beAWARE

Enhancing decision support and management services in extreme weather climate events

700475

D3.3
Basic techniques for content distillation from multilingual textual and audiovisual material

Dissemination level:	Public
Contractual date of delivery:	Month 16, 31 June2018
Actual date of delivery:	Month 16, 31 June2018
Workpackage:	WP3: Early warning generation
Task:	T3.2 - Concept and conceptual relation extraction from textual information T3.3 - Concept and event detection from multimedia
Type:	Report
Approval Status:	Final draft
Version:	0.7
Number of pages:	123
Filename:	D3.3_beaware_basic_techniques_for_content_ distillation_from_multilingual_and_audiovisual_material_31-05-2018_v0.7.pdf

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 700475

Abstract

The deliverable will reflect the baseline approaches and their performance to concept extraction, parsing and concept relation extraction from written and transcribed textual material, as well as to multimedia concept detection and summarize the progress of the material annotation task after the first half of the lifetime of the project.

Co-funded by the European Union

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 700475

History

Version	Date	Description / Reason of change	Revised by
V0.1	16.04.2018	Deliverable outline	UPF
V0.2	20.04.2018	Initial version and assignments distribution	CERTH
V0.3	08.05.2018	Incorporate information from all partners	All partners
V0.3-0.4	15.05.2018	1st Internal review	CERTH
V0.5	22.05.2018	Revised version incorporating all partners enhanchments	All partners
V0.5-0.6	25.05.2018	2nd Internal review	CERTH - IOSB
V0.7	31.05.2018	Final version	All partners

Author List

Organisation	Name	Contact Information
UPF	Stamatia Dasiopoulou	stamatia.dasiopoulou@upf.edu
UPF	Simon Mille	simon.mille@upf.edu
UPF	Beatriz Fisas Elizalde	beatriz.fisas@upf.edu
UPF	Ivan Latorre Negrell	ivan.latorre@upf.edu
UPF	Laura Pérez Mayos	laura.perezm@upf.edu
UPF	Leo Wanner	leo.wanner@upf.edu
CERTH	Konstantinos Avgerinakis	koafgeri@iti.gr
CERTH	Panagiotis Giannakeris	giannakeris@iti.gr
CERTH	Manos Michail	michem@iti.gr
CERTH	Gerasimos Antzoulatos	gantzoulatos@iti.gr

Executive Summary

This deliverable reports on the basic techniques for concept and conceptual relation extraction from multimedia and textual content. Specifically, methods for the analysis of textual (texts, SMS messages, social media post and transcribed spoken language) and multimedia data are profoundly described in this deliverable. The goal of textual extraction is to extract concept and conceptual relation amongst the acquired textual information, while methods for mutlimedia analysis focus on the detection of concepts from visual and audio content.

The document describes in detail the WP3 modules, which are related to T3.2 and T3.3 and the appropriate approaches, components, and resources that were adopted so as to accomplish the respective functionalities that were described in the DoA and later on the ones that documented from the users throughout the compiled user requirements (D7.1, D7.2). The deliverable introduces the basic techniques for textual and multimedia concept extraction that were deployed during the first phase of the project's lifetime, for the implementation of the 1st prototype (M18). Furthermore, a description of the analysis requirements for visual (i.e. image/video), audio and text is provided and analyzed appropriately. While, for each module an overview of the State-of-the-Art (SoA) and a comparison to other approaches is included. The evaluation approaches and results are finally explained and demonstrated at the end of the document.

More specifically, the following modules are described in further details:

a) The concept extraction module from **visual** content (image/video), which includes the dynamic texture recognition and localization in videos, a fire and flood detection system in social media with the goal to identify people and vehicles in danger and the deployment of a traffic management application that estimates speed and abnormal events from surveillance cameras.
b) The **Automatic Speech Recognition** (ASR) module, which is based on open-source framework CMU Sphinx and integrates expanded ASR dictionaries with missing words and especially location names in order to improve recognition accuracy and enable localization via speech transcriptions. A simple algorithm for automatic punctuation of speech has also been implemented in order to facilitate concept extraction (T3.2) based on the duration of silence intervals. An encoder has also been included, in order to convert input audio files into the appropriate format, as well as basic noise removal algorithms based on spectral subtraction have been deployed.
c) the text analysis module, which addresses the processing of the multilingual **textual** inputs including part-of-speech and morphology tagging, lemmatization, syntactic and semantic parsing and the translation of the resulting linguistic representations into a semantic one that captures the extracted entities and events to be fed to beAWARE knowledge base.

It is worth to note, that the performance of the above modules extensively evaluated in terms of their accuracy and the first experimental results are encouraging to continue to work on this direction.

Each partner has contributed equally to the completion of the two tasks of the WP3. UPF was responsible for the development and evaluation of the text analysis module (Task 3.2). CERTH was responsible for the development all the methodologies and modules for multimedia analysis as well as for the deployment of Automatic Speech Recognition (ASR) module (Task 3.3).

Abbreviations and Acronyms

AA	Activity Areas
API	Application Programming Interface
ASR	Automatic Speech Recognition
BoW	Bag-of-Words
CAP	Common Alert Protocol
CCTV	Closed Circuit TeleVision
CNN	Convolutional Neural Networks
CRF	Conditional Random Field
CWRT	CrossWords Reference Templates
DDP-HMM	Dependent Dirichlet Process-Hidden Markov Model
DRSs	Discourse Representation Structures
DUL	Dolce + DnS Ultralite
EL	Entity Linking
EM	Expectation Maximization
EmC	Emergency Classification
EmL	Emergency Localization
EM	Expectation Maximization algorithm
FC	Fully Connected
FN	False Negative
FP	False Positive
Fps	Frames per second
GMM	Gaussian Mixture Model
GMMs	Gaussian Mixture Models
GPD	Generalized Probabilistic Descent
GPR	Gaussian Process Regression
HMM	Hidden Markov Model
HoGP	Histograms of Grassmannian Points
HOOF	Histograms of Oriented Optical Flow
IoU	Intesection over Union
IRIs	Internationalized Resource Identifies
KB	Knowledge Base
KCF	Kernelized Correlation Filters
LAS	Labeled Attachment Score
LBPs	Local Binary Patterns
LDS	Linear Dynamical Systems
LDT	Linear Dynamic Texture
LOD	Linked Open Data
MAP	Maximum a Posteriori Adaptation
MCE	Minimum Classification Error
MCMC	Markov Chain Monte Carlo
MFCC	Mel-frequency cepstrum coefficients
MLLR	Maximum Likelihood Linear Regression

MLP	Multi-Layer Perceptron
MST	Minimum-Spanning Tree
MTA	Multilingual Text Analysis
NER	Named Entities Recognition
NEs	Named Entities
NLP	Natural Language Processing
NN	Neural Network
ObD	Object Detection
OWL	Web Ontology Language
PCA	Principal Component Analysis
POS	Part-Of-Speech tagging
PSAP	Public Safety Answering Point
PTB	Penn Treebank
RDF	Resource Description Format
ROI	Region Of Interest
SLIC	Simple Linear Iterative Clustering
SoA	State of the Art
STOEF	Spatio-Temporal Oriented Energy Features
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UAS	Unlabeled Attachment Score
UAV	Unmanned Aerial Vehicles
UD	Universal Dependency
URL	Uniform Resource Locator
VLBP	Volume Local Binary Patterns
VQ	Vector Quantization
WSD	Word Sense Disambiguation

Table of Contents

List of Figures

List of Tables

1 Introduction

This deliverable elaborates on the implementation of the basic techniques in WP3 during the first half of beAWARE project lifetime (M1-M17). Generally, the objective of WP3 is to provide the nessasary technological solutions that will allow beAWARE framework to provide early warning and decision support to authorities, PSAP operators and other stakeholders during pre-emergency and/or during the emergency phase.

The current report consists of the work that has already done so far in tasks T3.2 (Concept and conceptual relation extraction from textual information) and T3.3 (Concept and event detection from multimedia). As such, these tasks contribute to the 3rd Milestone MS3 "First Prototype" for the successful completion of the first SW development cycle of the project as shown in Figure 1.

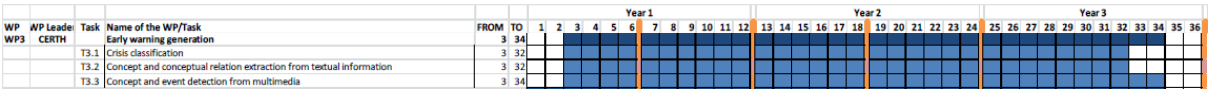

Figure 1: WP3 tasks and timeline

The tasks 3.2 and 3.3 of WP3 ***interact*** with almost all other WPs, especially with the tasks of *WP4 - Aggregation and semantic integration of emergency information for decision support and early warnings generation*, *WP5 - Multilingual report generation*, *WP6 - Main Public Safety Answering Point for emergency multimedia enriched calls* and *WP7 - System development, integration and evaluation* serving the objectives of beAWARE project.

1.1 Objectives

The objectives of Tasks 3.2 and 3.3 for the 1st period of the project are in aligned with the main goals, as they were described in the DoA, and summarised to the following:

- Extract event-centric information from multilingual textual inputs and project them onto an abstract representation that can be fed into the project ontologies (WP4).
- Develop the appropriate modules for concept and conceptual relation extraction for the beAWARE languages and domains.
- Deploy module for producing an integrated structure, which can be projected onto ontological representations (WP4).
- Develop computer vision and deep learning frameworks so as to detect crisis events in visual content (images and videos).
- Deploy an Automatic Speech Recognition (ASR) statistical method so as to transcribe voice messages for all supported languages in noisy environments and inform PSAP.

- Utilise multimodal fusion techniques to combine metadata from audio and visual channels and identify higher-level events in a specific crisis spatio-temporal area.

1.2 Results towards the foreseen objectives of beAWARE project

In order to achieve the abovementioned objectives, heterogeneous data from multiple resourses were collected and analysed. The Multilingual Text Analysis (MTA) processes textual inputs, namely tweets from social media, messages sent via the mobile beAWARE application, and transcribed spoken communications, and realizes the distillation of the conveyed information. The extracted information is event-centric, capturing what is happening (flood, traffic jam, etc.), the involved/impacted entities (e.g. people, buildings, cars, etc.) as well as location and temporal aspects. An uploaded audio file via the mobile beAWARE application is received by the Automatic Speech Recognition module for audio analysis. The transcription will then be analyzed by MTA in order to extract concepts such as locations, people and goods in danger etc. Similar, an image or video, which are sending to the beAWARE platform via mobile application, would be analysed utilising the appropriate componets in order to detect people or objects in danger and estimate their severity level.

More specifically, strong results have been obtained during the evalution of the image and video analysis components that include the fire and flood detection and localization modules, the people and vehicle detection systems and the traffic analysis and management pipelines, indicating further evidence of the applicability of the methods that were developed to extreme real life situations. ### Manos

Additionaly, regarding the text analysis, we have evaluated the generic UD-based pipelines at the surface-syntactic level for English, Greek, Italian and Spanish, and at the deep-syntactic level for English and Spanish (gold-standard reference corpora for Italian and Greek will be compiled during the second half of the project lifetime and the evaluation will be part of D3.4). Moreover, we have evaluated the English-specific pipeline, namely the Penn-Treebank-based one, which has been developed as part of the investigations into the performance trade-off between generic analysis pipelines, which can be easily ported and reused across languages, and analysis pipelines that are specific to a given language; during the second half of the project, respective evaluations for Spanish, further evaluations will be reported. The English-specific pipeline performs better than the generic UD-based one, but before reaching a conclusive obsercations, we need to study further the actual impact on the overall performance of the analysis component in beAWARE and validate the observations across the different languages.

1.3 Future plans

Although the evaluation process results indicate that all the components are in satisfactory condition, we have already planned to further elaborate and refine the proposed methodologies. Particularly, further ways to improve the developed visual analysis techniques have already been explored. These include the collection of additional training data to further fine tune the deep CNN architectures responsible for detection people and vehicles that are exposed to hazardous environments and the integration of holistic apparoaches in order to better analyse traffic motion patterns. Moreover, we plan to expand the current implementations in order to be applyied for the analysis of visual content from other sources, namely UAVs and fixed cameras. ## Manos

Additionaly, we are going to empower the text analysis with a localization strategy using OpenStreetMap data as the underlying reference knowledge base. Also, enhanchments to the semantic abstraction process, to the parsing evaluation metrics, to the tweet normalization and adaptations for spoken language parsing and finally to the compilation of annotated copora have been planed to do in order to increase the performance of the textual analytics component in the beAWARE framework.

1.4 Outline

The outline of this deliverable includes a briefly presentation of ***user requirements*** for the analysis of the textual, visual and audio content in order to enhance decision support and management services in extreme weather climate events as well as a description of the ***state-of-the-art*** methodologies in the scientific fields of computer vision, automatic speech recognition and text analysis. Each task is described in a different section of this document (Sections 4, 5, 6), evaluated in Section 7 and concluded in Section 8 with foreseen steps in the near future.

2 Content distillation requirements

In this Section we present the specifications and requirements for the analysis of the textual, visual and audio inputs considered in beAWARE.

2.1 Image and video analysis

Visual analysis is responsible to process images and videos obtained from static and aerial (e.g. UAV, satellite) cameras deployed on the pilot sites as well as reports sent via the beAWARE mobile application. The outcomes of the analysis process are intended to contribute to the detection of emergency events and the enhancement of contextual understanding, through the recognition of emergency indicative situations such as, traffic bottlenecks, flooded areas, fires, elements at possible risk (e.g. people, vehicles), etc. For that purpose, various compute vision and machine learning techniques have been deployed to meet the requirements that have been described formally in D2.1. In Table 1 a list of such requirements relative to visual analysis is presented. Many of these requirements are included also in D2.3 that addreses pilot use cases for the first prototype.

Table 1: Requerements relative to visual analysis.

UR#	UC#	Requirement name	Requirement description
UR_111	102	Detect flooded elements from video	Provide authorities with the ability to detect and count flooded elements (e.g. cars and people inside the river)from video and images sent from mobile phones and social media
UR_114	102, 103, 106	Detect water depth and velocity	Provide authorities with the ability to detect water depth and water velocity from video and images sent by the mobile app and social media
UR_118	106	River overtopping	Provide authorities/citizens with the ability to know if the river level is overtopping predefined alert thresholds
UR_123	106	Detect embankment exceeding	Provide authorities with the ability to detect from video, automatically (fixed and mobile cameras, social media and mobile app) , if a river embankment is overtopping and/o breaking
UR_201	201, 204	Detection of people and goods in danger	Display information authorities/first responders to detect people, cars and buildings in danger.
UR_205	201-202-204	Analysis of advancing fire	Provide authorities/first responders with an analysis of the the advancing fire (flame progression, height and length).

UR#	UC#	Requirement name	Requirement description
UR_207	201,202,204	Aerial images	Display authorities/first responders to visualize aerial images of the smoke and the trajectory flames. It will provide information about the extension and the damages (kind of damages, and so on), the tracking of the fire, vehicles and people around the spot, in order to find out possible suspects or victims. Furthermore, if these aerial images provide thermal information it can be used for looking over the fire perimeter once it has been extinguished, in order to locate sleeper fire and to avoid possible reproduction. This aerial images are a must, because the use case is in a forest, and we have not references in the forest, the only tool that can help the coordination center and first responders are aerial images to have information about forest fires (extension, direction of fires, damages, appropriate mobilization of resources, an soon)
UR_305	303, 304, 305	Possible locations for incidents	Display to the authorities' visual information about possible locations in the city (or outside the city) where a situation is more likely to develop that will require rescue team intervention (for example, based on past experience, traffic jam and/or accidents will be more likely to occur at a main street intersection/ public park/ entrance to hospitals or banks… etc.). In such cases a decision might be made to send rescue teams in advance to shorten response time if/when an incident occurs
UR_315	303, 304	Traffic Status	Display to the authorities to monitor the current traffic situation so that they can decide where to direct the first responders or inform them which roots to avoid
UR_316	305	Capacity of relief places	Provide to the authorities the current state of the available capacity of all relief places provided to the public

In order to meet the aforementioned requirements, the visual analysis components are expected to deploy relevant computer vision techniques for image classification, object detection, semantic segmentation, dynamic texture recognition and localization and motion analysis. The current version of the visual analysis components addresses many of the requirements presented above while the remaining ones are scheduled to be addressed by

future versions that will be implemented for the second and third prototype releases. More specifically related to UR_111 and UR_201, techniques for the analysis of video samples and images have been developed and examined later in Section 4.1 in order to localize targets in immediate inside hazardous regions. Additionly, we describe in Section 4.2 how spatio-temporal information is also leveraged in order to localize flooded or burning regions inside images and throughout video frames. For the purpose of meeting the UR_305 and UR_315 techniques for the analysis of static camera traffic videos from surveillance cameras has been developed so as to detect the traffic jam caused by power outage (traffic lights not working) or when many people leaving the city for seaside. Vehicle discrimination and traffic density could be determined throughout regular intervals of specific time periods. In order to detect the level of occupancy inside a place of relief as required in UR_316 the object detector described in detail in Section 4.1 is deployed so that people counting can thereafter take place.

UR_114, UR_118 and UR_123 related to static camera footage analysis for the monitoring and analysis of the water's depth are not currently addressed in this version but there are future plans that will be explored in order to achieve this functionality. The same applies to UR_205 and UR_207 which are related to the analysis of images and videos captures from UAV.

2.2 Audio analysis

The User Requirements that are relevant to Automatic Speech Recognition, as extracted from D2.1, are presented in Table 2. In all requirements, ASR has a partial contribution as will be described in detail in the following paragraph. It should be mentioned that, after discussions with end users, the ASR module will be used for the transcription of emergency audio recordings and not for online transcription of audio calls. Thus, in the following we will be refering to audio recordings only.

Regarding the localization of audio recordings (UR_107, UR_110, UR_333), there are two ways to extract location information.The first is geolocalization through gps trace, which is sent by the mobile device and the other one is semantic extraction of location information from audio trancriptions, which is performed by the MTA. The contribution of the ASR module in the second case is the transcription of speech to text in order to enable MTA to extract location information. Regarding the detection of people and goods in danger (UR_113, UR_201, UR_306, UR_318, UR_319) the ASR will also have the same contribution, as described previously, by providing the necessary text transcriptions to MTA.

Requirements UR_129, UR_224 are addressed in the sense that MTA is extracting notions from all supported languages (English, Spanish, Italian, Greek) and captures extracted info

into English in order to be stored in the KB. Again, the ASR has the same contribution as described previously.

Table 2: User Requirements for audio analysis

UR#	UC#	Requirement name	Requirement description
UR_107	102,103, 104, 105,106	Localize video, audio and images	Provide authorities with the ability to localize videos, audio and images sent by citizens from their mobile phones
UR_110	102	Localize calls	Provide authorities with the ability to localize Phone Calls to an emergency number concerning a flood event
UR_113	102	Detect element at risk from calls	Provide authorities with the ability to detect the number of element at risk andthe degree of emergencyfrom emergency calls
UR_129	All	Automatic translation from a foreigner applicant	Make easy the communication between people with different languages
UR_201	201, 204	Detection of people and goods in danger	Display information authorities/first responders to detect people, cars and buildings in danger.
UR_222	201,202	Filter of the emergency messages	Transfer emergency calls by writing (only minor emergencies or only information call). The aim is to save time operator and do not lose emergency calls
UR_224	201,202	Automatic translation from a foreigner applicant	Make easy the communication between people with different languages
UR_306	303, 305, 306	Number of people affected	Provide the authorities an estimation of the people that might be affected from the phenomenon and in which areas
UR_318	303, 306	Trapped citizens	Allow authorities to know if there are people trapped (e.g. in an elevator) and display where
UR_319	303, 306	Trapped elders at home	Allow authorities to know if there are elder people trapped in houses without an A/C and display where

UR#	UC#	Requirement name	Requirement description
UR_333	304, 305, 306	Localize calls	Provide authorities with the abilityto localize Phone Calls to an emergency number concerning citizens who are trapped

As far as audio analysis is concerned, the aforementioned requirements have been addressed in 1st Prototype of beAWARE system, by developing an ASR module able to transcribe audio in all four supported languages and integrating it in beaware platform, in order to receive audio files along with location, language and other relevant information, and communicate analysis results to MTA. However, in order for proper semantic extraction, speech recognition accuracy should be as accurate as possible. Thus, recognition accuracy will be improving throughout all the developing face until the second prototype, by adapting acoustic models to new recordings available, by expanding dictionaries and improving denoising algorithms. Specifically, for localization, the available language models already contain major location names (cities, districts) but CERTH has also started to expand ASR dictionaries in order to include as many location names as possible. However, this is an ongoing process and will be completed in the second prototype.

2.3 Text analysis

Text analysis addresses the processing of the multilingual textual inputs considered within the beAWARE system, that is, social media posts and messages (textual and transcribed calls) sent via the mobile application, and the extraction of information that contributes to situational awareness, such as what is happening (e.g., “an overflow”), where (e.g., “in Matteotti square”), what objects are involved (e.g., “sewers”), and so forth. The overall end goal is to avail of the real-time communication channels offered by social media and by the beAWARE mobile application in order to provide authorities better insights into the unfolding crisis, the elements at risk and the degree of emergency.

The following tables (Table 3 to Table 5) outline an indicative, but not exhaustive list, of information pertinent to the three pilots addressed with beAWARE based on the compiled Use Cases (UCs), as described in D2.1 and the refinements worked out in view of the first prototype, as reflected in D2.3. The “Message types” column lists indicative message categories that are of relevance for the emergency under consideration, while the “relevant notions” column outlines a breakdown of pertinent notions; indicative example inputs are shown in the third column.

Table 3: Examples of flood-pertinent information from text

Message types	Relevant notions (events, objects, …)	Exemplar inputs
▪ generic flood-related messages ▪ viability-related messages ▪ people in danger ▪ animals in danger ▪ river overtopping ▪ river breach ▪ bridge obstruction ▪ urban drainage ▪ …	events (flood, overflow, crack, interrupt, damage, etc.); infrastructure (streets, sewage network, bridges, airports, electricity & water supply network, buildings, etc.); people; animals; objects (cars, trunks, dumpsters, etc.); transportation (train, subway, bus network); anti-flooding devices (levees, embankments, etc.); rain; water level; …	▪ "Matteotti square is flooded." ▪ "Cars and dumpsters transported by the flow." ▪ "The embankment at Angeli bridge shows cracks." ▪ "Subway flooded. A car is trapped inside." ▪ "Sewer surcharge at Matteotti square." ▪ "Water has reached the level of cars. Traffic is interrupted." ▪ …

Table 4: Examples of fire-pertinent information from text

Message types	Relevant notions (events, objects, …)	Exemplar inputs
▪ generic fire-related reports (including possible causes, affected area, etc.) ▪ viability reports ▪ people in danger ▪ buildings in danger ▪ animals in danger ▪ …	events (fire, smoke, interrupt, damage, evacuate, etc.); people; animals; objects (cars, buildings, , etc.); traffic; weather aspects (temperature, wind, etc.); personal/material damages' ; …	▪ "I see smoke in Albufera national park." ▪ "The fire is heading to Pinedo." ▪ "The fire may get quickly out of control. There are very strong winds in the area." ▪ "About 20 square meters burned so far, mostly grass and scrubs." ▪ "Houses are in danger. We need to evacuate." ▪ …

Table 5: Examples of heatwave-pertinent information from text

Message types	Relevant notions (events, objects, …)	Exemplar inputs
▪ generic heatwave-related reports ▪ places of relief ▪ people in danger ▪ traffic jam problems ▪ electricity problems ▪ buildings with problems ▪ ...	events (power outage, traffic jam, etc.); infrastructure (places of relief, hospitals, etc.); transportation (bus, train, etc.); people; objects (cars, traffic lights; capacity; weather aspects (temperature, heat, etc.); …; …	▪ "Jammed in Toumpa's relief place. Why do they keep bring more people here?" ▪ "The relief place in Toumpa is full." ▪ "Man with breathing problems trapped in elevator." ▪ "Power outage. The traffic lights in Tsimiski Street don't work." ▪ "Stuck in traffic." ▪ …

As illustrated, information from textual inputs spans a variety of considerations and, thus, incidents that can be of relevance during the management of flood, fire or heatwave emergencies. In order to support the necessary conditions, namely:

- detection of elements at risk;
- localization of reported incidents;
- estimation of risk of emergency,

that underpin the user requirements as far as information extraction from text inputs is concerned across all three types of climate emergencies, as delineated in D2.1, text analysis needs to afford the means to support an adequate level of understanding and reasoning over the textual information. This translates to the requirements described in the following subsections.

