demo is as follows:

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Input text from the User to be analysed.</td>
</tr>
<tr>
<td>language</td>
<td>Dropdown</td>
<td>Supported text language. Possible values: (en, ar).</td>
</tr>
</tbody>
</table>
```

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 740688
7.2.2. Demo’s UI

The user inputs the text into a box and chooses the parameters.

![Demo UI](image)

**Figure 15. Demo UI.**

Main UI of the demo.

Once the user presses the button “extract”, the results are presented in the boxes on the right (side) of the UI.

Example of text output:

![Example Text Output](image)

**Figure 16. Example of output obtained in the demo.**

The feature outcomes are grouped by the score type.
7.3. Interface - API

7.3.1. API registry

<table>
<thead>
<tr>
<th>Path</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/api/linguistic-feature/extract</td>
<td>GET</td>
<td>Analyse text and respond with features and scores.</td>
</tr>
</tbody>
</table>

7.3.2. Extract API response model

They are 3 types of features, according to the type of number that represent each one:

- Boolean features (yes/no)
- Scores, from 0 to 1
- Scores, from 0 to infinite

![Diagram of features](image)

7.3.3. Extract API response JSON example

The output JSON has the following structure:

```json
{
  "threatScore":1,
  "contentSpec":{
    "containsEmoticons":true,
    "containsHashtag":false,
    "containsIntensifier":false,
    "containsMention":false,
```

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 740688
7.4. Results

The results of including the features extraction in the threat score model building is showing that there is an increase in the accuracy of all the models. Despite that, this increase is, in general, very low.

Figure 18. JSON output.

Figure 19. Accuracy improvement in English and Arabic.
Looking at the accuracy provided by each method:

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (English)</th>
<th>Accuracy (Arabic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without features</td>
<td>60.70%</td>
<td>71.14%</td>
</tr>
<tr>
<td>Top 10 features</td>
<td>62.21%</td>
<td>72.29%</td>
</tr>
<tr>
<td>Top 20</td>
<td>62.53%</td>
<td>72.40%</td>
</tr>
<tr>
<td>All the features</td>
<td>62.60%</td>
<td>72.55%</td>
</tr>
</tbody>
</table>

*Table 7. Accuracy with and without linguistic features.*

According to these results, more effort has to be done, by means of:

- Extract other features more relevant to the topic.
- Use different strategies to the different languages.
8. Next steps

8.1. Next iterations

WP2 tasks are scheduled in 3 iterations based on different objectives:

<table>
<thead>
<tr>
<th>Based on</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Iteration 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ends</td>
<td>LEA specifications</td>
<td>Integration feedback</td>
<td>LEAs pilot feedback</td>
</tr>
<tr>
<td>Objective</td>
<td>Meet LEA needs and requirements</td>
<td>Meet technical integration into the general tool</td>
<td>Meet the final testing issues from the LEAs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Month</th>
<th>Month 14</th>
<th>Month 28</th>
<th>Month 36</th>
</tr>
</thead>
</table>

Table 8. Summary of the Project iterations.

Generally speaking, the 3 main milestones that define the 3 iterations are, in chronological order:

1. Specifications: once the specifications are defined, the first iteration is based on fulfilling them.
2. Technical integration: the second iteration is devoted to integrating all the modules into a single product.
3. Testing: the third iteration is focused on fixing and improving the prototype thanks to the LEAs’ feedback provided during the pilots.

As previously explained, Task 2.3 is also based on this 3 iteration approach:

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 1</td>
<td>Build the methodology to extract the features (English, Arabic, Spanish and French). Test the tool and evaluate the results. Create the demo.</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>Extend the methodology to all the priority languages. Extend the methodology to formal text.</td>
</tr>
<tr>
<td>Iteration 3</td>
<td>Refine the models according to LEAs needs in terms of accuracy, processing time, etc.</td>
</tr>
</tbody>
</table>

Table 9. Main Methodology and Technology of each iteration.

During the first iteration, all technology is implemented in a local environment, as the results do not have to be shared with other modules.

The second iteration implies the integration with the other modules and work on real cases, meaning to work in a parallel approach to deal with large volumes of data. In that case, a Spark framework will be implemented to work in a distributed environment.

During the third stage of the Project, the module will be tested in real scenarios and all bugs will be fixed.

In terms of languages, that’s the roadmap of the implementation in other languages.
<table>
<thead>
<tr>
<th>Language</th>
<th>2018</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>M</td>
<td>J</td>
<td>J</td>
<td>A</td>
<td>S</td>
<td>O</td>
<td>N</td>
<td>D</td>
<td>J</td>
<td>F</td>
<td>M</td>
<td>A</td>
<td>M</td>
<td>J</td>
</tr>
<tr>
<td>ENGLISH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARABIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPANISH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRENCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROMANIAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TURKISH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GERMAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUSSIAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEBREW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UKRAINIAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 20. Planning of other languages.
9. Conclusions

The task of linguistic feature extraction is important for the improvement of the threat score models, i.e. the models that detect suspicious messages by analyzing their text content.

This set of features extracted from the text will be integrated into the tool and available for future users, helping them to identify suspicious messages.

During the first iteration, a software to extract these features has been developed, for English, Arabic, Spanish and French. The main tasks for the following iterations (before the pilots) will be:

- To create and test the feature extraction of the remainder of the priority languages.
- To extract more features and test them with the purpose to establish whether they can be relevant for the detection of suspicious messages, or not.

These improvements will be implemented also in other models that have to be developed in the following iterations, like the models applied to:

- New sources of information.
- New type of messages (long texts).

All these improvements will be included in the future versions of the D2.4 Deliverable.
10. References


[12] https://deeplearning4j.org/restrictedboltzmannmachine