2.3.1 Frame-based knowledge extraction

The results of text analysis are fed to the beAWARE knowledge base (KB), where semantic integration and reasoning take place. In order to enable the semantic integration of the information extracted from textual inputs, so that authorities can have a homogenous view of the unfolding crisis, and, subsequently, the semantic reasoning over the involved incidents and involved vulnerable objects, so as to further facilitate authorities in their decision making, text analysis needs to effectively extract entities and the relations between them.

In practical terms, this amounts to the identification of semantic frames (n-ary relational contexts), their participants as well as the semantic roles of these participants. For example, given the input *"The levee at Angeli bridge has collapsed"*, text analysis needs to identify "levee" and "Angeli bridge" as participants of the mentioned collapsing event; moreover, it needs to qualify "levee" as the entity that undergoes the collapsing and "Angeli bridge" as the place where the collapse happened, through the assignment of respective roles, namely that of "patient" and "location". It is important to stress that the accurate identification of the semantic roles is of equal importance to that of the participants in order to effectively capture the communicated semantics. If for instance, in our running example, "Angeli bridge" was the participant identified as the "patient" then this would mean that it is the bridge that collapsed, not the levee, a very different meaning and with very different, and crucial, implications within the crisis management context of beAWARE.

In the current implementation, as described in Section 6, the identification of relational contexts and their participants reflects directly the dependencies extracted by means of deep parsing, that is, no further processing is taking place in order to consolidate the resulting text analysis frame-based representations and ensure their semantic consistency and coherency. As a result, there is room for inaccuracies in the identification of both the

roles of participants (such as the above mentioned one, about the bridge having collapsed) as well as participants themselves (e.g. missing altogether the fact that "levee" participates in the collapse event, hence resulting not knowing what has collapsed). In the next versions that are planned for the second and third prototype releases, consistency enforcing strategies will be incrementally investigated for mitigating such phenomena and ensure robust and meaningful extraction.

2.3.2 Entity linking and disambiguation

The afore-described frame-based representations do not capture fully the underlying meaning though, unless the semantics of the identified relational contexts and their participants is determined. To accomplish this, the identity of Named Entities (NEs) mentions, i.e. that is mentions of proper names, needs to be determined, and the sense (meaning) of nominal entities is designated. Text analysis thus needs to determine, whether, for instance, a mention of "Matteotti" refers to Matteotti square in Vicenza or to the Italian politician Giacomo Matteotti; or, whether "Valencia" refers to the city of Valencia in the Iberian Peninsula or, among others, to the borough Valencia in Pennsylvania, United States, the football club of Valencia or the American, alternative rock, band Valencia, and so forth. Likewise, text analysis needs to distinguish whether the mention "bank" (e.g. in an input like "the water has reached the bank") refers to shore of a river or to a financial institution located in the affected area.

The designation of the underlying meanings is realized by means of reference (links) to respective sense (meaning) repositories, namely lexical and structured knowledge resources, such as WordNet[1], BabelNet (Navigli & Ponzetto, BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence, 2012), DBpedia[2] (which captures Wikipedia contents), etc. In the current implementation, linking against DBpedia is used for both entity linking and disambiguation (see Section 6). Towards the 2nd prototype, linking against BabelNet will be incorporated, in order to additionally avail of the cross-resource (including among others, WordNet, Wikipedia, GeoNames, VerbNet[3]) and the cross-language links it provides.

2.3.3 Location mentions recognition & geotagging

The ability to recognize mentions of places in textual inputs so as to determine the spatial context of the mentioned incident(s) is a key aspect when dealing with events, and even more so, within application contexts such as that of beAWARE, where the incoming reported

[1] https://wordnet.princeton.edu/
[2] http://wiki.dbpedia.org/
[3] https://verbs.colorado.edu/verbnet/

incidents need to be spatially clustered and positioned on the PSAP map. It is important to note here that for the considered textual inputs, there are two sources of location information. The first consists in coordinates metadata provided by the Twitter API, assuming that the user has turned on this option, and by the mobile application, for text messages and calls, provided the user device affords this capacity; besides availability considerations, it is worth keeping in mind that the location metadata do not necessarily coincide with the locations pertinent to the incidents reported in the message. The latter, that is mentions of place name in the message contents, is the second source of location information and the one of interest for text analysis purposes.

Although the processing of locations falls under the aforementioned NE recognition and entity linking tasks, its significance within the application context of beAWARE and the need for affording a wide coverage that goes well beyond the typical categories that are considered in these tasks (i.e., countries, cities, rivers, popular monuments, etc.), renders it a distinct requirement on its own. In particular, places of potential interest include roads, highways and bridges, public transportation network pertinent ones, such as bus, train and subway stations and stops, social infrastructure and amenities, such as hospitals, relief places, schools, parks and squares, cultural heritage monuments, places relevant to water, gas and electricity supply and distribution networks, to name but a few, including business names.

The breadth of places considered impacts not only the recognition but also the resolution of the identity and geographical coordinates of place mentions, as, the typically used reference resources, also lack the coverage required. DBpedia, for instance, misses key landmarks of the three pilot sites, namely Valencia, Vicenza, and Thessaloniki, or provides a very partial only coverage. For example, Angeli bridge (“ponte degli Angeli” in Italian) is not part of DBpedia, neither in its English version, where this is somewhat unsurprising given that the coverage for non-English contents is lower compared to English ones, but nor on the localized Italian version, where one would expect to find more localized information. Tsimiski street (“οδός Τσιμισκή” in Greek), a major avenue in Thessaloniki, is contained in DBpedia, but neither of the equally major, crossing, streets have an entry. As a result, there cannot be disambiguation through linking, nor any availing of possibly relevant interlinked datasets (e.g. GeoNames[4]) in the Linked Open Data (LOD) cloud[5].

In the current implementation, as described in Section 6, the focus has been on developing a basic framework for the recognition of place/location candidates and their preliminary

[4] http://www.geonames.org/
[5] https://lod-cloud.net/

linking, using DBpedia. Towards a more comprehensive and flexible recognition of place names and their geotagging, investigations will focus on a more intelligent location candidates identifications and on their linking against linked resources that afford the necessary coverage, namely OpenStreetMap[6] data, which preliminary on-going explorations have shown to meet the beAWARE needs.

2.3.4 Intra- and cross-language abstraction

Abstracting away from language specificities and distilling a structured representations of the information conveyed in the textual inputs is crucial in order to effectively cope with the richness of natural language and to accurately capture the intended meaning across phrasing variations. For instance, input messages reporting that "*the sewers are flooded*" and that "*the drainpipes have overflowed*" should result in the same normalised semantics-wise incident description, i.e. the flooding of the sewers. Likewise, for idiomatic expressions, such as "*raining cats and dogs*" and "στο κόκκινο ο υδράργυρος", which literally translates to "mercury in red", that should result in the extraction of a downpour and a high-temperature event respectively. However intra-language abstraction is not sufficient; given the multilingual application context of beAWARE and in order to enable the integration and subsequent reasoning over information originating in messages written in different languages, the aforementioned abstraction needs to further extend across languages.

2.3.5 Tweet normalization

The use of non-standard words (e.g., slang language, informal abbreviations, phonetic substitutions, alphanumeric tokens), misspelled words, hashtags, URLs, emoticons, usernames, etc., along with the frequent occurrence of ungrammatical sentence structures, make Twitter posts extremely noisy compared to typical written language (e.g. newspaper articles), necessitating their normalization prior to their analysis. Though not all of pertinent aspects impact analysis to the same extent, each comes with its own challenges. Hashtags, for example, may appear anywhere within a tweet, making it difficult to determine whether or not they form, linguistically-speaking part of the sentence structure; this is further aggravated by the challenges encountered in their segmentation, as the can be composed of one or more words, or even entire phrases, written without whitespaces or any obvious and consistent demarcation (camel case for instance, though frequent, is not a norm).

2.3.6 Open-domain analysis

Last but not least, and underlying all analysis aspects considered, is the need for decoupling the pertinent analysis tasks, and thus their performance, from domain-specific assumptions

[6] http://openstreetmapdata.com/

and enabling instead to analyze and extract information in an open-domain manner that considers the contents of the targeted inputs in their entirety and not selectively. Although domain-specific approaches can achieve very high accuracy, and overlooking re-usability and scalability considerations, the sheer gamut of situations that may be reported and be of relevance during the management of an emergency, renders tuning to closed-lists of predefined incidents, locations and impacted objects fairly impractical.

3 Relevant Work

In this section, we elaborate on the work that exists in the literature and is related to the tasks that are mentioned in WP3. For that purposes, we will see relevant work for: (i) visual concept detection, (ii) audio concept detection and (iii) textual concept detection.

3.1 Visual concept detection

State-of-the-Art for visual concept detection may include the analysis both image and video samples. Flood and fire detection will take place so as to contribute to PUC1 and PUC2, while traffic management is deployed so as to contribute to PUC3. Bellow, we analyze relevant work that focuses on these domains and more specifically: fire and flood detection in images and videos and then traffic management.

3.1.1 Fire and flood detection in social media images

Millions of images are daily uploaded on social media, while a great deal of them might include the existence of a crisis or emergency event. Taken this into account and inspired from the recent advance in image understanding, we suggest a novel framework that combines several technologies so as to detect and score the danger that people and vehicles might be in fire and flood scenarios solutions.

Semantic image segmentation SoA has also tend to use deep CNNs as well (Long, Shelhamer, & Darrell, 2015), (Ronneberger, Fischer, & Brox, 2015) by simply changing the objective of the classifier and label each pixel in the image individually, leading to a classification mask for the whole image instead of a recognition class. As far as security and safety domains are concerned, we scarcely find a technique that uses a deep CNN, as there are no groundtruth available masks and the training of these models is infeasible. A worth-to-note technique which performs fire detection in social images with the use of color and texture attributes was presented in (Chino, Avalhais, Rodrigues, & Traina, 2015).

Object detection on the other hand has numerous applications: autonomous vehicles, smart video surveillance, facial detection, ambient assisted living, etc. Naturally, deep CNN architectures were thoroughly examined for this. Early works such as (Girshick, 2015) include multi scale bounding box proposal generation techniques like Selective Search [13], as a feeding mechanism of candidate boxes to deep classifiers. The trend later became to incorporate this function into single shot object detectors, using end-to-end deep architectures (Ren, He, Girshick, & Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, 2015), (Redmon, Divvala, Girshick, & Farhadi, 2016), (Liu, et al., 2016). Those models achieved a better trade-off between accuracy and speed. In a previous work of ours (Avgerinakis, et al., Intelligent traffic city management from

surveillance systems (CERTH-ITI), 2017) we have proposed a novel scheme to detect vehicles and pedestrians from traffic surveillance cameras. The same framework has also been deployed in UA-DETRAC vehicle detection dataset (Wen, et al., 2015) achieving a really high detection rate.

3.1.2 Fire and flood detection in video samples

Dynamic texture recognition, localization, and more generally dynamic scene analysis in videos constitutes an intriguing topic within the computer vision community, due to its wide applicability in many scenarios. The term dynamic texture typically refers to moving textures, i.e. visual entities undergoing small, stochastic motions, encountered in real world indoor and outdoor environments. Current work mainly focuses on outdoor scenarios where a crisis event might occur (i.e. fire in a forest, a flooded river etc.), so we mostly examine classes of this category, even though, several instances of dynamic textures appearing in indoors videos, are also examined so as to prove our algorithm's efficacy and generalization. The automatic recognition of such textures has recently attracted attention, as it can provide a significant contribution to many real-world outdoor applications involving: scene analysis containing objects with high varying textures (e.g. water, smoke, trees), security applications for the prevention of a possible terrorist act and surveillance systems, responsible for the avoidance of natural disasters (e.g. fire in the forest or floods).

Dynamic texture recognition methods can roughly be separated into two main categories according to their adopted underlying model. The first category refers to Generative models which involve the extraction of global features throughout video sequences and their modeling is based on some hidden parameters (Fritz, Leibe, Caputo, & Schiele, 2005). Recent works such as (Doretto, Chiuso, Wu, & Soatto, 2003) use the spatiotemporal dynamics to train a Gauss-Markov recognition model, while (Chan & Vasconcelos, 2009) propose an expectation maximization (EM) algorithm to train the parameters of a statistical model. In (Chan & Vasconcelos, 2008) a Linear Dynamic Texture (LDT) scheme is proposed in order to represent a stochastic model of different appearance and motion dynamics. Lately, Linear Dynamical Systems (LDS) raised a lot of attention within this category, with the work of (Mumtaz, Coviello, Lanckriet, & Chan, Clustering Dynamic Textures with the Hierarchical EM Algorithm for Modeling Video, 2013) being a representative example. In their work, an hierarchical EM algorithm is deployed in order to cluster and learn the statistical model of the motion dynamics. LDS has recently been extended into a stabilized higher order LDS (shLDS) in (Dimitropoulos, Barmpoutis, Kitsikidis, & Grammalidis, 2017), who introduced Histograms of Grassmannian Points (HoGP). However, despite its high accuracy rates the method is computational costly, making it inappropriate for real-time applications.

While generative models seem quite promising for representing dynamic textures, their application to classifying the wider set of motion patterns found in dynamic scenes has been shown to perform poorly (Shroff, Turaga, & Chellappa, 2010). The complex, stochastic character of dynamic textures makes their precise modeling very challenging, so a second category of dynamic texture representation, namely Discriminative models has been considered. This category is based on the extraction of local, spatio-temporal features to describe moving texture dynamics by estimating local variations and statistics of intensity and optical flow values. Early techniques involved the accumulation of local spatio-temporal features using appearance features like GIST (Oliva & Torralba, 2001), motion histograms, such as the Histograms of Oriented Optical Flow (HOOF) (Chen, Zhao, Salo, Rahtu, & Pietikainen, 2013), swarm-intelligence (Kaltsa, Briassouli, Kompatsiaris, Hadjileontiadis, & Strintzis, 2015), spatio-temporal oriented energy features (STOEF) (Derpanis, Lecce, Daniilidis, & Wildes, 2012), and their successful and highly accurate Bag-of-Words (BoW) extension proposed in (Feichtenhofer, Pinz, & Wildes, 2014), named spatial energies. However, the coarse quantization of GIST and the rotation invariance of HOOF do not allow them to detect dynamic textures with accuracy, while on the other hand, the highly accurate STOEF, spatial energies and swarm dynamics suffer from computational efficiency making them inappropriate for real case implementations, such as surveillance and security scenarios.

Accurate texture classiffication has been achieved in images using Local Binary Patterns (LBPs), whose promising results have led to a number of its extensions as a dynamic texture descriptor. Volume Local Binary Patterns (VLBP) (Zhao & Pietikainen, 2006) and LBP-TOP (Zhao, Ahonen, Matas, & Pietikainen, 2012) are among the earlier methods, however they can easily reach a dimensionality of 2^{14} to 2^{26}, which is impractical in real-world applications involving large amounts of data that are to be processed in near real time. More recently in (Mettes, Tan, & Veltkamp, Water detection through spatio-temporal invariant descriptors, 2017) a hybrid spatio-temporal extension of LBP was introduced, which stacks the descriptor in time to obtain temporal information. Even though, the method achieved very high accuracy rates when discriminating between water and non-water scenes, its highly tailored character to exclusively water class, makes it inappropriate for more general classiffication and localization scenarios.

3.1.3 Traffic analysis and management

Smart city technologies for assistive transportation and safe driving, make up one of the most intriguing domains of computer science and have attracted significant attention during the last decade. Video surveillance, along with various other types of monitoring infrastructure provide a huge amount of exploitable data for extracting optimal traffic

management rules, increasing safety in busy streets, detecting or predicting and preventing accidents and numerous other applications of traffic monitoring. Moreover, increasing

industry trends towards autonomous driving, vehicles, and transportation in general, is changing the landscape of traffic analysis. The visual content from traffic cameras will, in the near future, also be used to manage autonomous vehicle navigation, by sending information about events elsewhere in the city, traffic conditions, pedestrian congestion, to optimally guide vehicles. The automated analysis of visual traffic data is necessary to extract useful information in a reasonable amount of time and with minimum human involvement in these cumbersome and extremely time consuming tasks. Many computer vision algorithms have already been developed for the automated analysis of traffic video data. Examples such as automatic vehicle detection and tracking, speed and traffic flow analysis, detection of abnormal events, have been developed and their levels of accuracy are continuously increasing. A big challenge, however, lies in the development of fast and computationally efficient methods to be used in actual real world scenarios that demand near real time solutions.

Speed Estimation

Traffic flow analysis from surveillance cameras can be decomposed to many different aspects of traffic understanding, such as vehicle detection and tracking, counting, traffic level classification and speed estimation. We focus here on a brief review of the existing methods for speed estimation. This task involves the translation of the displacement of pixels that belong to vehicles, into the real distances traveled and so, it relies heavily on proper camera calibration. As a result, most of the proposed algorithms focus on techniques for accurate retrieval of camera intrinsic and extrinsic parameters, as well as inference of the scene scale since we are only interested in the analysis of videos taken from a single monocular camera.

Methods are generally categorized into semi-automatic and fully-automatic. In the first case most of the calculations are performed automatically, but a user's manual input is required usually in the form of some known distance in the scene. In a method from this category, (Nurhadiyatna, et al., 2013) detected and tracked the vehicles using GMM background subtraction and Kalman filters, calibrated assuming a zero pan pinhole camera model. In (He & Yung, 2007) the calibration is based on patterns of lane markings on the road and image rectification to cope with perspective projection. A simple method using optical flow to compute displacement of pixels and relaxation of the perspective projection effect in (Lan, Li, Hu, Ran, & Wang, 2014) measured the speed inside a rectangle region of interest using known lane width as reference.

There are fewer works on fully automatic camera calibration methods. In (Dubská, Herout, & Sochor, Automatic Camera Calibration for Traffic Understanding., 2014) vanishing point detection from vehicle movement using a diamond space accumulator is performed and scale inference is computed by matching statistically detected vehicles' dimensions to mean dimensions of real vehicles. (Sochor, Juránek, & Herout, 2017) extends the previous work by matching pre-made 3D vehicle models to the detected vehicles' 3D bounding boxes. An evolutionary algorithm for camera parameter extraction is used in (Filipiak, Golenko, & Dolega, 2016), assuming constant speed of vehicles, and its accuracy is increased by license plate detection.

Traffic anomaly detetection

Methods dealing with anomaly detection in traffic videos can roughly be separated into two main categories. The first category comprises of methods that apply their models on raw image data, such as pixel location or other low level features. One recent work in this category is that of (Cheng, Chen, & Fang, 2015), using hierarchical feature representations and a Gaussian Process Regression (GPR) framework to build a low-level and a high-level codebook respectively. Anomalies are then detected, after the integration of local and global anomaly detectors. Probabilistic topic models are also proposed in a variety of works to capture spatiotemporal changes in traffic scenarios. The most typical works include the hierarchical Bayesian models of (Wang, Ma, & Grimson, 2009) which model the scene in two layers, the new Markov Clustering Topic Model of (Hospedales, Gong, & Xiang, 2012), the Probabilistic Latent Sequential Motifs introduced by (Varadarajan, Emonet, & Odobez, 2013) and the Dependent Dirichlet Process-Hidden Markov Model (DDP-HMM) framework proposed in (Kuettel, Breitenstein, Gool, & Ferrari, 2010). In all cases anomalies are determined in a probabilistic global framework. The main drawback of all these methods is their computational cost, which is usually high due to the complexity of their models. At the same time they deal with modeling at the pixel level, ignoring more complicated structures such as the objects themselves, thus missing important information.

The second category involves methods based on trajectory extraction and analysis. Objects, or even pixels are firstly localized and tracked to obtain their patterns, which are then clustered or modeled to represent the dominant underlying motions. A work in this category is that of (Saleemi, Shafique, & Shah, 2009) where object trajectories are modeled using kernel density estimations, while a unified Markov Chain Monte Carlo (MCMC) sampling-based scheme is then used to generate the most likely paths. Anomalies are detected based on the estimated probability density of the next state by comparing the actual measurements of objects with the predicted tracks. A different approach is followed in (Jiang, Yuan, Tsaftaris, & Katsaggelos, 2011), where three different levels of semantics are considered after tracking all moving objects in the video. Rules of normal events are

automatically extracted at each level and anomalies are defined as the events deviating from these rules. In (Jeong, Yoo, Yi, & Choi, 2014) a collection of trajectories is sent as an input to a two-stage inference model based on a probabilistic framework, while in (Yang, Gao, & Cao, 2013) trajectory segmentation and multiinstance learning are used for the detection of local anomalies. Finally, trajectory clustering and a single class Support Vector Machine (SVM) framework is used by (Piciarelli, Micheloni, & Foresti, 2008).

3.2 Automatic speech recognition

3.2.1 Speech recognition methodologies

According to (Jadhav & Pawar, 2012) and (Karpagavalli & Chandra, 2016), speech recognition methodologies are broadly classified into three approaches, namely, **acoustic-phonetic approach, pattern-recognition approach** and **artificial intelligence approach**.

3.2.1.1 Acoustic-Phonetic Approach

This approach is based on acoustic phonetics that postulates that there exist finite, distinctive phonetic units in spoken language. The phonetic units are characterized by a set of acoustic properties that are manifested in the speech signal, or its spectrum, over time. The first step in the acoustic phonetic approach is a spectral analysis of the speech combined with a feature detection that converts the spectral measurements to a set of features that describe the broad acoustic properties of the different phonetic units. The next step is a segmentation and labeling phase in which the speech signal is segmented into stable acoustic regions, followed by attaching one or more phonetic labels to each segmented region, resulting in a phoneme lattice characterization of the speech. The last step in this approach attempts to determine a valid word (or string of words) from the phonetic label sequences produced by the segmentation to labeling (Anusuya & Katti, 2009).

3.2.1.2 Pattern Recognition Approach

The pattern-matching approach involves two essential steps namely, pattern training and pattern comparison. In the pattern-comparison stage of the approach, a direct comparison is made between the unknown speech with each possible pattern learned in the training stage in order to determine the identity of the unknown speech.

The essential feature of this approach is that it uses a well-formulated mathematical framework and establishes consistent speech pattern representations, for reliable pattern comparison, from a set of labeled training samples via a formal training algorithm. A speech pattern representation can be in the form of a speech template or a statistical model and can be applied to a sound smaller than a word, a word, or a phrase. The pattern-matching

approach has become the predominant method for speech recognition in the last six decades (Karpagavalli & Chandra, 2016).

In ASR, there have been used both temporal pattern matching approaches, such as dynamic time warping (DTW) and Vector Quantization (VQ) and stochastic pattern matching approaches, employing hidden Markov models (HMMs) or Gaussian mixture models (GMM).

Dynamic Time Warping (DTW) is based on the observation that, in speech, the utterances[7] of the same word will have different durations. To obtain a global distance between two speech patterns a time alignment should be performed. In DTW the entire problem is divided into a small number of steps each requiring a decision to be made based on the local distance measures (Amin & Mahmood , 2008). The overall decision is made depending on these smaller decisions. To improve the accuracy of DTW a large number of templates per word is required. However, the disadvantage of using multiple templates per word is that the computational time for the calculation of DTW paths for each template increases. Thus, there exists trade-off between the recognition accuracy and the computational efficiency.

For example, a recent study (Zaharia, Segarceanu, & Cotescu, 2010) combinedDTW and Vector Quantization(which is described in the next paragraph), by using only one reference template for each class in the vector quantization method, instead of storing multiple reference templates. This reference template is called ‘centroid’. In the recognition phase, the unknown utterances are compared to the centroids. The performance of the system is evaluated on digits (0-9) in Romanian language and it is found that this technique increases the recognition speed while reducing storage space. In (Abdulla, Chow, & Sin, 2003) a technique called ‘crosswords reference templates’ (CWRT) is used to generate the reliable templates to improve the recognition accuracy. The templates are generated from a set of examples rather than a single example. The system is speaker dependent and is tested for 10 English digits. It is seen that the recognition accuracyis improved from 85.3% to 99% (Abdulla, Chow, & Sin, 2003).

Vector Quantization (VQ) is an efficient data compression technique, used in various applications such as VQ-based encoding and VQ-based recognition. A vector quantizer is a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for communication over or storage in a digital channel. The goal of such a system is data compression: to reduce the bit rate so as to minimize communication channel capacity or digital storage memory requirements while maintaining the necessary fidelity of the data.

[7] Chunks of speech between pauses, containingwords and other non-linguistic sounds, which are called fillers (breath, um, uh, cough).

In (Furui, 1991) VQ is used along with DTW/HMM. The experiments were performed on isolated utterances of 10 digits and it was found that the computation time and storage required was reduced. In (Zulfiqar, Muhammad , & Enriquez, 2009) a Vector Quantization technique is implemented through Linde–Buzo–Gray algorithm. Results show that **Mel-frequency cepstrum coefficients** (MFCC) based Speaker Identification system with VQ modeling technique has very good identification accuracy and therefore, it is robust against noise. It is seen that sampling frequency of speech and number of vectors in VQ codebook influences the identification accuracy greatly. The experiments were performed on a database having 600 voices of 30 males and 14 females.

Hidden Markov models (HMMs) are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short time-scale, speech can be approximated as a stationary process. Speech can be thought of as a Markov model for many stochastic purposes. HMMs are usually used in state of the art systems instead of DTW due to better generalization properties and lower memory requirements. HMMs provide a simple and effective framework for modelling time-varying spectral vector sequences. As a consequence, almost all present day large vocabulary continuous speech recognition systems are based on HMMs.

Automatic speech processing systems are employed more and more often in real environments. Thus, they are confronted with high noise levels and their performance degrades drastically. In (Lishuang & Zhiyan, 2010), a generic algorithm is used for training a Hidden Markov Model to improve speech recognition rate in noisy environmental conditions. Wavelet transform, MFCC and format frequencies are used for feature extraction. The experiment is performed on six different Chinese vowels at different SNR levels. The MFCC algorithm can be used along with the HMM but it cannot extract the features of speech signal at lower frequencies (Patel & Rao, 2010). When MFCC is used with frequency sub-band decomposition as feature extraction and HMM as recognizer gives better recognition results than compared to MFCC alone as feature extraction and HMM as recognizer (Patel & Rao, 2010). In (He, Deng, & Chou , 2006), the Minimum Classification Error (MCE) is used along with extended Baum Welch algorithm to optimize the HMM parameters. The MCE has faster convergence rate and is more stable than Generalized Probabilistic Descent (GPD). MCE algorithm is also well suited for large scale training.

In **Gaussian mixture models** (GMMs) each speaker has an independent GMM model. In (Ting, Salleh, Tan, & Ariff, 2007), GMMs are used for text-independent speech recognition. The database consists of Malay clean sentence speech of 10 speakers consisting of 3 females and 7 males. The model training based on highest likelihood clustering is shown to outperform the conventional Expectation Maximization training and is more

computationally efficient. In (Yantorno, Iyer, & Sha, 2004), GMM is used to distinguish between speech segments of different speakers in a multi-speaker environment.

3.2.1.3 Artificial Intelligence Approach

The Artificial Intelligence approach is a hybrid of the acoustic phonetic approach and pattern recognition approach. The main methodologies that made significant change in speech recognition area are described below.

Multi layer perceptrons (MLP) are one of many different types of existing neural networks. The idea behind neural networks stems from studies of the structure and function of the human brain. Neural networks are useful to model the behaviour of real-world phenomena. Standard back-propagation algorithm is a gradient descent algorithm, in which the network weights are moved along the negative of the gradient of the performance function. However, the performance of the network degrades in noisy environments. By using MLP in log spectral domain minimizes the difference between noisy and clean speech (Ghaemmaghami, Razzazi, Sameti, Dabbaghchian, & BabaAli, 2009). In (Paliwal K.K., 1990), the performance of MLP is tested in noisy environment and is compared with other pattern classifiers such as Maximum Likelihood classifier and k-nearest neighbour. MLP outperforms the Maximum Likelihood classifier and k-nearest neighbour at different SNR environments.

HMM-GMM combination is the most common generative learning approach in ASR. Conventional speech recognition systems utilize GMMs with HMM emmisions to represent the sequential structure of speech signals. Typically, each HMM state utilizes a mixture of Gaussian to model a spectral representation of the sound wave. The HMM state is typically associated with a sub-segment of a phone in speech (Anusuya & Katti, 2009), (Bilmes, 2006).

State-of-the-art systems use hidden markov models to achieve good levels of performance. One of the reasons for the popularity of HMMs is that they readily handle the variable length data sequences which result from variations in word sequence, speaker rate and accent. Even though the HMM-GMM approach had become the standard tool in ASR, it has its own advantages as well as disadvantages. HMMs-based speech recognition systems can be trained automatically and are simple and computationally feasible to use. However, one of the main drawbacks of Gaussian mixture models is that they are statistically inefficient for modeling data that lie on or near a non-linear manifold in the data space.

HMM-Neural Networks (Discriminative Learning). The paradigm of discriminative learning involves either using a discriminative model or applying discriminative training to a generative model. The use of neural networks in the form of Multilayer Perceptron (MLP) was popular in 1990's. Due mainly to the difficulty in learning MLPs, this line of research has been switched to a new direction where the MLP simply produces a subset of feature

vectors in combination with the traditional features for use in the generative HMM ([Morgan, et al., 2005). Neural networks, trained by back-propagation error derivatives, emerged as an attractive acoustic modeling approach for speech recognition in the late 1980s. In contrast to HMMs, neural networks make no assumptions about feature statistical properties.

Neural networks allow discriminative training in a natural and efficient manner, when used to estimate the probabilities of a speech feature segment. However, in spite of their effectiveness in classifying short-time units, such as individual phones and isolated words, neural networks are rarely successful in continuous recognition tasks (Smith & Gales, 2002), mainly because of their lack of ability to model temporal dependencies. These kind of shallow architectures have been proved effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving human speech. Thus, one alternative approach is to use neural networks as a pre-processing e.g. feature transformation, dimensionality reduction for the HMM based recognition.

HMM-Deep Neural Networks. Deep learning sometimes referred as representation learning or unsupervised feature learning, is a new area of machine learning. Deep learning is becoming a mainstream technology for speech recognition and has successfully replaced Gaussian mixtures for speech recognition and feature coding at an increasingly larger scale. The first type of deep architectures consists of generative deep architectures, which are intended to characterize the high-order correlation properties of the data or joint statistical distributions of the visible data and their associated classes. Use of Bayes rule can turn this type of architecture into a discriminative one. Examples of this type are various forms of deep auto-encoders, deep Boltzmann machine, sum-product networks, the original form of Deep Belief Network (DBN) and its extension to the factored higher-order Boltzmann machine in its bottom layer.

The second type of deep architectures are discriminative in nature, which are intended to provide discriminative power for pattern classification and to do so by characterizing the posterior distributions of class labels conditioned on the visible data. Examples include deep-structured Conditional Random Fields, tandem-MLP architecture, deep convex or stacking network and its tensor version, and detection-based ASR architecture.

In the third type, or hybrid deep architectures, the goal is the discrimination, but this is assisted with the outcomes of generative architectures. The generative component is mostly exploited to help the discrimination as the final goal of the hybrid architecture (Deng & Li, 2013), (Hinton, et al., 2012).

3.2.2 Speech Recognition Tools

Researchers on automatic speech recognition have several potential choices of open-source toolkits for building a recognition system. Notable among these are: HTK[8], Julius[9] (both written in C), Sphinx4[10] (written in Java) of the Carnegie Mellon University and Kaldi[11], a free, open-source toolkit for speech recognition research. Kaldi provides a speech recognition system based on finite-state transducers (using the freely available OpenFst), together with detailed documentation and scripts for building complete recognition systems (Povey & Ghoshal, 2011).

Some other less popular open-source systems and kits are RWTH Aachen Automatic Speech Recognition System (RASR)[12], Segmental Conditional Random Field Toolkit for Speech Recognition (SCARF)[13], Improved ATROS (iATROS)[14], SRI International's Decipher[15], idiap's Juicer and SHoUT speech recognition toolkit[16].

3.2.3 Measures of Performance

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy may be measured in terms of performance accuracy which is usually rated with word error rate (WER), whereas speed is measured with the real time factor. Other measures of accuracy include Single Word Error Rate (SWER) and Command Success Rate (CSR). The performance of the speech recognizer is measured in terms of Word Error Rate (WER) and Word Recognition Rate (WRR) (Karpagavalli & Chandra, 2016). Word errors are categorized into number of insertions, substitutions and deletions. Consequently, **Word Error Rate** is defined as follows:

$$WER = \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{No. of Reference Words}}$$

whereas **Word Recognition Rate** is defined as

$$WRR = 1 - WER$$

[8] http://htk.eng.cam.ac.uk/
[9] http://julius.osdn.jp/en_index.php
[10] https://github.com/cmusphinx/sphinx4
[11] https://github.com/kaldi-asr/kaldi
[12] https://www-i6.informatik.rwth-aachen.de/rwth-asr/
[13] https://www.microsoft.com/en-us/research/publication/scarf-a-segmental-conditional-random-field-toolkit-for-speech-recognition-2/
[14] https://www.prhlt.upv.es/wp/resource/iatros-improved-atros
[15] http://www.speech.sri.com/projects/decipher/
[16] http://shout-toolkit.sourceforge.net/

3.3 Semantic text analysis

Knowledge extraction from text, as means for populating Semantic Web knowledge bases towards the semantic integration and reasoning over the distilled semantic information, is a very challenging, interdisciplinary task that has attracted increasing interest over the past few years. Relevant approaches build on NLP pipelines for performing typical information extraction tasks, such as Named Entity recognition and classification, entity linking (EL), word sense disambiguation (WSD), and semantic parsing (i.e., the identification of semantic predicates, their arguments and their semantic roles). The outputs of the NLP tasks are then aggregated and refactored into a Semantic Web compliant representation commonly referred to as a knowledge graph.

In LODifier (Augenstein, Pado, & Rudolph, 2012), for example, Discourse Representation Structures (Kamp & Reyle, 1993) (DRSs), extracted by means of deep semantic parsing, are converted to RDF graphs using transformation rules that map the unary and binary DRS conditions to respective class and property assertions, while RDF reification is used for logical and modal descriptions, such as disjunction and possibility. Adopting a more knowledge-oriented paradigm, PIKES (Corcoglioniti, Rospocher, & Aprosio, 2016) extracts entities and complex relations between them, using deep semantic parsing and linguistic frames, and subsequently converts them into respective OWL graphs. The translation follows a neo-Davidsonian representation style, where frames are represented as reified objects, connected to each of their participants by means of properties that reflect the semantic roles of the participants, using, among others, the VerbNet and FrameNet semantic role repositories. SPARQL-like rules are used to refactor the linguistically grounded representations (so called “mention layer") to respective knowledge assertions (“instance layer"), while post-processing is applied to materialise implicit knowledge and compact redundant structures.

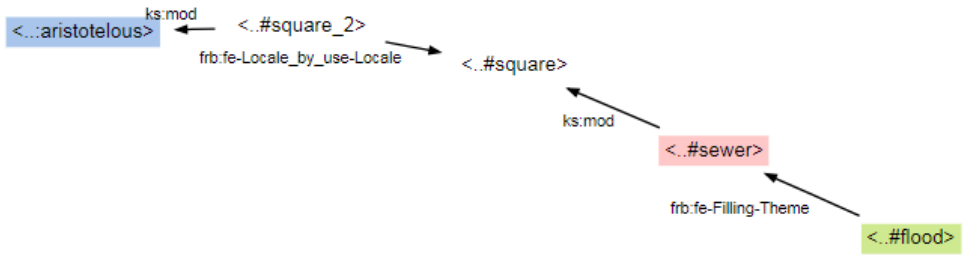

Figure 2: Visual rendering of the knowledge graph produced by PIKES for the input "The sewers in Aristotelous square are flooded."

FRED (Gangemi, Presutti, Recupero, Nuzzolese, & Mongiovì, 2017) combines Discourse Representation Theory, linguistic frames, and ontology design patterns (Gangemi & Presutti,

Ontology Design Patterns, 2009), to produce RDF/OWL ontologies and linked data from text. Deep semantic parsing is used to capture entities and the relations between them as DRS structures. Semantic role labelling is performed using VerbNet and FrameNet roles. What distinguishes FRED from other approaches and renders it as the work that is most relevant to our pursuits, is that it maximises modelling choices in accordance to Semantic Web principles and grounds the transformation and reengineering of DRS structures to RDF/OWL graphs on the event and situation semantics as defined in DOLCE+DnS Ultra Lite[17] (DUL), modelling semantic roles as object properties.

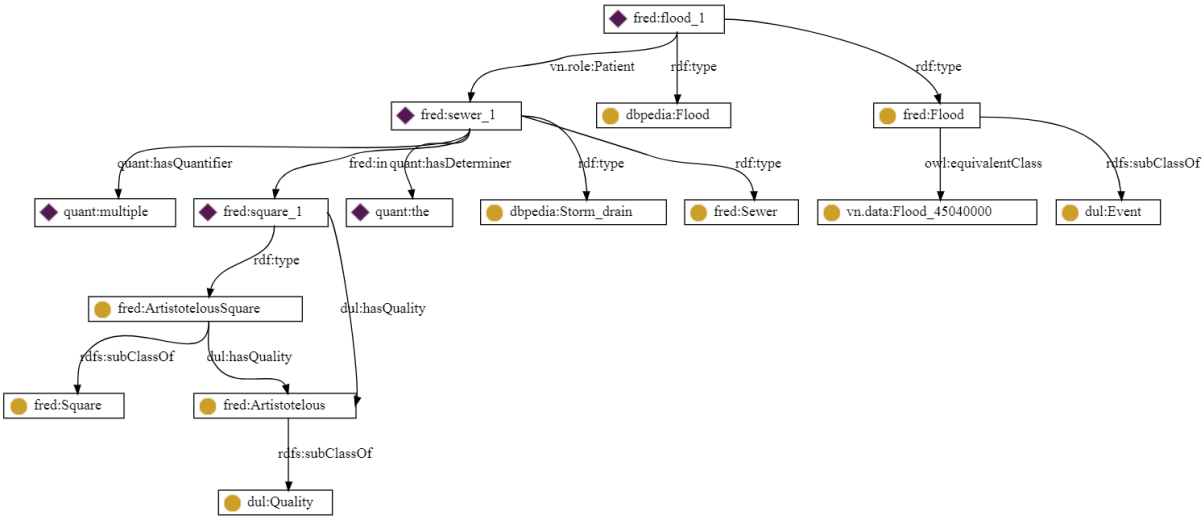

Figure 3: Visual rendering of the knowledge graph produced by FRED for the input "The sewers in Aristotelous square are flooded."

Question-answering, semantic search, summarization and semantic sentiment analysis, are only but a few examples of applications that benefit from the formalization of textual inputs semantics that such knowledge graphs afford. Event-centric graphs that capture the dynamics of the ever-increasing streams of information, as encountered in e.g., news wires, are gaining increasing popularity. In (Rospocher, et al., 2016), for example, a model for event-centric knowledge graphs is presented along with a method for large-scale extraction from news articles, while, following a different paradigm, RDF2VEC graph embeddings are used along with the subsumption hierarchy of semantic roles are used for reconciling knowledge graphs and enabling their merging based on the underlying events in (Alam, Recupero, Mongiovì, Gangemi, & Ristoski, 2017). As described in the following, social media is another application domain where event-extraction has received increasing interest and is being been extensively researched.

[17] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#

3.3.1 Semantic social media analysis for crisis management

Social media services have been increasingly acknowledged and adopted as valuable communication channels during crisis situations, ranging from natural disasters, to protests and terrorist attacks (Hughes & Leysia, 2009), (Vieweg, Hughes, Starbird, & Palen, 2010). Besides excellent information propagation channels for authorities and emergency agencies to post alerts and advices to the public, social media posts, affording near real-time streaming of information about what people experience and/or learn from others, have been shown to contribute to enhanced situational awareness and provide support for decision making tasks. As a result, over the last decade a research into social media analysis has received considerable attention, spanning a multitude of pertinent challenging tasks, from data acquisition, filtering, topic and trend detection and tracking, clustering and classification (e.g., with respect to information source, thematic content categories, etc.), to the extraction of granular information from the post contents and their geotagging; for a comprehensive survey, the reader is referred to the literature reviews of (Imran, Castillo, Diaz, & Vieweg, 2015) and (Farzindar, 2015). As within the beAWARE system, text analysis is invoked once potentially relevant to the unfolding emergency social media posts have been identified, for the remaining, we focus on the last two of the aforementioned tasks, namely geotagging and semantic information extraction.

Geotagging

Social media geotagging, i.e., attaching geo-coordinates to posts, is a typical requirement in emergency management contexts (Graham, Hale, & Gaffney, 2014), (Ikawa, Enoki, & Tatsubori, 2012), (Lingad, Karimi, & Yin, 2013), enabling, among others, location-based information retrieval, clustering and aggregation, as well as the visualization of the reported information on a map. The availability of machine-readable location information in social media messages, in the form of metadata, depends on the user's device having the capacity to know its location, on the specific client software having the capability to read this from the device, and most importantly, on the user enabling this feature explicitly. Although, in the majority of crisis-related messages, location metadata are absent (Burton, Tanner, Giraud-Carrier, West, & Barnes, 2012), many social media posts contain references to place names inline their contents (e.g. "The traffic lights in Tsimiski are out." where Tsimiski is a street in the center of Thessaloniki). Geotagging addresses the identification of place names in the text messages and their linking them to respective coordinates. This is typically accomplished using Named Entity Recognition tools to extract potential place mentions candidates, followed by a disambiguation and linking step against a reference geographical knowledge base. Gazetteer-based search and n-gram matching, pattern-matching and regular expressions, as well as supervised and semi-supervised approaches have been heavily researched for NER purposes (Nadeau & Sekine, 2009), while examples of prominent

publicly available off-the-shelf tools include, among others, Stanford NER[18], OpenNLP[19], LingPipe[20], etc.

Typical NER tools however, fall short when considering "non-traditional" NER locations, such as local streets and buildings, or when dealing with non-standard place name abbreviations and slang or misspelled references, frequently encountered in Twitter posts (MacEachren, et al., 2011), (Gelernter & Mushegian, 2011). While tweet-specific NER and linking tools have been investigated towards ameliorating the particularities of such inputs, even in very recent approaches (Inkpen, Liu, Farzindar, Kazemi, & Ghazi, 2017), the targeted scope tends to still remain within the typical NER location categories, e.g. countries, states, provinces, etc., and consider mainly English contents, which have a considerably higher coverage in relevant resources such as DBpedia and GeoNames. In an attempt towards a more comprehensive, and fine-grained, location extraction from Twitter text, a collocation-driven language model is developed from region-specific gazetteers for extracting and identifying location mentions (Al-Olimat, Thirunarayan, Shalin, & Sheth, 2017); the use of OpenStreeMap as one of the reference geographical information gazetteers, but as the evaluation considers English tweets only, it is fairly hard to assess its portability to other languages and the workload of required adaptations.

Semantic information extraction

Semantic information extraction from social media posts involves typical NLP tasks, including tokenization, lemmatization, part-of-speech tagging (POS), named entity recognition and entity linking, semantic role labeling and dependency parsing, only now, there is an explicit focus on the extraction of structured representations from tweet texts that are additionally semantically enriched via means of linking against semantic knowledge bases (i.e. knowledge sources of structure information with well-defined, formal semantics).

Prominent approaches in entity recognition and disambiguation by means of linking against Linked Data knowledge bases (DBpedia, YAGO, BabelNet, FreeBase, etc.) include among others, Agdistis (Usbeck, et al., 2014), DBpediaSpotlight (Daiber, Jakob, Hokamp, & Mendes, 2013) and Babelfy (Navigli, Camacho-Collados, & Raganato, 2017). Entities semantics are identified via de-referenceable Uniform Resource Identifiers (URIs) and Internationalized Resource Identifies (IRIs), such as, *http://dbpedia.org/resource/Rome* and *http://it.dbpedia.org/resource/Roma*. Social media-specific entity recognition and linking tools have been investigated, including among others the works by Ritter (Ritter, Clark,

[18] https://nlp.stanford.edu/ner/
[19] https://opennlp.apache.org/
[20] http://alias-i.com/lingpipe/

Mausam, & Etzioni, 2011), Bontcheva (Bontcheva, et al., 2013), Li (Li, et al., 2012) (TwiNER) and Al-Olimat (Al-Olimat, Thirunarayan, Shalin, & Sheth, 2017). However, entity recognition and linking are not the only NLP tasks for which social media-specific techniques and adaptations have been investigated. The particularities pertinent to tweets have spurred research into other NLP tasks, including POS-tagging (Owoputi, et al., 2013), (Derczynski, Ritter, Clark, & Bontcheva, 2013), as well as dependency parsing (Kong, et al., 2014), (Liu, et al., 2018) and treebanks for training. Examples of the latter include, among others, Foster et al. (Foster, et al., 2011), who annotated 7,630 tokens' worth of tweets according to the Penn Treebank (PTB) phrase-structure conventions, enabling conversion to Stanford Dependencies, and TWEEBANK (Kong, et al., 2014), the tweet Treebank developed by Kong et al. that consists of 12,149 tokens and also draws upon PTB.

Last but not least, examples of semantic social media-specific analysis systems in the domain of emergency management include the following: Twitcident (Abel, Hauff, Houben, Stronkman, & Tao, 2012), a system that supports semantic filtering, faceted search and summarization of tweets; Twitris (Purohit & Sheth, 2013) that addresses the capturing of event-related information and its semantic analysis and integration along spatio-temporal-thematic aspects, sentiment and subjectivity, as well as community evolution aspects; ArmaTweet (Tonon, Cudré-Mauroux, Blarer, Lenders, & Motik, 2017) that addresses the extraction of semantic events from tweets and their capturing as structured RDF knowledge graphs.

Despite the availability of tweet specific NLP tools and resources, their sparseness, compared to those for regular written language, and limited coverage across different languages and application domains, still favor the use of non-social media-specific tools and methodologies, especially for core tasks such as parsing, in synergy with normalization steps that precede their application. Having gone over representative approaches and tools for entity recognition and linking tasks, and locations in particular, in the remaining of the Section, we focus on resources, namely training corpora and lexicons, used for parsing, as to understand textual semantics, the relations between the words need to be established.

3.3.2 Parsing resources

The following tables compile the state of the art resources in semantic text analysis for the four targeted languages: Italian, Greek, Spanish and English. Given that UPF has previous experience working on English and Spanish corpora and lexicons, most of the following listed resources concern Italian and Greek.

Corpora of annotated sentences are needed in order to train statistical analyzers (e.g., part-of-speech taggers, lemmatizers, or syntactic parsers). For all languages, UPF will develop

Universal Dependency-based tools (see Section 6). However, in case these representations do not allow us to produce fully adequate semantic representations, UPF will resort to alternative resources, as compiled during the first half of the project (UD and best alternative are marked in green in the tables).

Table 6 describes the main characteristics of Italian corpora. With its several annotated layers, the ISST corpus is the best candidate to replace the UD corpus we use so far.

Table 6: Italian corpora

ITALIAN CORPORA				
Name	**Short description**	**format**	**size**	**license**
TUT	TUT[21] (Bosco et al., 2000) is a morpho-syntactically annotated collection of Italian sentences, which includes texts from different text genres and domains (newspapers, civil code, Acquis, Wikipedia, Italian constitution), released in several annotation formats.	TUT-native dependencies TUT-PENN constituent TUT-CCG CoNLL format Stanford Dependencies	3452 sentences, 102.000 tokens, in 6 sections	Available for download from the web: http://www.di.unito.it/~tutreeb/treebanks.html
IIST IIST-CoNLL[22] IIST-TANL	ISST (Montemagni et al., 2003) has a five-level structure covering orthographic, morpho-syntactic, syntactic and semantic levels of linguistic description. Syntactic annotation is distributed over two different levels: the constituent structure level and the dependency annotation level. The fifth level deals with lexico-semantic annotation. None of the ISST syntactic annotation levels presupposes the other. ISST has evolved into	PENN-constituents FAME functional annotation ISST semantic tags IWN senses CoNNL format (for ISST-CoNNL) Frame information (from FrameNet) for	1. a "balanced" corpus, testifying general language usage, for a total of 215,606 tokens; 2. a specialised corpus, amounting to 89,941 tokens, with texts belonging to the financial domain.	

[21] http://www.di.unito.it/~tutreeb/
[22] http://medialab.di.unipi.it/isst/ISST-CoNLL.pdf

	ISST-CoNNL and ISST-TANL	ISST-TANL (Lenci, Montemagni, Venturi, & Cutrulla, 2012)		
PAISÀ	PAISA is a (Lyding, V. et al., 2014) large collection of Italian texts annotated with morpho-syntactic tags and dependency[23] relations (also used for the annotation of the ISST-TANL dependency annotated corpus.)	The annotated corpus adheres to the standard **CoNLL** column-based format (Buchholz & Marsi, 2006) (Buchholz and Marsi, 2006), is encoded in UTF-8.	The corpus contains approximately 380,000 documents coming from 1,067 different websites, for a total of about 250 million words. All documents contained in the PAISA` corpus date back to Sept./Oct. 2010. The documents come from several web sources.	Creative Commons *Attribution-Noncommercial-ShareAlike* license. Available for download and it can be queried via its online interface: **http://www.corpusitaliano.it/en/access/advanced_interface.php** For corpus download, both the raw text version and the annotated corpus in CoNLL format are provided.
UD-Italian-ISDT[24]	The Italian corpus annotated according to the UD annotation scheme was obtained by conversion from ISDT (Italian Stanford Dependency Treebank), released for the dependency parsing shared task of Evalita-2014	CoNLL-X uses 17 UPOS tags	14,167 sentences, 278,429 tokens and 298,344 syntactic words.	License: CC BY-NC-SA 3.0

Table 7 describes the main characteristics of Greek corpora. Only the GDT is annotated with syntactic and semantic dependencies, and is then our main contingency corpus in Greek.

Table 7: Greek corpora

GREEK CORPORA				
Name	**Short description**	**format**	**size**	**license**
GDT-Greek Dependency	GDT (Prokopidis et al., 2005) is a reference corpus for Modern Greek, annotated at multiple levels:	Dependency-based annotation scheme	175.000 tokens 7000 sentences	

[23] http://universaldependencies.org/treebanks/it_isdt/index.html
[24] http://universaldependencies.org/treebanks/it_isdt/index.html

Treebank	Morphological, syntactic and semantic. The texts include: manually normalized transcripts of European parliamentary sessions, articles from the **Greek Wikipedia** and web documents pertaining the politics, health, and travel domains.			
UD_Greek -GDT	The Greek UD treebank (Prokopidis & Papageorgiou, 2017) is derived from the Greek Dependency Treebank (http://gdt.ilsp.gr), a resource developed and maintained by researchers at the Institute for Language and Speech Processing/Athena R.C.[25]	CoNLL-X	61,673 tokens 2,521 sentences	Creative Commons Attribution-NonCommercial-ShareAlike, CC BY-NC-SA 3.0.
HNC Hellenic National Corpus	HNC (Hatzigeorgiu, N. et al., 2000) is a corpus of written texts from several media (books, periodicals, newspapers etc.), which belong to different genres (articles, essays, literary works, reports, biographies etc.) and various topics (economy, medicine, leisure, art, human sciences etc.).	PAROLE format	34 million words	available over the Internet, for research use only
CGT Corpus of Greek Texts	CGT (Goutsos, 2010) is a corpus of texts from radio, television, live, book, telephone, newspaper, magazine, electronic, other Mixed corpus, including both spoken and written material		30 million words	available and freely accessible online

[25] http://www.ilsp.gr

Table 8 describes the main characteristics of Spanish corpora. As part of beAWARE, UPF develops the AnCora-UPF corpus, which should reach over 10,000 sentences by the end of the project. AnCora-UPF is naturally an annotation which will be used for the beAWARE experiments.

Table 8: Spanish corpora

SPANISH CORPORA				
Name	**Short description**	**format**	**size**	**license**
AnCora	Ancora (Taulé, Martí, & Recasens, 2008)Consists mainly of newspaper texts annotated at different levels of linguistic description: morphological (PoS and lemmas), syntactic (constituents and functions), and semantic (argument structures, thematic roles, semantic verb classes, named entities, and WordNet nominal senses). All resulting layers are independent of each other.	CoNLL	17,680 sentences ~500,000 words	freely available from the Web: http://clic.ub.edu/corpus/es/ancora
IULA Spanish Treebank	IULA Spanish Treebank (Marimon & Bel, 2015) is a technical corpus of Spanish annotated at surface syntactic level, following the dependency grammar theory	Dependency format	over 40,000 sentences	publicly and freely available from the META-SHARE platform5 with a Creative Commons Attribution 3.0 Unported License
AnCora-UPF	Following Meaning-Text Theory, MTT (Mel'čuk, 1988), the Ancora-UPF treebank (Mille & Wanner, 2010) proposes a hierarchical annotation schema that accommodates for both fine-grained language-specific dependency structures and a generic picture of abstract dependency relations.	CoNLL	3,513 sentences ~100,000 tokens	freely available from the Web: http://clic.ub.edu/corpus/es/ancora

UD-Spanish Ancora[26]	The UD-Spanish Ancora (Martínez Alonso & Zeman, 2016) was automatically converted from AnCora into UD.	CoNLL-X uses 17 UPOS tags	17,680 sentences, 547,681 tokens and 549,570 syntactic words.	GNU GPL 3.0
UD-Spanish - GSD[27]	Automatically converted	CoNLL-X uses 16 UPOS tags	16,013 sentences, 423,346 tokens and 431,587 syntactic words.	CC BY-NC-SA 3.0 US

Finally, Table 9 displays the main features of the two reference English corpora we will use:

Table 9: English Corpora

ENGLISH CORPORA				
Name	**Short description**	**format**	**size**	**license**
PennTreebank	The PennTreebank (Johansson & Nugues, 2007) is a dependency conversion of a constituency treebank, mainly containing Wall Street Journal articles	CoNLL	~40,000 sentences ~1,000,000 tokens	LDC
UD	UD (Nivre,J., et al. , 2016) is a manually revised version of open textual material from electronic journal articles, blogs, etc.	CoNLL-X	~16,000 sentences ~150,000 tokens	GNU GPL 3.0

Good quality **lexical resources** are needed in order to obtain reliable semantic structures. We aim at identifying descriptions of lexical units that include their *government patterns* (or subcategorization frames), that is, how many participants does one unit usually have and how they combine with each other. There is a great variety of lexical resources for a great variety of purposes. We focus on the resources that can be used for language analysis, but also in the context of language generation. Lexicons with more generic semantic information can be very useful, and those that include mappings to standard resources (such as BabelNet, PropBank, or VerbNet for instance) are preferred.

In the following, lexical resources relevant to our purposes are outlined. More specifically, Table 10 summarizes the main characteristics of 2 Italian lexicons; Table 11 describes the

[26]http://universaldependencies.org/treebanks/es_ancora/index.html
[27]http://universaldependencies.org/treebanks/es_gsd/index.html

main characteristics of Greek lexicons; Table 12 describes the main characteristics of Spanish lexicons; and Table 13, the main characteristics of English lexicons.

Table 10: Italian lexicons

ITALIAN LEXICONS				
Name	**Short description**	**format**	**size**	**license**
PAROLE-SIMPLE-CLIPS	PAROLE-SIMPLE-CLIPS (Ruimy & al., 2002) is a multilayered lexicon (4 layers) Provides interconnected syntactic and semantic information. Includes argument structure.	Based on the PAROLE-SIMPLE model.	~55,000 lemmas	Non-Commercial Use - ELRA END USER
RDF-converted PAROLE SIMPLE CLIPS	RDF-converted PAROLE SIMPLE CLIPS (Del Gratta, 2015) is a conversion of PSC into RDF. Linking to the semantic web and Linked Data cloud. Follows the qualia structure of the generative lexicon theory and the lemon view of lexical sense as a reified pairing of a lexical item and a concept in an ontology.	RDF		Open Data Commons Attribution License

Table 11: Greek lexicons

GREEK LEXICONS				
Name	**Short description**	**format**	**size**	**license**
LEXIS[28] GDT-LEXIS LEXIS-EmotionVerbs	A Greek Computational Lexicon of general language based on corpora, language with **morphological, syntactic and semantic** information. GDT-LEXIS (Papageorgiou & al., 2006) s a lexical resource with semantic information for verbal predicates. LEXIS-Emotion Verbs (Giouli & Fotopoulou, 2012) details the argument structure, distributional properties and possible transformations of greek emotion verbs.		Comprises ~60,000 entries with morphological information, of which a subset of 30,000 entries also have syntactic information and a further subset of 15,000 with semantic information. In GDT-LEXIS: about 800 verbs	

[28] http://www.ilsp.gr/en/infoprojects/meta?view=project&task=show&id=140

SKEL	SKEL (Petasis & al., 2001) is a morphological lexicon that was used to develop a lemmatizer and a morphological analyser that were included in a controlled language checker for Greek.		~60.000 lemmas that correspond to ~710.000 different word forms.	
Conceptual Lexicon	ConceptualLexicon (Fotopoulou & al., 2014) encodes morpho-syntactic and semantic properties of nominal and verbal MWEs.		~1000 entries	
EKFRASI	EKFRASI (Tzortzi & Markantonatou, 2014) is a conceptually organised lexicon encoded with Protégé, Includes conceptual and lexical relations as well as their morpho-syntactic properties			

Table 12: Spanish lexicons

SPANISH LEXICONS				
Name	**Short description**	**format**	**size**	**license**
AnCora_Verb_ES	AnCora_Verb_ES (Aparicio, Taulé, & Martí, 2008) provides semantic info, subcategorization, Argumental patterns and thematic roles. Pbank id, Verbnet Id, Framenet id, Wordnet id	XML	2,820 verbs	Freely available
AnCora_Nom_ES	AnCora_Nom_ES (Peris & Taulé, 2011) covers deverbal nouns: Denotative type, Wordnet synset, argumental pattern and thematic roles. Link to verb.	XML	1,658 lemmas	Freely available
AnCora-Net	AnCora-Net (Taulé & al., 2011) contains the AnCora-Verb lexical entries linked to different English knowledge source: VerbNet, PropBank, FrameNet, WordNet 3.0 and OntoNotes.	XML		Freely available
ADESSE	ADESSE (García-Miguel & al., 2010) covers subcategorization frames, diathesis alternations and syntactic semantic schemes.		~4,000 verbs	

GLiCOm[29]	Computational lexicon of inflected wordforms in Spanish. The lexicon is distributed in two sublexicons: 1. word forms 2. verb-clitic combinations		1,152,242 word forms and 4,283,637 verb-clitic combinations	Freely available

Table 13: English lexicons

ENGLISH LEXICONS				
Name	**Short description**	**format**	**size**	**license**
PropBank & NomBank	PropBank (Kingsbury & Palmer, 2002) and NonBank (Meyers, MacLeod, Szekely, Zelinska, & Young, 2004) cover subcategorization frames for verbs and nouns respectively, and correspondences between syntactic and semantic roles	XML	11,781 disambiguated lemmas	CC BY-SA 4.0
VerbNet	VerbNet (Schuler, 2005) provides a classification of verbs into 270 semantic classes; Subcategorization frames, diathesis alternations and syntactic semantic schemes.	XML	2,380 disambiguated verbs	CC BY-SA 4.0

Table 14 gives details about BabelNet.

Table 14: Multilingual lexicons

MULTILINGUAL LEXICON				
Name	**Short description**	**format**	**size**	**license**
BabelNet	BabelNet (Navigli & Ponzetto, BabelNet: Building a very large multilingual semantic network, 2010) is a multi-lingual semantic network with fine-grained senses, definitions and mappings to VerbNet among others	RDF / HTTP API	284 languages ~6,000,000 concepts 10,000,000 named entities	CC BY-NC-ND 4.0

A very large amount of NLP tools has been developed in the recent years; most tools are language-agnostic and simply need to be trained on the resources of a desired language. One of the most widely used toolkit is Stanford CoreNLP (Manning, Surdeanu, & Baue,

[29] https://www.upf.edu/documents/107805982/109136461/tec0128_glicom_tbadia.pdf/07632628-f275-425e-b59c-417433c6a327

2014), which contains all the basic components needed in an NLP analysis pipeline: sentence splitting, tokenization, lemmatization, morphological tagging, coreference resolution, dependency parsing. Another popular toolkit is MATE Tools (Bohnet & Nivre, 2012), developed at the University of Stuttgart. As described in Section 6, we currently use components of these two off-the-shelf toolkits, which we trained for our purposes. For dependency parsing, we use a different tool, namely the MST parser, which (McDonald, Lerman, & Pereira, 2006) has the advantage of having low memory requirements at execution time compared to other parsers, with a competitive accuracy.

UPF has recently been working on English and Spanish, but has little experience with Italian or Greek. Table 15 and Table 16 outline some alternative NLP tools for the latter two languages.

Table 15: NLP tools for Italian

ITALIAN NLP TOOLS	
Name	**Short description**
TULE	A linguistic framework including a Morphological analyzer, a Tokenizer and a Chunk-rule based dependency Parser in Allegro Common Lisp. Supports two languages: Italian and English. TULE adopts a representation format based on the dependency paradigm centred upon Augmented Relational Structure (ARS), where each relation is implemented as a feature structure that can include values for morpho-syntactic, functional-syntactic, and syntactic-semantic components. **http://www.tule.di.unito.it/index.html**
LINGUA	Includes: sentence-splitter, tokenizer, pos-tagging, lemmatization and dependency parser. **http://linguistic-annotation-tool.italianlp.it/**
TINT	TINT (Palmero & Moretti, 2016) is a Java-based pipeline for Natural Language Processing (NLP) in Italian, based on Stanford CoreNLP. Includes: tokenization, sentence splitting, morphological analysis, lemmatization and modules for part-of-speech tagging, dependency parsing and named entity recognition. **http://tint.fbk.eu/index.html**
TANL	Text Analytics and Natural Language software including: sentence-splitter, tokenizer, pos-tagger, lemmatizer, morph-tagger, NE-tagger, anaphora resolution, super-sense tagger and parser **http://medialab.di.unipi.it/wiki/Tanl**

Table 16: NLP tools for Greek

GREEK NLP TOOLS	
Name	**Short description**
ILSP	Natural Language Processing services developed by the NLP group of the Institute for Language and Speech Processing: chunker, dependency parser, FBT pos-tagger, lemmatizer, named-entity recognizer, sentence splitter and tokenizer, transliterator and Wikipedia multilingual domain-related terma and URL lists extractor **http://nlp.ilsp.gr/ws/**
AUEB	NLP software developed by the Natural Language Processing Research group in the Dept.Informatics of the Athens University: pos-tagger, named entity recognizer and NaturalOWL generator for Greek and English. http://nlp.cs.aueb.gr/software.html
ELTL	NLP tools: lemmatizer, pos-tagger, grammatical tagger, VerbTagGr and link to WordNet **http://hermes.di.uoa.gr/glosseng.htm**
LEXISCOPE	A compound language tool that provides information about a Modern Greek word or phrase, combining the functionality of Neurolingo's Hyphenator, Speller, Lemmatizer, Morphological Lexicon and Thesaurus. **http://www.neurolingo.gr/en/online_tools/lexiscope.htm**

3.4 Summary

Summarize, outlining approaches/results upon which the beAWARE analysis components build upon and respective limitations (that the beAWARE analysis approaches envisage to mitigate).

4 Image and video analysis v1

4.1 Fire and flood detection in social media images

For the task related to the analysis of fire and flood images from social media the image analysis component includes several interoperable modularities deploying an array of cutting-edge computer vision techniques:

a) **Image classification** so as to determine whether an image contains an emergent event or not (i.e. a fire of flood event),
b) **Emergency localization** in order to detect the regions where fire and flood pixels exist in flood and fire pictures,
c) **Object Detection** so as to find people and vehicles that exist in the image.

Each one of them is assigned to process an image separately from the others in order to decide about the existence of fire and flood concepts and objects that are of particular interest like people and vehicles and later locate their position inside the image. Then, a severity level estimation module is assigned with the task of deciding about the danger that the people and vehicles undergo based on their proximity to the emergent event. The overall diagram is depicted in Figure 4:.

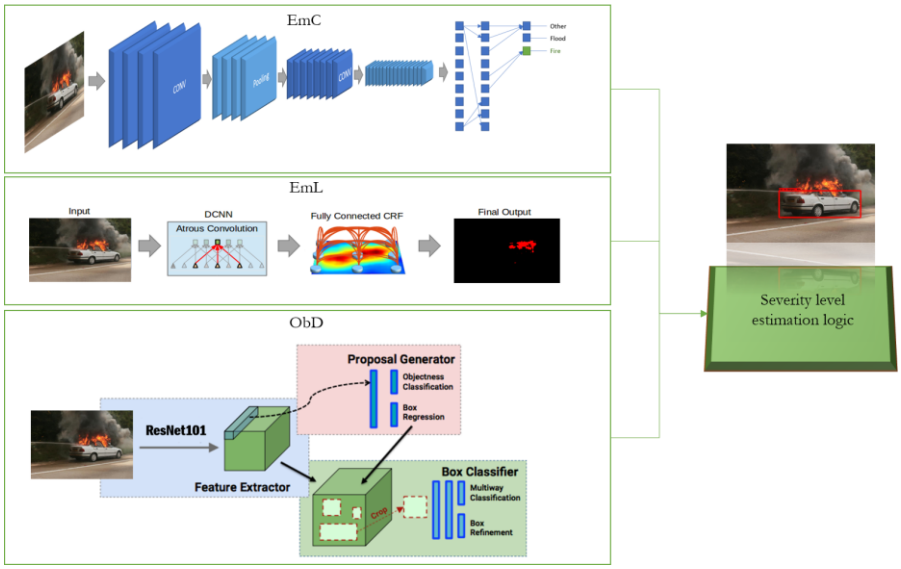

Figure 4: Overall diagram of fire and flood social media images analysis

4.1.1 Emergency Classification (EmC)

The Emergency Classification component is based on State-of-the-Art image classification techniques and is used so as to determine which images contain an emergency event. Inspired from the recent success that deep learning showed in image understanding (Simonyan & Zisserman, 2014) and scene recognition (Zhou, Lapedriza, Xiao, Torralba, &

Oliva, 2014), fine-tuning of the pre-trained parameters of the ***VGG-16*** on ***Places365*** dataset was performed so as to leverage useful distinctions between various visual clues that relate to generic scenery images. A set of amendments were performed on its architecture so as to fit it to our purposes: Initially the final Fully Connected (FC) layer was removed and replaced with a new FC layer with a width of 3 nodes freezing the weights up to the previous layer and finally a softmax classifier was deployed so as to enable multi-class recognition. More specifically the EmC results into three-class image recognition: "Fire", "Flood" and "Other", where "Other" may represent any theme except for fire and flood events, e.g. scenes of interior, forests, snowy mountains, crowded streets, urban life etc.

The EmC results are integrated in the framework to indicate the existence of fire and flood events in a holistic manner and the component's purpose is to give an early indication and a first segment of solid information about the existence of an emergency in the image. This information, taken from an initial observation of the whole image, is rather useful to be integrated into the severity level analysis.

4.1.2 Emergency localization (EmL)

Simultaneously with EmC, an Emergency Localization (EmL) component is deployed, which is responsible to semantically segment the regions where fire and flood pixels exist in case EmC's result indicates an emergency situation. Inspired from the recent success that semantic image segmentation achieved by (Chen, Papandreou, Kokkinos, Murphy, & Yuille, 2016), the DeepLab architecture of "atrous convolution", which uses convolution with up-sampled filters was adopted so as to solve the emergency localization problem in images. Atrous convolution allows a wider reception field of the convolution filters, leading to richer context representations, while it also combines the result feature vectors of the final convolutional layer with a fully connected Conditional Random Field (CRF) which provides refined segmentation masks as it includes neighboring context on its calculations.

4.1.3 Object Detection (ObD)

The Object Detector (ObD) component is responsible to provide a set of bounding boxes of the persons and vehicles in social media images, as well as their immediate surroundings. Groups of people or individuals are detected as persons, while vehicles may contain one of the following categories: cars, trucks, buses, bicycles and motorcycles. The basis of our object detection component is inspired from Faster R-CNN (Ren, He, Girshick, & Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, 2015), pretrained on COCO dataset (Lin, et al., 2014), with some alterations so as to make it fit to our emergency event purposes. More specifically, based on (Huang, et al., 2017), the ***ResNet101*** feature extractor was deployed for the extraction of deep features and then an Region of Interest (RoI) pooling scheme was used to classify candidate boxes. The model was trained in

COCO dataset and only the relevant object classes were kept as valid preditions (e.g. vehicles, people). Vital information is expected to be extracted about the existence of important targets and their locations in the image that cannot be missed if a successful warning system is to be created. The accurate prediction of the targets' location also matters greatly. This piece of information that is given in the form of bounding box coordinates, encloses not only the silhouette form of a target but also it's immediate surroundings. Accurate correlation of fire or flood localization and target location wouldn't be possible in many cases, if an exact segmentation of a target's form was to be used.

4.1.4 Severity level estimation

The framework is completed with the severity level estimation component which combines EmC, EmL and ObD results so as to define a severity level label for each in-danger bounding box of the gathered social images: (a) 'Safe target', (b) 'Target possibly in danger', (c) 'Target in danger', which can also be interpreted as a qualitative risk assessment scale of three levels: 'Low', 'Medium', or 'High' respectively.

The possible outcomes of the system logic are described here:

a) Low risk for an emergency event we have when the EmC classifies the candidate image as 'other'. All the detected bounding boxes from the ObjD are declared 'Safe targets'.
b) Medium risk we have when an image is classified as emergency from the EmC component (i.e. fire/flood). All targets that are detected from ObD are automatically characterized as 'Possibly in danger'.
c) Elevation to high risk we have when a bounding box detected by ObD coincides with EmL emergency masks (i.e. fire/flood).

After 10-fold cross-validation tests, it was concluded that a 60% confidence for object detection score provided a great balance between accuracy detection and severity level estimation. More specifically this threshold may result in multiple overlapping boxes for the same objects, a behavior that not only cannot harm the warning system but also in some samples could catch cases where only a portion of the object can be seen (i.e. a car half occluded from water or a fire-fighter fighting with flames).

4.2 Fire and flood detection in video samples

4.2.1 Spatio-temporal Representation of Dynamic Textures

In order to effectively deal with the challenging nature of videos containing outdoors unconstrained environments, their representation should be firstly carefully examined and determined. The stochastic movements of the ensembles comprising dynamic textures in

combination with their non-rigid nature, require the adoption of general descriptors, capable of managing highly unpredictable and ambiguous types of videos. To this end, the LBP-flow descriptor is adopted, which is then encoded by Fisher vectors resulting in an informative mid-level descriptor. The process is shown to be able to accurately classify dynamic scenes whose complex motion patterns are difficult to separate otherwise.

LBP-flow

The descriptor LBP-flow introduced in (Avgerinakis, et al., LBP-flow and hybrid encoding for real-time water and fire classification, 2017) was adapted and further investigated in order to accurately describe videos' underlying structure, as it has proven to effectively encode both appearance and motion induced variations, present in dynamic textures. LBP-flow constitutes an extension of the well-known LBP (Wang & He, 1990), which was chosen due to its successful applicability in a variety of texture classification tasks (Liu, Zhao, Long, Kuang, & Fieguth, 2012), (Qian, Hua, Chen, & Ke, 2011), (Zhao, Ahonen, Matas, & Pietikainen, 2012), (Zhao & Pietikainen, 2006) and face recognition tasks (Shan, Gong, & McOwan, 2009), (Ahonen, Hadid, & Pietikainen, 2006). Thus, inspired by its success, LBP-flow builds upon the original LBP and extends it over time providing a powerful shallow spatio-temporal descriptor. In classic LBP, the LBP value of a particular pixel is computed by comparing its intensity value with that of its neighboring pixels. LBP-flow extends this definition to also include the values of the optical flow around the pixel, so as to embed motion information. The representation of motion as a temporal texture is introduced by calculating LBP over the optical flow values in the x and y directions, x - t and y - t respectively. This inclusion of motion information in the LBP-flow representation enriches the descriptor's spatio-temporal characteristics leading to a more robust and efficient shallow representation.

Fisher Encoding

LBP-flow includes rich spatiotemporal information as a low-level local representation, but also allows for redundancies, such as intra-class pattern deviations and noise-induced artifacts. In order to constrain this noise and subsequently increase the discriminative ability of our descriptor, the Fisher Vector representation is adopted, transforming initial LBP-flow vectors of each video sample into a mid-level single vector representation, based on the detected most discriminating features (visual vocabulary) of a training video database. In this way, the size of the descriptor is significantly reduced, while at the same time recognition accuracy is increased. The computation of the most discriminating samples is performed by applying unsupervised clustering (Gaussian Mixture Model (GMM)) in the shallow representation hyperspace, as formed by the LBP-flow feature collection of the dynamic texture dataset.

4.2.2 Dynamic Texture Recognition and Localization

Given the aforementioned powerful descriptor, a framework for dynamic texture recognition and localization was built. Fisher vectors are either used to train a binary/multi-class Support Vector Machine (SVM) classifier or a Neural Network (NN), in order to learn to discriminate between two or more classes. The framework including the NN can be characterized as a hybrid representation scheme, as it leverages both shallow and deep parameters to train a final classification model. Dynamic texture localization follows, to spatio-temporally localize the selected dynamic texture inside, and throughout, sequential video samples. The scheme exploits the resulting binary model of the aforementioned recognition process and based on a superpixel clustering procedure leads to an accurate and computationally efficient localization framework.

Dynamic texture recognition

Dynamic texture recognition requires an accurate sampling process, so as to collect a sufficient number of informative feature samples to train the discriminative model. Activity Areas (AA) (Avgerinakis, Briassouli, & Kompatsiaris, Activity detection using Sequential Statistical Boundary Detection (SSBD), 2016) are used as an initial step to detect regions of interest and to sample interest points in them to be used for training purposes. AA are binary masks, extracted according to the premise that ow estimates originate either from actual motion, or noise, e.g. from the video capture or compression process. LBP-flow is then calculated over a block of 32X32 pixels, around each interest point. Subsequently, Fisher encoding is deployed after a Principal Component Analysis (PCA) dimensionality reduction step and the total set of descriptors representing the whole training corpus are led as an input into a Neural Network (NN) scheme. As inspired from the successful results presented in (de Souza, Gaidon, Vig, & López, 2016) the architecture consists of three hidden layers, each of which is followed by a dimensionality reduction step. The statistical power that Fisher vectors encapsulate in their scheme, passes in the NN as well and leads to a highly discriminative vector. The block diagram of the texture recognition framework is depicted in Figure 5.

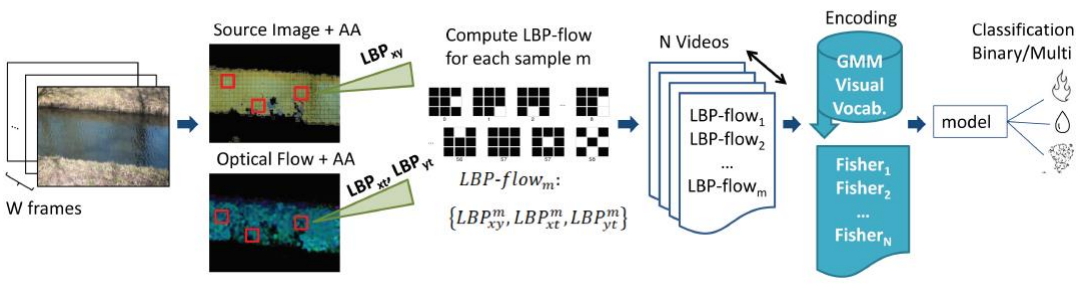

Figure 5: Block diagram of the texture recognition framework.

Dynamic texture localization

Although the aforementioned descriptor can effectively capture scene dynamics for video classification, the adoption of a local approach is needed in order to achieve accurate localization of a dynamic texture within a video frame. For this purpose, a multi-scale superpixel scheme was implemented, as superpixels enable the grouping of pixels into regions with a homogeneous appearance, which are highly likely to correspond to the same object. Furthermore, this process also eliminates redundant image information, leading to the extraction of more accurate object contours. Superpixels extracted according to the Simple Linear Iterative Clustering (SLIC) method in (Achanta, et al., 2012), are used to segment the video frames. SLIC is based on a local version of K-means algorithm, where the only parameter that needs to be specified is the number of approximately equally sized superpixels. Then, an iterative 2-step process begins with each pixel being assigned to its nearest cluster center followed by the computation of the residual error between the new cluster center and previous cluster center locations as derived from L2-norm. This process is repeated until convergence. The distance measure used for the clustering is based on pixels' color and location. Finally, a post-processing step to cluster some remaining individual "orphaned" pixels takes place by using a connected components algorithm.

Superpixels are then deployed in a 2-layers scheme, with each layer corresponding to a different scale, following a fine to coarse structure. This way, both coarse and fine details are successfully captured, and the influence of local noise is avoided. Next, superpixel clustering is carried out, which relates each superpixel in the top coarser layer with multiple superpixels of the finer bottom layer according to the overlap they have with each other, and a final descriptor characterizing the whole area covered by the superpixel of the top layer is extracted.

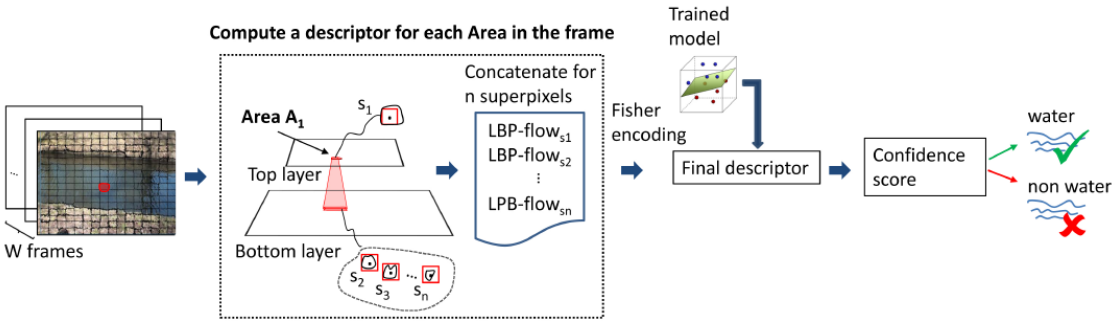

Figure 6: Block diagram of the overall localization framework.

After the extraction of area's descriptor, the discriminative models that have been trained in the aforementioned binary classification task, are used in order to localize the desired dynamic texture in a spatio-temporal manner. The decision is conducted locally for each

area covered from superpixels of the top layer. The complete localization scheme is depicted in Figure 6.

4.3 Traffic analysis and management

4.3.1 Traffic flow analysis

A robust algorithm for detection and tracking of moving vehicles from surveillance camera footage operates at the basis of the traffic flow analysis system. Passing vehicles have to be successfully detected in each frame and then subsequently tracked as they follow their full trajectory on screen. The task is simultaneously performed in a cooperative manner by two separate modalities: (a) a generic object detector, specifically trained to discover bounding boxes of vehicles at an adjustable detection rate, and (b) a tracker which accepts new image patches as input queries and is assigned to discover the most probable position of each query in subsequent frames. This finally leads to the incremental construction of the movement trajectories for each individual vehicle that has been captured by the detector at some point in the frame sequence.

To accurately compute the velocities of a detected vehicle, known pixel coordinates of the corresponding trajectory in the image plane have to be back-projected in real world coordinates in the road plane. Then, given a vehicle's traveled distance in meters and the time duration in seconds, velocities can be calculated. The frame per second capture ratio of the recorded videos is known and can be used to compute time intervals in seconds between consecutive frames. What is not directly available however, is a way to translate displacement of pixels in the image plane to the real distance in meters a particular vehicle has traveled in a given amount of time. In order to calculate this precisely and effectively, an algorithm for automatic camera calibration is needed. Once camera parameters have been discovered and a few basic assumptions tailored to this specific application have been integrated, pixel coordinates in the image plane can be successfully back-projected to 3D world coordinates in the road plane and therefore real vehicle displacements can be measured.

Camera calibration

In order to automatically obtain camera parameters of a traffic surveillance scene, the algorithm proposed by (Dubská, Herout, & Sochor, Automatic Camera Calibration for Traffic Understanding., 2014) was deployed which is based on detection of two vanishing points. As examined in (Sochor, Juránek, & Herout, 2017), knowledge of two vanishing points is enough to calculate camera intrinsic parameters. Moreover, the third vanishing point position can be easily found by application of orthogonality. The model makes some basic assumptions for static cameras, zero pixel skew, square shaped pixels and location of the principal point in

the center of the image that produce tolerable errors. By aiming to recover 3D coordinates belonging to vehicle trajectories, the model is used to retrieve points lying on the road plane, and is not applied to find arbitrary 3D locations in the video frames.

The method resorts to vehicle motion analysis in the scene as a means of retrieving the first vanishing point whose direction is parallel to the road and coincides with the stream of traffic. The detection algorithm uses the Hough transform on successfully tracked trajectory points based on parallel coordinates, mapping the projective plane onto a finite space, the so-called diamond space, as detailed in (Dubská, Herout, Juránek, & Sochor, 2015). In order to detect good features to track, background subtraction is performed to limit the candidates to possibly only vehicle edges. Then, using the KLT tracker, features that are correlated with significant movement are interpreted as small straight fragments of valid trajectories and are allowed to vote in the diamond space accumulator. Based on the highest number of votes, the first vanishing point coordinates are retrieved. To discover the location of the second vanishing point, which is perpendicular to the first and parallel to the road plane, the diamond space accumulator is used again in the same manner but with the following constrains: edges supporting the first are excluded this time and an assumption of approximately horizontal scene horizon filters out nearly vertical edges. Again, the point with the most votes is selected as the second vanishing point.

Figure 7: Illustration of three orthogonal vanishing points detected

Once the first two vanishing points have been found, the third vanishing point, the focal length and the road plane normal vector which defines the road plane up to a scale are calculated. Subsequently, back-projection of a 3D image plane point onto the road plane is possible following the procedure described in (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001). However, the distance of the road plane to the camera center is

calculated only up to a scale. This means that any distance that is calculated from 3D points is not expressed directly in real units of distance. To overcome this back-projection of two 3D points in the scene with known distance in meters (or other unit) can be applied and then solving for the scale factor is possible. Another method for scale inference would be to fit pre-made 3D car models in 3D bounding boxes in order to make the algorithm fully automatic with no required user input. Figure 7:illustrates the three vanishing point orientations as detected from the algorithm, as well as the horizon line.

Vehicle detection and tracking

For the purpose of detecting vehicles in video frames a deep CNN object detector was used that extracts deep image representations from a CNN and predicts pixel coordinates of bounding boxes. Similarly to 3.1.1 the aforementioned object detection architecture was adopted here.

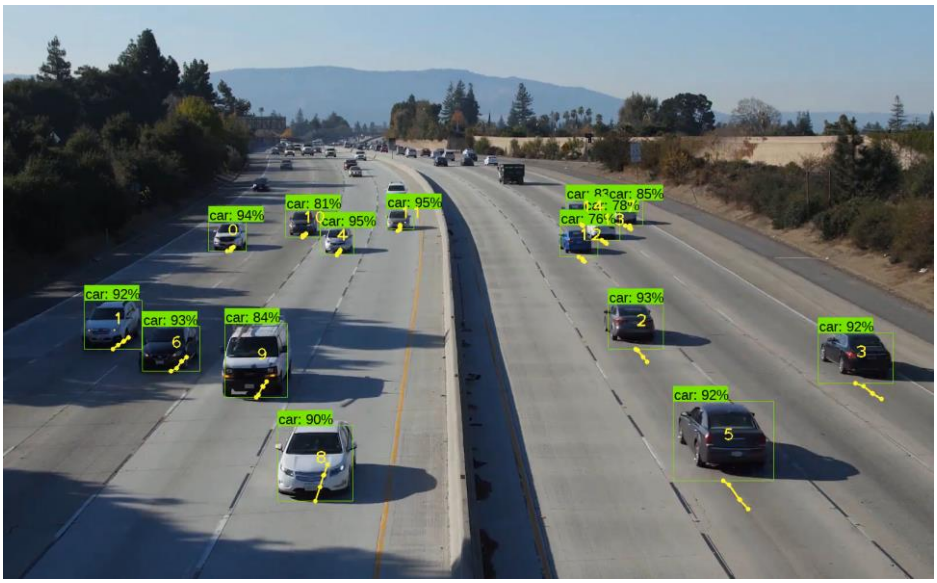

Figure 8: Visualization of detected vehicles and their trajectories.

The core functionality of the tracker was based on the KCF tracking algorithm that was proposed in (Henriques, Caseiro, Martins, & Batista, 2015). The vehicle detector is used initially in order to detect vehicles every *r* video frames and initialize the new vehicle candidate database with new entries. Bounding box coordinates are stored over time so that full trajectories can be build. For every new ID its corresponding class label and a detection score is saved as well. Immediately after, the algorithm checks the new detections from the candidate pool for overlaps with already existing recent trajectories. Then, based on an IoU score check it rejects found boxes that exceed an overlap threshold to avoid creating multiple identities for the same vehicle. Next, the KCF tracker is fed with the remaining boxes in order to localize their position throughout sequential video frames. Future

detections of already tracked vehicles are also utilized in order to rectify the bounding boxes of the monitored vehicles. When a detection is missed, the bounding box is relocalized relying only on KCF update coordinates, while when the algorithm does not localize any tracked vehicle for several sequential video frames the vehicle is presumed to have traveled off the frame. To tackle overlaps between True Positive (TP) cases, which translate to when a correctly tracked vehicle passes in front of another confusing the tracker, the trajectories are merged at the current frame and the oldest ID is assigned to the resulting trajectory. Figure 8 depicts bounding boxes and trajectories of vehicles successfully detected and tracked using this methodology.

Velocity estimation

To estimate the velocity of a tracked vehicle at a certain frame the KLT tracker is fed with points inside the previous box instance of the previous frame and several displacement calculations are produced for each point. Then, back-projection of all the displacement pixel pairs is applied to the road plane according to the calibration parameters that have been found on that specific scene and the median displacement is selected as the true value. To calculate the velocity in meters per second the true distance is divided with the time duration of a frame which is equal to 1/fps seconds.

4.3.2 Anomaly detection in traffic scenes

In order to deal with the challenging nature of traffic videos derived from real surveillance systems, the scheme should exhibit features such as generality, scalability, independence in external conditions (e.g. illumination changes, camera motion etc) and also simplicity for computational reasons. To this end, an algorithm based on the object detection described in the previous section is proposed.

Initially an early descriptor is formed in a pre-defined time window concerning each object in the scene consists of the concatenation of all the values describing object's speed and position in a specific spatiotemporal volume. Subsequently, all early descriptors extracted from a particular video sequence are led into a Fisher encoding scheme. This way, a visual vocabulary based on the most discriminating features of the whole video is built, and a more efficient representation is provided. Next, the computation of the most discriminating samples is performed by applying unsupervised clustering (Gaussian Mixture Model (GMM)) in the shallow representation hyperspace, as formed by the feature collection of each video. Next, fisher encoding follows based on the created GMM vocabulary in order to efficiently capture the whole frame's dynamics.

Finally, in order to infer about anomalous trajectories, the Support Vector Method For Novelty Detection of (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001) is adopted. Support Vector Machines (SVMs) are chosen, as they generally exhibit good performance

relatively to other machine learning methods at a low computational cost, while at the same time they are able to handle large data sets, which generally appear in real life situations.

5 Audio analysis v1

The current section describes the implementation and integration of the ASR module. After studing state-of-the-art ASR approaches and after a thorough review of existing ASR tools (ie. Kaldi, CMU Sphinx, RWTH ASR, VoxForge, LumenVox etc.), investigating technical characteristics, such as supported platforms, language models, type of license, development capabilities etc., CMU Sphinx[30] was selected as the base for Speech Recognition in the framework of beAWARE. The choice of CMU Sphinx was based on a number of advantages, such as open source license, large number of publications[31], support for both Windows and Linux, available models for all the targeted languages etc. Specifically, Sphinx4 library is used, which is a pure Java speech recognition library. It provides a quick and easy API to convert the speech recordings into text with the help of CMU Sphinx acoustic models.

5.1 Recognition process

Speech Recognition is performed by taking an audio waveform, splitting it at utterances[32] by using silences and then trying to recognize what is being said in each utterance, by matching all possible combinations of words with the audio. The extraction of the best matching combination is based on three entities:

An **acoustic model,** which contains acoustic properties for each basic speech segment.

A phonetic **dictionary,** which contains a mapping from phones (speech segments) to words. A dictionary can contain alternative pronunciations for the same word.

A **language model,** which is used to restrict word search. Language models contain statistics of word sequences and define which word could follow previously recognized words. They help to significantly restrict the matching process by stripping words that are not probable.

It should be mentioned here that, CMU Sphinx uses Hidden Markov models (HMMs) to represent acoustic model states. HMMs are an elegant generalization that leads to more robust performance and they are the most common framework for acoustic models in modern speech recognition systems. HMMs model speech by breaking it into short “consistent” segments that are relatively uniform within themselves. E.g. a speech signal for “SAM” can be broken down into three such segments, namely: ‘S’, ‘A’, ‘M’. Each segment is modeled by an HMM state. Each state has an underlying probability distribution that

[30] https://cmusphinx.github.io/
[31] https://cmusphinx.github.io/wiki/research/
[32] Chunks of speech between pauses, containingwords and other non-linguistic sounds, which are called fillers (breath, um, uh, cough).

describes the feature vectors for its segment. The entire word can be described as a linear sequence of such states.

The aforementioned three entities (language model, acoustic model and dictionary) are combined together in an engine to recognize speech (Transcriber). Sphinx4 is designed in order to be language independent, which means that the transcriber is able to perform Speech Recognition without modifications for different languages, by carefully designing these three entities in order to capture the characteristics of the language of interest. For many languages there are acoustic models, phonetic dictionaries and even large vocabulary language models available for download, however the availability resources and the vocabulary coverage for Greek and Italian is limited compared to English and Spanish. This means that Greek and Italian models need more expansion and adaptation.

In order to use CMU Sphinx4 in a java project, we have to add the Sphinx4 libraries (namely sphinx4-core and sphinx4-data) to the dependencies of the project. Sphinx4 is available as a maven package in the Sonatype OSS repository, which should also be included in the project's repositories.

In order to perform recognition a high-level recognition interface is used, called **StreamSpeechRecognizer**, by setting four attributes: a) acoustic model, b) dictionary, c) language model and d) source of speech. The first three attributes are the previously described models, which are either existing models available online or models adapted by the user. They are defined by using a Configuration object, which is then passed to the recognizer. The source of speech is the audio file that will be analyzed. The audio format for the decoding must have the following formats:

- RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 16000 Hz or
- RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 8000 Hz.

Especially, the sampling frequency depends on the acoustic model that is being used and the speech corpus that was used for the training of this model. Currently, all acoustic models are trained on 16kHz. Thus, input audio files should have Fs=16kHz. However, in order to avoid possible errors, in case an audio file of different format is passed as input to ASR, we have added a pre-processing step, by integrating an audio encoder able to convert different audio formats into the appropriate format. The java library that is used in this step is **Jave**[33].

An additional preprocessing step is the inclusion of a deniosing algorithm based on Power-Normalized Cepstral Coefficients.

[33] http://www.sauronsoftware.it/projects/jave/

After StreamSpeechRecognizer finishes the analysis of the audio, another class called **SpeechResult** can be used to provide access to various parts of the recognition result, such as the recognized utterance, a list of words with timestamps, the recognition lattice, etc.

In order to assist MTA in the semantic analysis of the trasncriptions, we developed a simple post-processing step in order to split text into potential sentences, by using the time durations of the silences between words. In order to make Automatic Punctuation step more robust, future work could be based on Dynamic Sentence Length features (Ueffing, Bisani, & Vozila, 2013).

5.2 Extending the phonetic dictionary

A phonetic dictionary provides the system with a mapping of vocabulary words to sequences of phonemes. It might look like this:

Hello H EH L OW

World W ER L D

A dictionary should contain all the words we are interested in, otherwise the recognizer will not be able to recognize them. However, it is not sufficient to have the words in the dictionary. The recognizer looks for a word in both the dictionary and the language model. Without the language model, a word will not be recognized, even if it is present in the dictionary.

In order to start with, in beAWARE ASR module, we used CMUSphinx English, Spanish, Italian and Greek dictionaries provided here:

https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models

However, even though English and Spanish models are widely supported in the web, the other two languages are poorly supported. Additionaly, for the beAWARE needs, we should extend the list of location names in all dictionaries and also include code words and keywords, in order to improve recognition accuracy and enable event localization through semantic extraction. For these reasons, we have started collecting lists of location names and other words and we have also built g2p-seq2seq tool [34], which is used in order to extend the dictionary. It is based on neural networks, it is implemented in the Tensorflow framework and provides a state-of-the-art accuracy of conversion. In order to expand an existing dictionary (the CMU English dictionary[35] for example) an initial graph-to-phoneme

[34] https://github.com/cmusphinx/g2p-seq2seq
[35] https://github.com/cmusphinx/cmudict

model is needed (for example, an English model 2-layer LSTM with 512 hidden units, trained on the CMU English dictionary is available for download in CMU Shpinx website[36]). By training the existing model in the new dictionary, we create pronounciations for the new words included in the extended dictionary.

As previously mentioned, after extending the dictionary, the language model should also be extended, in order to include the new words. Thus, a new language model should be build. Shortly, building a statistical language model consists of the following steps:

1) Text preparation: a large collection of clean texts containg all the words of interest, without abbreviations or non-word items. In order to clean Wikipedia XML dump, for example, special Python scripts like Wikiextractor can be used.
2) Generation of the vocabulary file. This is a list of all the words in the file.
3) Generation of the language model file (in text ARPA format, binary BIN format or binary DMP format) by using one of the availble training toolkits, like: SRILM[37], CMUCLMTK[38], IRSLM[39] or MITLM[40].

Until now, we have started creating lists with missing words and we have trained some initial language models. However, this is an ongoing process and will be completed in the second Prototype.

5.3 Adapting the acoustic model

In order to improve recognition accuracy, we also performed an initial acoustic model adaptation. The adaptation process takes new transcribed data and improves the model we already have. It does not necessary adapt for a particular speaker. It just improves the fit between the adaptation data and the model. In order to perform adaptation we had to build PocketSphinx[41] (a lightweight recognizer library written in C) in Eclpise, along with the prerequisite libraries and training tools SphinxBase and Sphinxtrain.

The first step of the adaptation is the creation of the adaptation corpus. The corpus consists of: a) a list of sentences, b) the corresponding speech recordings of these sentences and c) a dictionary describing the pronunciation of all the words in that list of sentences.

[36]https://sourceforge.net/projects/cmusphinx/files/G2P%20Models/g2p-seq2seq-model-6.2-cmudict-nostress.tar.gz/download
[37] http://www.speech.sri.com/projects/srilm/
[38] https://cmusphinx.github.io/wiki/cmuclmtkdevelopment/
[39] https://sourceforge.net/projects/irstlm/
[40] https://github.com/mitlm/mitlm
[41] https://github.com/cmusphinx/pocketsphinx

For this reason, we conducted a review on publicly available speech corpora containing 'clean' speech and noisy recordings in English, Spanish, Italian and Greek. Several corpora were selected after using criteria such as: language spoken, duration, noise contamination, availability of annotations, etc. Some of these databases are: Voxforge speech corpus[42], LibriSpeech ASR corpus[43], Santa Barbara Corpus of Spoken American English[44], CMU_SIN Database[45]. We also created new voice recordings, containing simulated emergency calls from real past incidents in Spanish, Greek and Italian. Collected data were mentioned in D3.2. From the collected data, a subset of annotated speech recordings was selected and prepared for acoustic model adaptation.

For the adaptation process, existing acoustic models are copied into the working directory of Pocketsphinx, along with the dictionaries and the language models. Then, by using Pocketsphinx, a set of acoustic model feature files is generated from the audio recordings. This is done with the sphinx_fe tool from SphinxBase. The same acoustic parameters should be used for the extraction of these features as were used to train the standard acoustic model.

The next step is to collect statistics from the adaptation data. This is done using the bw program from SphinxTrain. Here, also the arguments in the bw command should match the parameters of the acoustic model.

The next step is the creation of a transformation with Maximum Likelihood Linear Regression (MLLR). MLLR is a cheap adaptation method that is suitable when the amount of data is limited. For best accuracy MLLR adaptation is combined with Maximum a Posteriori Adaptation (MAP) (Oh & Kim, 2009). The mllr_solve program of SphinxTrain generates the MLLR transform, which is passed to the decoder to adapt the acoustic model at run-time. This command will create an adaptation data file called mllr_matrix. Subsequently, the adaptation is completed by running the map_adapt program, in order to adapt acoustic model files.

The aforementioned adaptation process has currently been performed on a subset of the collected datasets, which was carefully prepared for adaptation. In order to improve recognition accuracy, we will continue to collect and prepare new telephone audio recordings and we will further adapt models as new recordings become available.

[42] http://www.voxforge.org/home/downloads
[43] http://www.openslr.org/12/
[44] http://www.linguistics.ucsb.edu/research/santa-barbara-corpus
[45] http://www.festvox.org/cmu_sin/

5.4 Integration of ASR component

The ASR component has been integrated into the operational beAWARE platform. The component supports all the four beAWARE languages (English, Spanish, Italian, Greek). The ASR communicates with the Media Hub component through socket messages. The Media Hub suscribes to TOP021_INCIDENT_REPORT and triggers the ASR module in case the attachmentType in this topic is set to "audio". The socket message form the Media Hub to ASR is a JSON message with the link to the audio file, the language, and the timestamp and the Incident ID. The input audio should be “RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, monophonic, 16000 Hz”. However, an encoder has also been included, in case the input format is not the appropriate. By reading the language information, ASR component selects the corresponding language model, invokes the transcriber, creates the transcription and sends a JSON message back to the Media Hub, with the transcription text. Subsequently, the Media Hub creates a TOP010_AUDIO_ANALYZED topic, containing, among other fields, the transcribed text and the language of the user. MTA, which is subscribed to this topic, receives this information in order to further analyze the transcription. The ASR is integrated as a Maven Java project and it currently uses 200m cpu and 2048Mi memory.

6 Text analysis v1

In the following we describe the currently developed and deployed text analysis pipeline. In its basic version, the same tools and resources, with the exception of the tweet normalization step, are applied for all three types of considered inputs. In particular, text messages sent through the mobile application share the instant messaging characteristics, including short length and limited context, found in tweets. Moreover, as far as the analysis of transcriptions of calls is concerned, the performance of ASR on the types of inputs considered within the 1st prototype has not necessitated for the moment spoken language-specific investigations. If in the course of development towards the 2nd and the final prototype, the quality of transcriptions deteriorates to an extent that prohibits the extraction of meaningful information from received calls, investigations into more flexible parsing techniques as well as overall pipeline adaptations to such types of inputs and the entailed ramifications will be investigated.

6.1 Text preprocessing

This step involves the sentence splitting, POS-tagging, lemmatization and morphological tagging tasks that are applied prior to the parsing and the steps that implement the extraction of the knowledge graph representation.

For sentence splitting and POS-tagging we use the Stanford CoreNLP toolkit, version 3.8.0, provided by DKPro 1.9.1. POS-tagging models were trained using the default options; an example property file for training the English POS-tagger is shown in Figure 9:Figure 9:.

```
model = ModelEN-UD.tagger
 arch = left3words
wordFunction =
trainFile = format=TSV,wordColumn=0,tagColumn=1,/parsers/trainingCorpus/UD/en-ud-train-pos.tsv
closedClassTags =
 closedClassTagThreshold = 40
 curWordMinFeatureThresh = 2
debug = false
debugPrefix =
tagSeparator = /
encoding = UTF-8
iterations = 100
lang =
learnClosedClassTags = false
minFeatureThresh = 5
 openClassTags =
rareWordMinFeatureThresh = 10
rareWordThresh = 5
search = qn
sgml = false
sigmaSquared = 0.5
regL1 = 1.0
```

```
tagInside =
tokenize = true
tokenizerFactory =
tokenizerOptions =
verbose = false
verboseResults = true
```

Figure 9: Default property file example for POS-tagging model training

For lemmatization, i.e., is the reduction of inflectional forms and sometimes derivationally related forms of a word to a common base form, and the extraction of morphological features, e.g. number, case, tense, etc., we used MateTools version 3.5, provided by DKPro 1.9.1. For training, we used anna-3.5 version for the lemmatizer and anna-3.3 for the morphological feature extractor; no specific training options were used, as illustrated in the following example training commands:

```
java -Xmx2G -classpath ~/misc/parsers/anna-3.5.jar is2.lemmatizer.Lemmatizer -model
/home/misc/parsers/definitiveTalnModels/es/LemmaES.tagger -train /home/misc/parsers/trainingCorpus/
AnCora-UPF -train.conll

java -Xmx2G -classpath ~/misc/parsers/anna-3.3.jar is2.mtag.Tagger -model
/home/misc/parsers/definitiveTalnModels/es/MorphES.tagger -train /home/misc/parsers/trainingCorpus/
AnCora-UPF -train.conll
```

In the case of tweet inputs, an additional, normalization step, is applied. The current implementation includes the removal of emoticons and other special Twitter-specific elements (e.g. RT, user, etc.), as well as the normalization of hashtags, which currently consists in the removal of "#" character and the splitting of CamelCase words; towards a more tweet language-tailored approach, the ArktweetTokenizer, version 0.3.2, provided by DKPro 1.9.1 has been used in the current implementation.

6.2 Parsing

Till out ongoing investigations into a linguistic-driven methodology for parser evaluation with respect to downstream application requirements conclude, and given its low memory requirements compared to other state of the art parsers along with its competitive accuracy, the MST parser has been selected for deployment in the current implementation. More specifically, we are using version 0.5.1, provided by DKPro 1.9.1. For its training, the following corpora (see Section 3.3.2) have been used: UD-Italian-ISDT for Italian; UD_Greek-GDT for Greek; AnCora-UPF and UD-Spanish Ancora for Spanish; PennTreeBank and UD for Enligh.

The current NLP analysis pipeline outputs three different types of structures, which correspond to three different levels of abstraction of the linguistic description:

- SSynt: surface-syntactic structures (SSyntSs), i.e., language-specific syntactic trees with fine-grained relations over all the words of a sentence;
- DSynt: deep-syntactic structures (DSyntSs), i.e., language-independent syntactic trees with coarse-grained relations over the meaning-bearing units of a sentence;
- PredArg: predicate-argument structures (PerdArgSs), i.e., language-independent directed acyclic graphs with predicate-argument relations over the meaning-bearing units of a sentence.

This stratified view is strongly influenced by the Meaning-Text Theory (MTT) presented in (Mel'cuk, 1988). The MTT model supports fine-grained annotation at the three main levels of the linguistic description of written language: semantics, syntax and morphology, while facilitating a coherent transition between them via intermediate levels of deep-syntax and deep-morphology. At each level, a clearly defined type of linguistic phenomena is described in terms of distinct dependency structures.

In the framework of beAWARE, UPF is using primarily a Universal Dependency-based pipeline, which uses similar approaches and tagsets across languages. In order to circumvent possible issues due to the unequal annotation quality of the UD structures, we develop in parallel tools that target a language in particular. During the first half of the project, a Penn Treebank-based pipeline has been setup in English. If the approach shows significantly more efficient than the UD-based pipeline, we will research similar approaches for other languages, using the alternative resources described in Section 3.3 .

6.2.1 Towards a uniform UD-based pipeline

Universal Dependencies is a generic framework for cross-lingual syntactico-semantic annotation that has been applied to over 60 languages so far, for a total of over 100 different treebanks[46]. Most treebanks have been obtained through automatic conversions of other treeebanks, themselves in general obtained via automatic annotation. The resulting annotations are known to lack consistency and quality, but they have the advantage to provide a framework that reduces the differences across different languages. In beAWARE, we intend to test the usability of Universal Dependencies as intermediate representations for multilingual relation extraction.

For surface-syntactic parsing, we train the off-the-shelf MST parser on the freely available UD corpora of the beAWARE languages (English, Spanish, Italian, and Greek); see Section 3.3 . The resulting surface structures are syntactic trees with lemmas, part-of-speech tags,

[46] http://universaldependencies.org/

morphological and dependency information under the form of grammatical functions such as *subject*, *object*, *adverbial*, etc.

The deep structures in this configuration consist of predicate-argument structures obtained through the application of graph-transduction grammars to the UD surface-syntactic structures. The deep and surface structures are aligned node to node. In the deep structures, we aim at removing all the information that is language-specific and oriented towards syntax:

- determiners and auxiliaries are replaced (when needed) by attribute/value pairs, as, e.g., Definiteness, Aspect, and Mood:
  - auxiliaries: *has overflowed-> overflow*;
  - determiners: *the bridge-> bridge*;
- functional prepositions and conjunctions that can be inferred from other lexical units or from the syntactic structure are removed;
  - *reported by X-> reported X*
- edge labels are generalized into predicate argument (semantics-oriented) labels in the PropBank/NomBank fashion:
  - *subject*(*reported, by X)-> FirstArgument(report, X)*

The UD-based pipeline doesn't make any use of lexical resources at this point; the predicate-argument relations are derived using syntactic cues only. The deep input is a compromise between (i) correctness and (ii) adequacy in a generation setup. Indeed, the conversion of the UD structures into predicate-argument structures depends not only on the mapping process, but also on the availability of the information in the original annotation. Table 17 shows that different labels that the UD-based graph-transduction grammars currently produced.

Table 17: Semantic labels in the output of the UD-based pipeline

Semantic label	Type	Description	Example
A1/A1INV	Core	1st argument of a predicate	reported-> a citizen
A2/A2INV	Core	2nd argument of a predicate	reported-> a flood
A3/A3INV	Core	3rd argument of a predicate	reported-> to the authorities
A4, A5, A6	Core	4th to 6th arguments	*Very uncommon*
AM	Non-Core	None of governor or dependent are argument of the other	reported-> sending a message
LIST	Coordinative	List of elements	reported-> and-> warned
NAME	Lexical	Part of a name	bridge-> Angeli
DEP	UKN	Undefined dependent	N/A

The following phenomena should be highlighted:

- **Alignment between surface and deep nodes**
  On the deep nodes, we use one or more feature id with as suffix the line number of the corresponding surface nodes: on a deep node, id1=4|id2=15 means that this deep node is aligned with the surface nodes on the lines 4 and 15 of the corresponding surface structure. Only elements triggered by other elements (as opposed to be triggered by the structure of the sentence) are aligned with deep nodes. That is, a subcategorized preposition is aligned with a deep node, while a void copula or an expletive subject is not.
- **Core relations**
  Each defined core relation is unique for each predicate: there cannot be two arguments with the same slot for one predicate. If a predicate has an A2 dependent, it cannot have another A2 dependent, and it cannot be A2INV of another predicate.
- **Auxiliaries**
  Auxiliaries are mapped to the universal feature "Aspect".[47]
- **Conjunctions/prepositions**
  The prepositions and conjunctions maintained in the deep representation can be found under an A2INV dependency. A dependency path Gov-AM-> Dep-A2INV-> Prep is equivalent to a predicate (the conjunction/preposition) with 2 arguments: Gov <-A1-Prep-A2-> Dep.
- **Modals**
  They are mapped to the universal feature "Mood".
- **Pronouns**
  - Relative: only subject and object relative pronouns directly linked to the main relative verb are removed from the deep structure.
  - Subject: a dummy pronoun node for subject is added if an originally finite verb has no first argument and no available argument to build a passive; for a pro-drop language such as Spanish, a dummy pronoun is added if the first argument is missing.
- **Punctuations**
  Only the final punctuations are encoded in the deep representations: the main node of a sentence indicates if the latter is declarative, interrogative, exclamative, suspensive, or if it is involved in a parataxis, with the feature "clause_type".

Our graph-transduction grammars are rules that apply to a subgraph of the input structure and produce a part of the output structure. During the application of the rules, both the

[47] http://universaldependencies.org/u/feat/index.html

input structure (covered by the leftside of the rule) and the current state of the output structure at the moment of application of a rule (i.e., the rightside of the rule) are available as context. The output structure in one transduction is built incrementally: the rules are all evaluated, the ones that match a part of the input graph are applied, and a first piece of the output graph is built; then the rules are evaluated again, this time with the rightside context as well, and another part of the output graph is built; and so on. The transduction is over when no rule is left that matches the combination of the leftside and the rightside. Table 18 sums up the current state of the graph-transduction grammars and rules for the mapping between surface-syntactic structures and UD-based semantic structures.

Table 18: Graph-transduction rules for UD-based deep parsing.
***Includes rules that simply copy node features (~40 per grammar)**

Grammars	# rules*	Description
Pre-processing	76	Identify nodes to be removed Identify verbal finiteness and tense
SSynt-Sem	120	Remove idiosyncratic nodes Establish correspondences with surface nodes Predict predicate-argument dependency labels Replace determiners, modality and aspect markers by attribute-value feature structures Identify duplicated core dependency labels below one predicate
Post-processing	60	Replace duplicated argument relations by best educated guess Identify remaining duplicated core dependency labels (for posterior debugging)

Figure 10 and Figure 11 respectively show a syntactic structure as parsed by the MST parser and the semantic structure produced by the graph-transduction grammars. *Bacchiglione* is correctly identified as the first argument of *overflow* (A1), and the relation between overflow and bridge is correctly identified as non-core (AM), but no more information is provided at this point (in particular, that *bridge* is a location in this case); the fact that *Angeli bridge* is a named entity is not recognized by the pipeline. The relations with the suffix *INV* indicate an inverted core relation between the two elements; their purpose is to maintain a tree format (in which every node has at most one governor), easier to process, as opposed to a graph format (in which a node can have several governors).

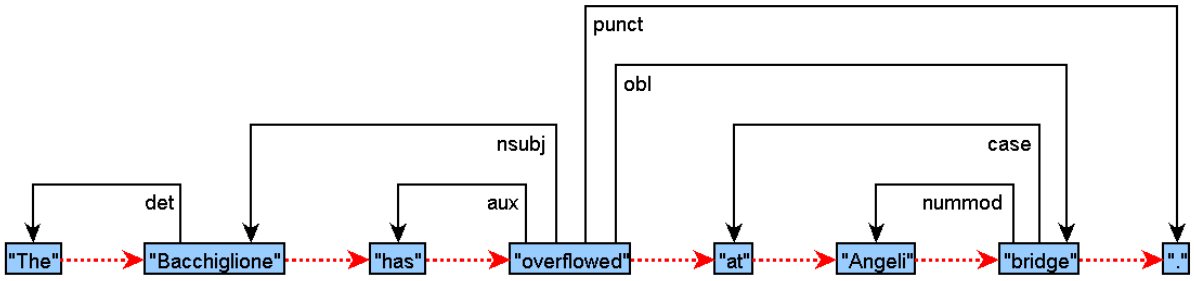

Figure 10: Surface-syntactic UD-Structure

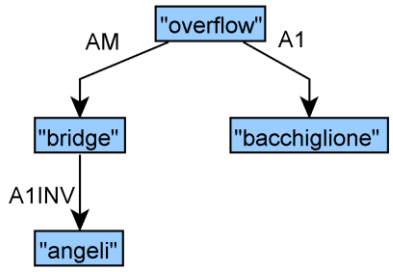

Figure 11: UD-based predicate-argument structure

The main issue with these deep structures is that they are underspecified, that is, some information is missing. The main reason is that one of our main objectives has been to remove as much information as possible that cannot be inferred from a deeper level of abstraction, as, e.g., an ontological representation. For instance, if it is not possible --or too risky- to predict an argument slot, we leave it undefined; if, because the annotation doesn't allow to distinguish between the two, we have a choice between leaving too many syntactic elements or removing meaningful words, we choose to remove words. This way, our deep representation is much closer to one actually used in a generation pipeline that starts from abstract data, as the one used in beAWARE in WP5.

At this point, all structures are still containing language-specific lexical units, and they need to be mapped to a language-independent vocabulary. This is performed through the use of a “conceptual” dictionary that contains abstract unambiguous labels (in English) and the corresponding lexical units in the different languages, as shown in the following sample entry (“NN” refers to nouns, “VB” to verbs, and “JJ” to adjectives):

```
"danger" {
  ENG = {
    lex = "danger_NN_01"
    lex = "endanger_VB_01"
```

```
    lex = "dangerous_JJ_01"
  }
  ITA = {
    lex = "pericolo_NN_01"
    lex = "pericoloso_JJ_01"
  }
  SPA = {
    lex = "peligro_NN_01"
    lex = "amenazar_VB_01"
    lex = "peligroso_JJ_01"
  }

}
```

Another graph-transduction grammar (under development) takes care of producing the final structure. Table 19 summarizes the tools used in the UD-based pipeline.

Table 19: Tools used in the UD-based pipeline

	System used
Tokenizer	Stanford Core NLP
Lemmatizer	Stanford Core NLP
PoS Tagger	Stanford Core NLP
Morhological tagger	Stanford Core NLP
Syntactic parser	MST
Semantic parser	UPF grammars

6.2.2 Towards language-specific pipelines

For alternative English surface-syntactic (SSynt) annotation, many annotation schemes are available. We chose to use the Penn Treebank representation followed in the CoNLL'09 shared task on dependency parsing, because we believe it is one of the most syntactically sound representations that are available; in particular:

- Its dependency tagset is fine-grained enough to take into account the most basic syntactic properties of English; unlike the UD-based tagset that is a hybrid syntax/semantics tagset, which does not reach the same level of syntactic fine-grainedness.
- One lexeme corresponds to one and only one node in the tree. For instance, in a relative clause, the relative pronoun is viewed from the perspective of its function in the relative clause and not from the perspective of its conjunctive properties.
- Unlike in UDs, the subject is a dependent of the inflected top verb, not of the non-finite verb, which might also occur in the sentence. This accounts for the syntactic agreement that holds between the auxiliary and the subject; the relation between the non-finite verb and the subject is more of a ``semantic'' one, and thus made explicit at a higher level of abstraction. The finite verb in an auxiliated construction is a dependent of the closest auxiliary.

- Again unlike UDs, subordinating and coordinating conjunctions depend on the governor of the first group, and govern the one of the second group. This hierarchical approach accounts for the linking properties of conjunctions. Exceptions to this are the relative pronouns, as mentioned above.

For the Penn Treebank-based analysis pipeline, we use Bohnet and Nivre's 2012 joint parser and tagger (see Section 3.3 for reference), to which we plugged in another set of graph-transduction grammars. The pipeline currently outputs deep structures at two different levels of representation:

- **DSynt**: deep-syntactic structures (DSyntSs), i.e., syntactic trees with coarse-grained relations over the meaning-bearing units of a sentence;
- **PredArg**: predicate-argument structures (PerdArgSs), i.e., directed acyclic graphs with predicate-argument relations over the meaning-bearing units of a sentence.

Deep syntactic (DSynt) structures are dependency structures that capture the argumentative, attributive and coordinative relations between full words (lexemes) of a sentence. Compared to SSynt structures, in DSynt structures, functional prepositions and conjunctions, auxiliaries, modals, and determiners are removed, as in the deep UD structures. Each lexeme is associated with attribute/value pairs that encode such information as part of speech, verbal finiteness, modality, aspect, tense, nominal definiteness, etc. The nodes are labeled with lemmas; in addition, they are aligned with the surface nodes through attribute/ value pairs (each DSynt node points to one or more SSynt node, using the surface IDs). All nodes have a PoS feature, which is copied from the SSynt output. The abstraction degree of the DSynt structures is in between the output of a syntactic dependency parser and the output of a semantic role labeler as the PredArg structures presented below: on the one hand, they maintain the information about the syntactic structure and relations, but, on the other hand, dependency labels are oriented towards predicate/argument relations, and the dependencies directly connect meaning-bearing units, that is, meaning/void/functional elements are not available anymore. Predicate-argument relations include I, II, III, IV, V, VI; modifier relations include ATTR and APPEND (the latter is used for modifiers that generally correspond to peripheral adjuncts); the other two relations are COORD (for coordinations) and NAME (connecting parts of proper nouns). Table 20 summarizes the different labels used at this level.

Table 20: Deep-syntactic labels

I	Core	1st argument of a predicate	reported-> a citizen
II	Core	2nd argument of a predicate	reported-> a flood
III	Core	3rd argument of a predicate	reported-> to the authorities
IV, V, VI	Core	4th to 6th arguments	*Very uncommon*
ATTR	Non- Core	Adjunct	reported-> yesterday
COORD	Coordinative	List of elements	reported-> and-> warned
NAME	Lexical	Part of a name	bridge-> Angeli

APPEND	Non- Core	Peripheral adjunct	reported-> (sending a message)

In order to obtain DSynt structures, as for the UD-based pipeline, we run a sequence of rule-based graph transducers on the output of the SSynt parser. But unlike the UD-based grammars, the SSynt-DSynt mapping is based on the notion of hypernode. A hypernode, known as syntagma in linguistics, is any surface-syntactic configuration with a cardinality equal or superior to 1 that corresponds to a single deep-syntactic node. For example, to *report* or *a citizen* constitute hypernodes that correspond to the DSynt nodes *report* and *citizen* respectively. Hypernodes can also contain more than two nodes, as in the case of more complex analytical verb forms, e.g., *would have been reported*. In this way, the SSyntS–DSyntS correspondence boils down to a correspondence between individual hypernodes and between individual arcs, such that the transduction embraces the following three subtasks: (i) hypernode identification, (ii) DSynt tree reconstruction, and (iii) DSynt arc labeling.

Table 21 shows the different steps of the SSynt–DSynt mapping. During a two-step preprocessing, specific constructions and hypernodes are marked. Auxiliaries, meaning-void conjunctions and determiners are easy to identify, but to know which prepositions belong to the valency pattern (subcategorization frame) of their governor, we need to consult a lexicon extracted from PropBank and NomBank. The output of these preprocessing steps is still a SSynt structure. The third transduction (SSynt-DSynt) is the core of this module: it "wraps" the hypernodes into a single node and manages the labeling of the edges, again looking at the PropBank-based lexicon (i.e., at the valency pattern of the predicates), together with the surface dependencies. For instance, a subject of a passive verb is mapped to a first argument (I), while the subject of a passive verb is mapped to a second argument (II). An object introduced by the functional preposition to is mapped to second argument in the case of the predicate want, but to the third in the case of give, etc. The SSynt-DSynt mapping inevitably produces duplications of core relations, which need to be fixed. The post-processing grammar evaluates the different argument duplications and modifies some edge labels in order to get closer to a correct structure.

Table 21: Graph-transduction rules for deep-syntactic parsing. *Includes rules that simply copy node features (~30% of the rules in each grammar)

Grammars	# rules*	Description
Pre-processing 1	15	Assign default PB/NB IDs. Mark passive, genitive, possessive constructions.
Pre-processing 2	17	Mark hypernodes.
SSynt-DSynt	55	Wrap hypernodes. Assign DSynt dependencies. Transfer aspect/modality as attr. Mark duplicate relations.

		Mark relative clauses.
Post-processing	78	Relabel duplicate relations. Reestablish gapped elements. Mark coord. constructions.

Predicate-argument (PredArg) structures are representations with abstract semantic role labels which also capture the underlying argument structure of predicative elements (which is not made explicit in syntax). Lexical units are tagged according to several existing lexico-semantic resources, namely PropBank, NomBank, and VerbNet. The current system is limited to choose the first meaning for each word. During this transition, we also aim at removing support verbs. For the time being, this is restricted to light be-constructions, that is, constructions in which the second argument of be in the DSyntS is a predicate P that can have a first argument and that does not have a first argument in the structure. In this case, the first argument of the light be become the first argument of P in the PredArg representation; for instance, a structure like *levee <-I be II-> cracked* is annotated as *cracked A1-> levee*.

The predicate-argument relations are sorted in two subtypes: on the one hand, the argumental, or "core" relations: Argument1, Argument2, Argument3, Argument4, Argument5, Argument6; and, on the other hand, the "non-core" relations: Benefactive, Direction, Extent, Location, Manner, Purpose, Time, NonCore (which is the only underspecified relation). The non-core labels come mainly from the corresponding labels in the Penn Treebank, that is, they are provided by the surfacesyntactic parser. Table 22 lists the relations used at the PredArg level.

Table 22: Predicate-argument labels

Semantic label	Type	Description	Example
Argument 1	Core	1st argument of a predicate	reported-> a citizen
Argument 2	Core	2nd argument of a predicate	reported-> a flood
Argument 3	Core	3rd argument of a predicate	reported-> to the authorities
Argument4,5,6	Core	4th to 6th arguments	*Very uncommon*
Benefactive, Direction, Extent, Location, Manner, Purpose, Time	Non-Core	Circumstancials	flood-> at Angeli Bridge (Location)
NonCore	Non-Core	None of governor or dependent are argument of the other	reported-> sending a message
NAME	Lexical	Part of a name	bridge-> Angeli
Set	coordinative	List of elements	reported-> and-> warned
Elaboration	Non- Core	Underspecified	reported-> (sending a message)

In order to obtain the PredArg structures, we run another sequence of graph-transducers on the output of the DSynt parser. The first grammar in this module creates a pure predicate-

argument graph, with the mapping of DSynt relations onto PredArg relations according to PropBank/NomBank. Coordinating conjunctions are linking elements in the Penn Treebank and DSynt representations; in a predicate-argument graph, they are represented as predicates, which have all the conjuncts as arguments and which receive all incoming edges to the coordinated group. Lexical units are assigned a VerbNet class. Once this is done, a few post-processing grammars are applied; they recover the shared arguments in coordinated constructions, remove light verbs, and remove the distinction between external and non-external arguments (i.e., for all predicates that have an A0, we push all the arguments one rank up: A0 becomes A1, A1 becomes A2, etc.). PropBank, NomBank, VerbNet classes are assigned through a simple dictionary lookup. For this purpose, we built dictionaries that can be consulted by the graph-transduction environment and that contain the classes and their members, together with the mappings between them. Tab summarizes the different steps of this module.

Table 23: Graph-transduction rules for mapping to PredArg structures. *Includes rules that simply copy node features (~30% of the rules in each grammar)

Grammars	# rules*	Description
DSynt-Sem	59	Assign core dependencies. Recover shared arguments. Establish coord. conj. as predicates. Assign VerbNet classes.
Post-processing 1	11	Recover shared arguments in coordinated constructions. Mark light verbs.
Post-processing 2	23	Remove light verbs. Assign frames (FrameNet).
Post-processing 3	30	Normalize argument numberings.
Post-processing 4	31	Introduce non-core dependencies

Figure 12 and Figure 13 show sample structures at the three points in this analysis pipeline.

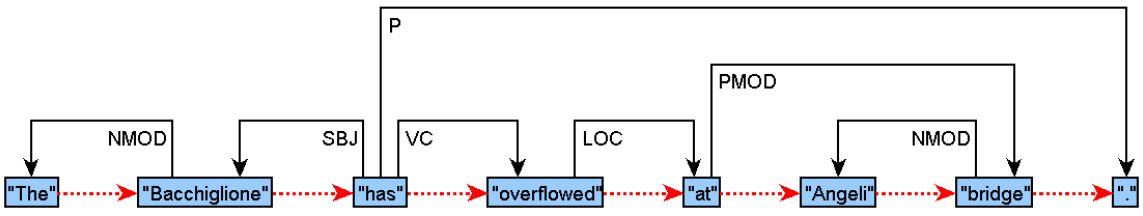

Figure 12: Surface-syntactic Structure

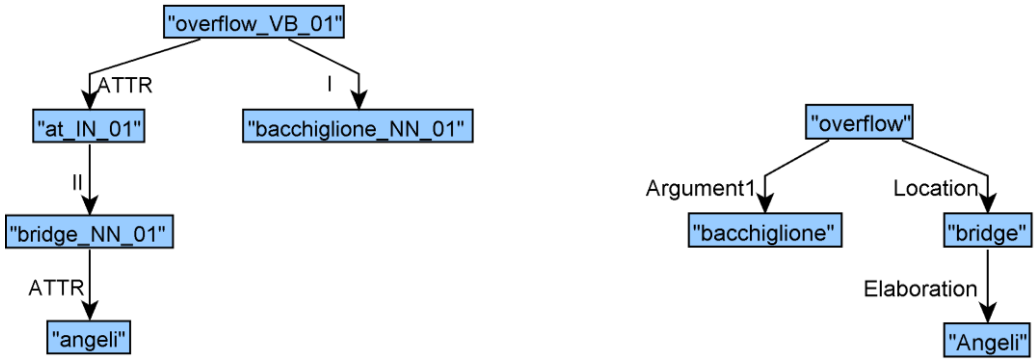

Figure 13: Deep-syntactic (Left) and PredArg (Right) structures

Table 24 summarizes the tools used in the English-specific pipeline.

Table 24: Tools used in the Penn Treebank-based pipeline

	System used
Tokenizer	Stanford Core NLP
Lemmatizer	MATE tools
PoS Tagger	MATE tools
Morhological tagger	MATE tools
Syntactic parser	MATE tools
Semantic parser	UPF grammars

6.3 Knowledge graph derivation

6.3.1 Entities and events semantic resolution

In order to determine the semantics of the extracted entities and events, the current implementation uses DBpediaSpotlight (off-the-shelf REST service version 1.0, through the implementation of a pipeline-compliant wrapper) to obtain de-referenceable links against DBpedia.

At this stage, the obtained DBpedia links are propagated as such in the subsequent steps that deal with the identification of locations and the factoring of the resulting graph-based representation of the analyzed inputs. Specifically, there is no checking and validation with respect to their meaningfulness and coherency considered within the overall context of the analyzed input, thus leaving room for noisy identity resolution. For example, English mentions of “square” tend to be identified as referring to Square Tallaght, a shopping mall in Dublin (http://dbpedia.org/resource/The_Square_Tallaght), rather than as instances of the DBpedia class “dbo:Square” (http://dbpedia.org/ontology/Square); this said, it is interesting to note that even for explicit mentions of given squares (e.g., the Times square, the Aristotelous square, etc.) for which corresponding DBpedia resources exist, in several cases

it is not straightforward to extract such instance-of relationships as such information is not necessarily included in the DBpedia knowledge base (i.e., posing a SPARQL query in order to retrieve the types of the given resources, will not include dbo:Square in the returned classes).

The addition of a validation and overall coherency encoding methodology will also contribute to the mitigation of ramifications pertinent to incompleteness of the reference knowledge base, in our case DBpedia. Although not integrated in the current version, we have been already investigating in parallel linking against BabelNet towards the leverage of further external knowledge that can enable the amelioration of such phenomena.

For location extraction, we combine Stanford NER and linguistic dependency-based patterns to identify candidate location mentions, in synergy with the DBpediaSpotlight links, in order to determine whether a mention refers actually to a location or not. The approach is based on the following premises: i) if a place-indicating mention, such as “square”, “bridge”, “street”, etc., is linked via a NAME dependency to a proper name one, then the concatenation of is marked as a location; ii) if a DBpedia resource link has been obtained for a, single- or multi-word, mention, and among its DBpedia types, the classes dbo:Place or dbo:SpatialThing are included, then the mention is marked as a location; iii) likewise, if the mention under consideration has been tagged by the NER tool as a location.

Though fairly simple, such approach can afford adequate performance, provided that the locations of interest are included in DBpedia and that dependency parsing achieves reasonably robust performance. However, neither of the two can be taken for granted, and as already outlined neither state-of-the-art NER tools nor DBpedia, as a resource, can afford the extent of coverage required for the location detection and geotagging needs within beAWARE. Towards the latter, we have started investigations into the use of OpenStreetMap as the reference geo-knowledge base and into the extension of the candidate selection mechanism so as to ensure as a much a comprehensive coverage as possible, while avoiding the scalability and portability issues resulting from gazetteer-based search approaches and hand crafted patterns.

6.3.2 Event-centric representation

Once the outputs of the afore-described NLP tasks are available, the final step consists in their aggregation and factoring into a structured representation that can form the basis for the population of the beAWARE knowledge base, where the analysis results of all modalities end up in order to be semantically integrated and enable the elicitation of further, implicit knowledge, by means of reasoning.

The resulting linguistic structures described in the previous Section form the backbone for the derivation of structured, knowledge graph representations. Each predicate and argument mention is mapped to a respective instance occurrence; the type currently is a direct reflection of the lexical form, further augmented via DBpedia pertinent information; the latter though, is currently only implicit, as in view of the under-investigation semantic coherency approach, we refrain from currently propagating the associated DBpedia classes, that would currently amount to a mere list rather than their consolidation.

Drawing upon the knowledge graph extraction paradigms discussed in Section 3.3 the extracted frames (i.e. the semantic predicates), with the current focus being on verbal and nominal ones, are represented as reified objects, connected to their participants by means of properties that determine their semantic roles. In addition, to semantic types and roles information, instances are enriched with textual features relevant to linguistic generation, such as number and label information; this is needed in order to ensure that the KB is populated with accurate information (e.g., distinguishing between plural and singular mentions has a direct impact on the cardinality of the mentioned entity). Thereby, we also ensure that the report generation requests, that follow the semantic integration and reasoning tasks taking place in the knowledge base, provide the level of detail required for producing the desired reports (e.g., in order to be able generate number-sensitive statements and thus report whether it is one car that is impacted or several cars). Likewise for label that is used for capturing the proper name of Named Entities occurrences in the pertinent language (e.g., “Matteotti square”; “piazza Matteotti”, etc). As currently there has been no need for catering for coreferential mentions, no further actions are taken; would this necessity materialize, then all mentions referring to the same real word entity or event would be mapped into a single instance.

In the figures that follow, example outputs are given in the JSON-based format developed for communication between the text analysis and knowledge base service modules. Figure 14: illustrates the extracted structured representation, given the input sentence "The sewers have flooded". As illustrated, two instances have been derived, namely one of type “Sewer” and one of type “Flood”, with the former being the element that undergoes the incident (flood in our case) denoted by the latter. As also shown, the “number” field for Sewer has the value “PL”, which stands for plural, while both instances have no “label” values, as neither of them refers to a Named Entity.

```
"data": {
        "sewer_1001607183": {
            "type": ["Sewer"],
            "label": "null",
            "location": false,
            "seeAlso": "http://dbpedia.org/resource/Sanitary_sewer",
            "number": "PL"
        },
        "flood_170488431": {
            "type": ["Flood"],
            "label": "null",
            "location": false,
            "seeAlso": "http://dbpedia.org/resource/Flood",
            "participants": [
                {
                    "role": "A2",
                    "participant": "sewer_1001607183"
                }
            ],
            "number": "null"
        }
    }
```


Figure 14: Resulting knowledge graph for the input sentence "The sewers have flooded."


```
"data": {
        "square_1275910779": {
            "type": [
                "Square"
            ],
            "label": "Matteotti_square",
            "location": true,
            "number": "singular"
        },
        "flood_1464244210": {
            "type": [
                "Flood"
            ],
            "label": "null",
            "location": false,
            "seeAlso": "http://dbpedia.org/resource/Flood",
            "participants": [
                {
                    "role": "A2",
                    "participant": "square_1275910779"
                }
            ],
            "number": "null"
        }
    }
```


Figure 15: Resulting knowledge graph for the message "Matteotti square has flooded."

Figure 15 shows the resulting graph representation for the sentence “Matteotti square has flooded”, which does contain a Name Entity instance, i.e. “Matteotti square, thus resulting, as shown, in a corresponding “label” value. Last, Figure 16 shows an example graph for a non-English input, namely “L' argine vicino a ponte è crollato.", which means “The levee near the bridge has collapsed.”, depicting the rendering of the extracted Italian mentions into respective class instances. As can be observed, since for the moment the obtained DBpedia links are not processed further, only the URL of the localised Italian DBpedia is included.


```
"data": {
    "crollare_1776608452": {
       "type": ["Collapse"],
       "label": "null",
       "location": false,
       "participants": [
          {
             "role": "A1",
             "participant": "argine_261651685"
          }
       ],
       "number": "null"
    },
    "argine_261651685": {
       "type": ["Levee"],
       "label": "null",
       "location": false,
       "participants": [
          {
             "role": "AM",
             "participant": "ponte_892304417"
          }
       ],
       "number": "SG"
    },
    "ponte_892304417": {
       "type": ["Bridge"],
       "label": "null",
       "location": true,
       "seeAlso": "http://it.dbpedia.org/resource/Ponte",
       "number": "null"
    }
 }
```


Figure 16: Resulting knowledge graph for the message "L' argine vicino a ponte è crollato."

7 Evaluation

7.1 Visual analysis

In this section, an evaluation report for the developed methods and techniques regarding visual analysis is provided.

7.1.1 Fire and flood detection in social media images

For the fire and flood detection in social media images pipeline, as described in section 4.1 , quantitative results are provided for the EmC and the EmL modules and qualitative results for the overall framework's performance. We made several experiments on benchmark fire and flood datasets. For Emergency Classification (EmC) we use the MediaEval's Disaster Image Retrieval from Social Media (DIRSM) dataset (Avgerinakis, et al., 2017), while for Emergency Localization (EmL) we use the BowFire dataset (Chino, Avalhais, Rodrigues, & Traina, 2015).

Emergency classification evaluation

Image classification evaluation took place in MediaEval's Disaster image retrieval from social media (DIRSM) dataset, where flood and other type of images were provided. A 10 fold cross validation was followed to evaluate Emergency Classification (EmC) module. Recognition accuracy results and comparison with State-of-the-Art are provided in Table 25, where we can see that EmC outperforms all image classification methods that were presented in MediaEval's Multimedia Satelite Task 2017, scoring 1.77% higher from the second rival.

Table 25: Image classification results on DIRSM Dataset and comparison with SoA.

Authors	Accuracy
CERTH	**97.5%**
(Nogueira, et al.)	87.88%
(Avgerinakis, et al., 2017)	92.27%
(Ahmad, Konstantin, Riegler, Conci, & Holversen, 2017)	95.11%
(Lopez-Fuentes, Weijer, Bolaños, & Skinnemoen)	70.16%
(Dao, Pham, Nguyen, & Tien, 2017)	87.87%
(Ahmad, Ahmad, Ahmad, & Conci)	95.73%
(Bischke, et al., 2017)	95.71%

A separate classification evaluation also took place on a collection of social media images that were retrieved from the Flickr API[48] using search querries related to realistic fire of flood scenarios such as "flooded city", "forest fire" etc. Here, the classes were 3: "fire",

[48] https://www.flickr.com/services/api/

"flood" and "other", while there were some examples where some of the two coincide and the discrimination was quite difficult to tell. Nevertheless, our framework achieved a mean accuracy recognition rate that reached 87.32%, with 83.7% "other" class achieving the lowest score, "fire" the highest with 93.3% and 88.96% for "flood".

Emergency Localization evaluation

Table 26: Fire localization results on BowFire Dataset and SoA comparison.

Authors	Precision	Recall	F1-Score
CERTH	39%	77%	52%
(Celik & Demirel, 2009)	52%	68%	53%
(Rossi, Akhloufi, & Tison, 2011)	< 40%	20% - 30%	< 30%
(Rudz, Chetehouna, Hafiane, Laurent, & Séro-Guillaume, 2013)	**63%**	<50%	50% - 60%
(Chino, Avalhais, Rodrigues, & Traina, 2015)	50%	60 – 70%	50% - 60%
(Chen, Yeh, & Yin, 2009)	37%	**84%**	45%
(Avalhais, Rodrigues, & Traina, 2016)	62%	77%	**63%**
(Zhang, Wang, & Lv, 2013)	50%	31%	29%

Emergency localization evaluation took place on BowFire (Chino, Avalhais, Rodrigues, & Traina, 2015) and VideoWaterDB (Mettes, Tan, & Veltkamp, Water detection through spatio-temporal invariant descriptors, 2017) datasets for fire and flood segmentation respectively. Comparisons regarding the fire segmentation results took place on BowFire by computing recall and precision metrics and are depicted in Table 26. As far as recall is concerned, we can observe that our results are really close to (Chen, Yeh, & Yin, 2009) and tied for second place with (Avalhais, Rodrigues, & Traina, 2016) outperforming the rest, meaning that we found a great deal of pixels that were groundtruthed as fire. On the other hand, as far as precision is concerned, we didn't achieve as well as we expected, as there were a great deal of background pixels that were misclassified as fire, leading to lower precision rates than other SoA techniques. These false alarms however, can be alleviated in our warning framework, as the use of EmC component can eliminate a great deal of images that do not contain a threat, which can eventually increase the precision rate on the final severity level estimation.

Qualitative results

The overall framework's capabilities are evaluated with qualitative results and a set of successful and failure images are provide in Figure 17 bellow.

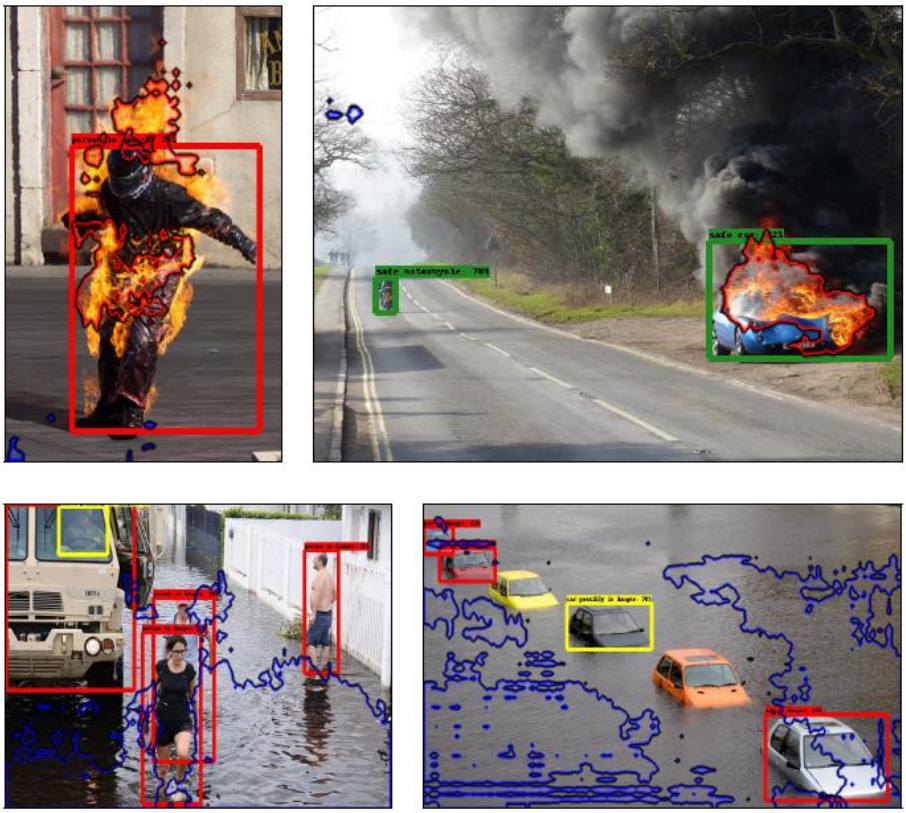

Figure 17: Qualitative results for the fire and flood detection system.

We visualize the severity level of danger in the resulting candidate target bounding boxes from the ObD component by using a three color palette to draw them: (a) Green for 'Safe' targets, (b) yellow for 'Possibly in danger' targets, and (c) red for targets classified as being 'in danger'. A second color pallet is used to draw the embossed regions that result from EmL component and are colored as red for fire and blue for flood regions. Analysing the qualitative results, we can see that in very demanding situations such as the top-left picture the framework is responding well, where a person fired up can be easily isolated from the background environment which is quite irrelevant with the emergent event. A successfully captured flood event is depicted in the bottom left picture, where we can see the people who are in the water obtain an 'in danger' label contrary to the one who is in the car and far from the flood and is labelled as 'possibly in danger'. Analysing now the failure cases, we can see that in some cases we might have a good EmL mask, but fail to recognize the picture as emergent using EmC, giving an erroneous 'safe' label, like the car which is on fire on the top-right picture. Other, more frequent cases of failure are showcased in the bottom right picture, where a series of flooded cars is depicted. As we can see there might be some cases where the ObD may not find all the targets or the EmL mask is not so well formed, leading to missing or erroneous labels. This was very usual in flood scenarios, where the water covers a great deal of the object or the object occludes the water, leading to bad bounding boxes and

segmentation masks, contrary to fire events where the fire usually occludes the target and not vice versa. Overall, on the most of the test samples that were examined, rarely a target in danger did not get at least a 'Medium' level tag. The most frequent cases of inconsistency and confusion happened between 'Medium' and 'High' level tags, because EmL did not work so well in flood cases, which is mainly attributed to the lack of groundtruth masks to train the model. Flood detection worked very well in the provided dataset but it the needs for more training data to achive better generalization in more emergent situations is evident.

7.1.2 Fire and flood detection in video samples

Regarding the Dynamic texture recognition and localization techniques described in 4.2 the evaluation took on four challenging benchmark datasets, namely Dyntex (Peteri, Fazekas, & Huiskes, 2010), MovingVistas (Shroff, Turaga, & Chellappa, 2010), Yupenn (Derpanis, Lecce, Daniilidis, & Wildes, 2012) and VideoWaterDatabase (Mettes, Tan, & Veltkamp, Water detection through spatio-temporal invariant descriptors, 2017). All datasets were split into 1/3 for testing and 2/3 for training, creating 3 different train/test splits. In all cases, our algorithm's accuracy was calculated in multiple tasks and compared with the SoA, demonstrating improved performance.

Dynamic texture recognition evaluation

Dyntex is one of the earliest and most renowned benchmark datasets for dynamic textures, containing a wide variety of texture classes including dynamic water textures. In our experiments, we use the benchmark classification split of DynTex dataset into three subsets: alpha, beta and gamma. These subsets contain video samples from 3, 10 and 10 different classes respectively, often including high intra-class variance. The overall average score of the method proposed is provided in Table 27, where it can be seen that it outperforms the SoA in all 3 subsets. More specifcally, compared against 6 other SoA works the results indicate remarkably high scores, exceeding 97% in all cases.

Table 27: Comparisons with SoA in DynTex dataset for alpha, beta and gamma splits.

	alpha	beta	gamma
CERTH	**100%**	**97.4%**	**98.0%**
(Dubois, Peteri, & Menard, 2015)	88%	66%	65%
(Smith, Lin, & Naphade, 2002)	83%	67%	65%
(Xu, Quan, Zhang, Ling, & Ji, 2015)	85.2%	76.9%	74.8%
(Zhao, Ahonen, Matas, & Pietikainen, 2012)	83.3%	73.4%	72%
(Xu, Huang, Ji, & FermΓOller, 2012)	83.6%	73.2%	72.5%
(Ji, Yang, Ling, & Xu, 2013)	84.8%	75.2%	73.3%

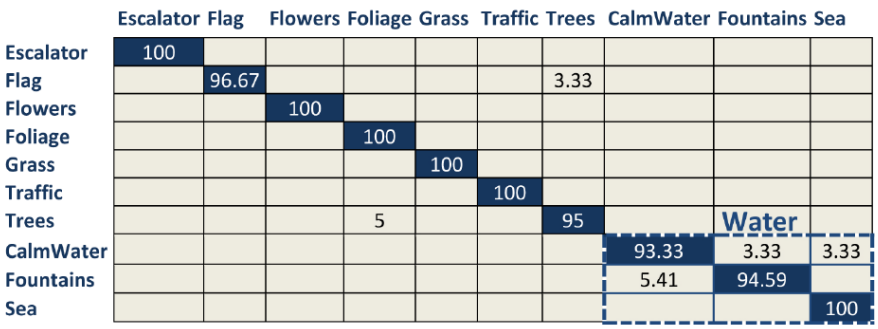

Figure 18: Multi-class classification accuracy of LBP-Flow in gamma split of DynTex dataset.

The confusion matrix is provided in Figure 18. As shown, the descriptor achieves high accuracy, over 95%, for 8 out of 10 classes. In the lower-right part of the matrix, it can be seen that misclassifications of water-related classes usually refer to other classes related to water texture, showing that the algorithm still detects water-related dynamic scenes robustly.

Moving vistas is the most challenging dataset of all, as it contains video samples of low quality using a moving camera, different viewpoints and significant illumination changes. The multi-class recognition accuracy of LBP-flow was estimated and compared with the SoA on scene recognition in (Derpanis, Lecce, Daniilidis, & Wildes, 2012) and (Shroff, Turaga, & Chellappa, 2010). The results, depicted in Table 28, show that our hybrid scheme achieves significantly better recognition rates compared to the SoA for the multi-classification task, with detailed classification accuracy for each class provided in Figure 19.

Table 28: Recognition accuracy on Moving vistas dataset.

	Score
CERTH	**67.7%**
(Derpanis, Lecce, Daniilidis, & Wildes, 2012)	41%
(Shroff, Turaga, & Chellappa, 2010)	52%

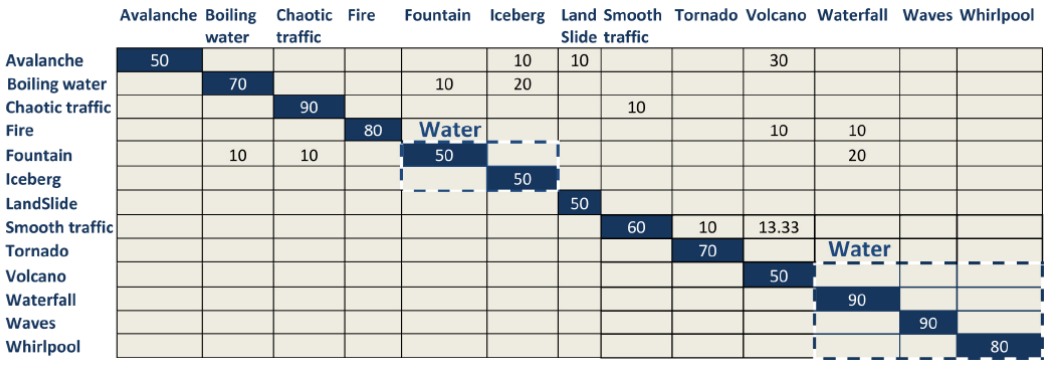

Figure 19: Multi-class classification accuracy of LBP-Flow in Moving vistas dataset.

Similarly, to the Dyntex data, the cost of feature extraction in the low resolution Moving vistas dataset is kept quite low, requiring about 9 fps. This low computational cost makes proposed method appropriate for near real time monitoring in surveillance applications.

YUPENN comprises of 420 videos, mainly of low quality, from 14 different classes including water and a forest fire dynamic texture class. It constitutes a challenging dataset, as each class is represented by a limited number of videos of short duration, ranging from 37 up to 180 frames. Despite these drawbacks, experiments on multi-classification tasks were conducted for all classes, with the results depicted in Table 29. In these experiments, LBP-flow is compared with many approaches from the SoA, also reported in (Mumtaz, Coviello, Lanckriet, & Chan, A Scalable and Accurate Descriptor for Dynamic Textures Using Bag of System Trees, 2015) and (Derpanis, Lecce, Daniilidis, & Wildes, 2012). It is clear that LBP-flow achieves remarkable accuracy rates for all classes, near or above 90%, including the water and fire tecture classes outperforming the SoA in many cases.

Table 29: Comparisons with SoA in YUPENN dataset for all classes.

Scene classes	**CERTH**	(Mumtaz, Coviello, et al., 2015)	(Nister & Stewenius, 2006)	(Derpanis, Lecce, Daniilidis, & Wildes, 2012)	(Grossberg & Huang, 2009)	Oliva and Torralba, 2001)	(Marszalek, Laptev, & Schmid, 2009)	(Shroff, Turaga, & Chellappa, 2010)
Beach	83.3	83	63	87	50	90	37	27
Street	100	90	70	83	47	50	83	17
Elevator	100	100	80	83	47	50	67	50
Forest fire	83.3	100	80	83	47	50	67	50
Fountain	87	67	37	47	13	40	30	7
Highway	95.7	87	73	77	30	47	33	17
Lighting storm	66.7	100	80	90	83	57	47	37
Ocean	91.7	90	80	100	73	93	60	43
Railway	95	80	73	87	43	50	83	3
River	95.8	80	73	93	57	63	37	3
Sky	95.8	93	77	90	30	90	83	33
Snowing	100	83	77	33	53	20	57	10
Waterfall	95.8	67	53	43	30	33	60	10
Farm	100	77	57	57	57	47	33	17

The VideoWaterDatabase introduced in (Mettes, Tan, & Veltkamp, Water detection through spatio-temporal invariant descriptors, 2017) consists of 260 high definition videos, where the presence of water needs to be detected. This dataset contains water and non-water samples from 7 and 5 classes respectively. The patterns between the two classes are quite similar and very diffidicult to model. Comparisons with other dynamic texture modeling

methods based on LBP are provided in Table 30, where the method is compared against 4 other SoA works, which use different approaches for texture representation.

Table 30: Comparisons with SoA in VideoWaterDatabase.

	Score
CERTH	**98.8%**
(Mettes, Tan and Veltkamp, 2017)	98.4%
(Zhao & Pietikainen, 2006)	93.8%
(Zhao, Ahonen, Matas, & Pietikainen, 2012)	93.3%
(Qi, Li, Zhao, Hong, & Pietikäinen, 2016)	97.2%

Dynamic tecture localization evaluation

Instances of the localization process for VideoWaterDatabase are provided in Figure 20. As it can be seen, the algorithm succeeds in capturing local nonwatery areas occupying only a small part of the frame (a),(e), while at the same time challenging water scenes containing shadows and running water (c),(h) are also correctly localized. The minor errors of our algorithm can be attributed to its general non-water based nature, and the omission of any post-processing steps which would smooth the final results.

We also examine our method's eficacy in localization task for the fire texture, using videos from the YUPENN dataset. Qualitative results of our algorithm's performance on the test videos are presented in Figure 21, where transparent blue color is used to depict regions where fire was detected by our method. The variations in fire texture and appearance are captured in most cases, demonstrating that our method can be efectively applied on different textures.

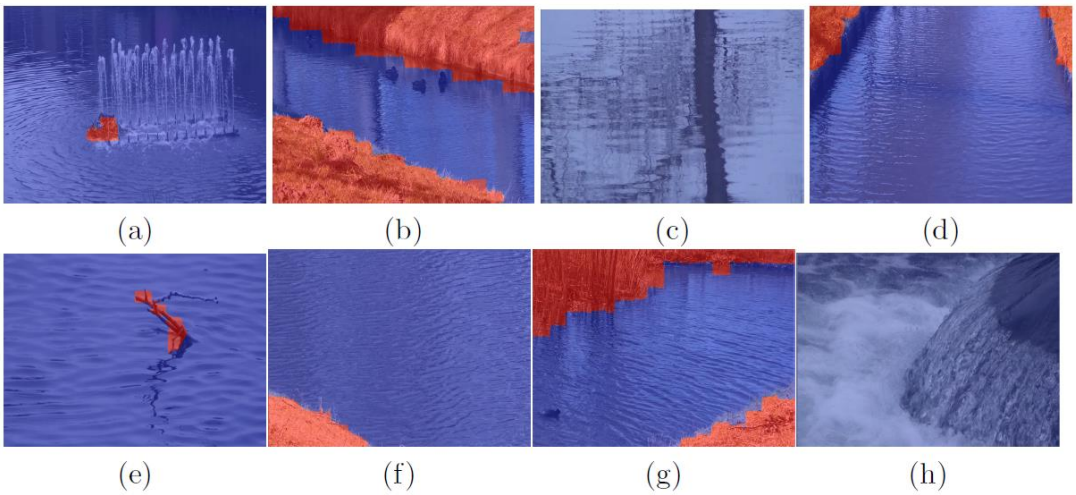

Figure 20: Instances of water localization in VideoWaterDatabase dataset.

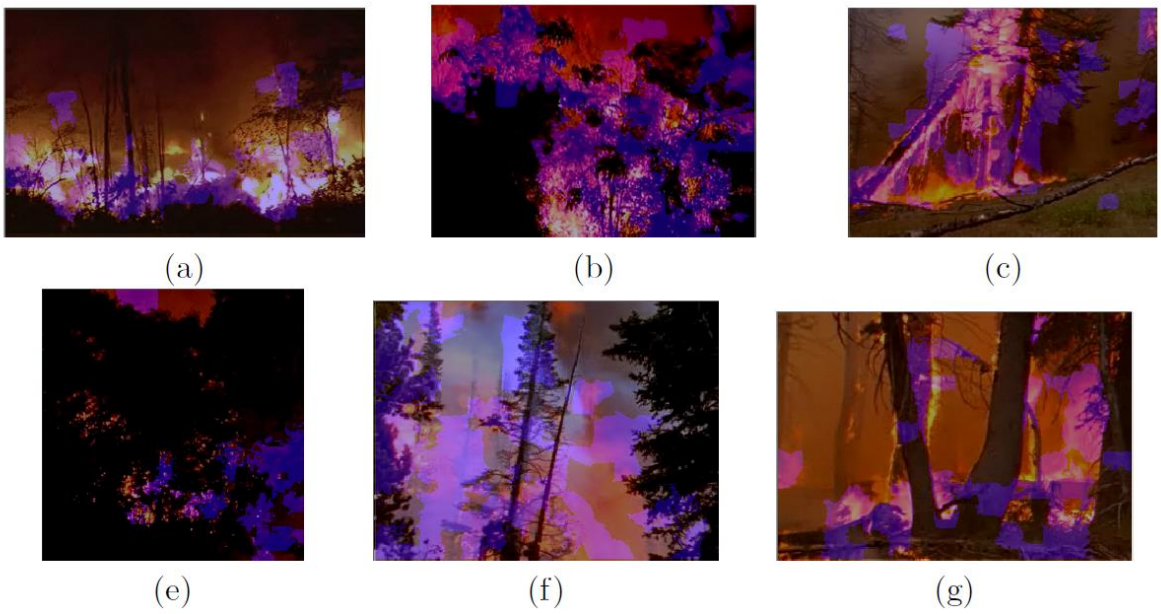

Figure 21: Instances of fire localization in Yupenn dataset.

7.1.3 Traffic analysis and management

We evaluated our traffic analysis methods in the NVIDIA AI CITY challenge[49]. Teams that entered the competition had the opportunity to experiment and evaluate their algorithms on three separate traffic analysis challenge tracks: (a) traffic flow analysis that focused on speed estimation of vehicles, (b) anomaly detection, for the detection of anomalous events such as car crashes or stalled vehicles and (c) multi-camera vehicle detection and re-identification. Moreover, a real-world evaluation dataset was made available for each track. In the sections that follow we present the experiments that took place and our evaluation scores for the first two tracks of the challenge.

Vehicle tracking and speed estimation evaluation

The dataset that was provided for the speed estimation track is composed of 27 videos and each one is a sequence of 1800 high definition resolution frames. The recorded videos depict highway traffic at several locations from various viewpoints. The camera in most of the videos is static, except at some locations where small trembling can be spotted, presumably due to windy conditions. Some videos also contain duplicate sequential frames at an unpredictable rate for an unknown reason which makes vehicles appear static. We presumed that when such frames appear no real time has passed and we chose to copy previous speed estimates. The overall score S1 of this track incorporates a measure of the quality of the detections and the accuracy of the speed measurement and is defined as:

[49] https://www.aicitychallenge.org

$$S1 = DR(1 - NRMSE)$$

where, DR is the detection rate and NRMSE is the normalized root mean square error (RMSE) of speed estimation. A vehicle is said to be detected if it was localized in at least 30% of frames it appeared in and NRMSE is the normalized RMSE across all submissions, obtained via min-max normalization. Our vehicle detector and tracker did manage to achieve a high detection rate score of 89%. The RMSE score of our speed estimator is 27.30 which appears to be higher that other teams participating in the challenge. Further investigation needs to take place in order to detect the possible inefficiencies of our speed estimation approach and further improve its accuracy.

Anomaly detection evaluation

The NVIDIA dataset for anomaly detection track comprises of 100 videos of 15 minutes each, at 800X410 resolution. It constitutes a challenging dataset as it contains a great variety of real traffic scenarios, severe camera motion, different weather conditions, illumination changes, occlusions and many low resolution shots. In our effort to handle video sequences containing a variety of camera motions, such as zoom in/out or even a complete change of view, we chose to divide them into sub clips characterized by static view, so as to develop different models in each of them. Evaluation is based on anomaly detection performance, measured by the F1-score, and detection time error, measured by RMSE. More specifically the score is calculated as:

$$S2 = F1(1 - NRMSE)$$

where a true-positive (TP) detection will be considered as the predicted anomaly within 5 minutes absolute time distance of the true anomaly that has the highest confidence score, a false-positive (FP) is a predicted anomaly that is not a TP for some anomaly and a false-negative (FN) is a true anomaly that was not predicted. RMSE is calculated between the ground truth anomaly time and the predicted time for all TP predictions. NRMSE is the normalized RMSE obtained from min-max normalization across all submissions. The algorithm reached an F1-score of 0.33 and the detection time RMSE was 227. Overall, we managed to surpass two other competing teams.

7.2 Audio analysis

In order to evaluate recognition results, we developed a testing framework, with the use of the 5prealpha release of Pocketshinx and Sphinxtrain. Before building Pocketshinx, first, we had to build also Sphinxbase (which is a support library required by Pocketsphinx). In order to test the effect of different audio sampling frequencies, we created multiple instances of the same audio file, by changing its sampling frequency. In our testing framework, we created separate folders for each language and each sampling frequency, containing test

audio files, along with the corresponding 'test.fileids' files (containing the names of the test audio files) and 'test.transcription' files (containing their corresponding transcriptions). To test recognition we used pocketsphinx_batch decoder, after setting the required parameters, as shown below:

```
pocketsphinx_batch \
 -adcin yes \
 -cepdir wav \
 -cepext .wav \
 -ctl test.fileids \
 -lm `<your.lm>` \    # relative path to the language model
 -dict `<your.dic>` \ # relative path to the dictionary
 -hmm `<your_hmm>` \  # relative path to acoustic model folder
 -hyp test.hyp
```

The pocketsphinx_batch command performs speech recognition in all audio files mentioned in the test.fileids file and writes the transcription result to test.hyp. Then, we use word_align.pl in order to compare recognition results with the original transcriptions, stored in test.transcription file, by using the following command:

```
word_align.pl   test.transcription   test.hyp
```

The word_align.pl is a perl script, which is part of Sphinxtrain distribution and is used to compute the error rate.

For evaluation of the ASR performance, we used the following datasets:

- http://www.openslr.org/12/
- http://www.repository.voxforge1.org/downloads/it/Trunk/Audio/Main/8kHz_16bit/
- http://www.repository.voxforge1.org/downloads/it/Trunk/Audio/Main/16kHz_16bit/
- http://www.repository.voxforge1.org/downloads/el/Trunk/Audio/Main/
- http://www.repository.voxforge1.org/downloads/es/Trunk/Audio/Main/

The overall resulting WERs for the four languages were

English = 20.11%, Italian =13.09%, Spanish =14.03%, Greek =21.09%

The poor recognition of English can be explained by the differences in the accent of the speakers. The model is trained on US English accent. Additionaly, since the testing datasets contain clean speech, results are expected to deteriorate if the models are tested in noisy environments. Future work will aim on more advanced denoising techniques.

Some other observations we made from the different testing trials are that:

- The original transcriptions should contain no punctuation marks, because word_align.pl counts punctuation marks and the words that are connected to them as errors.

- Special attention should be given to special characters in Italian, Spanish and Greek by saving the transcriptions in UTF-8 format.

- The sampling frequency of the audio recording and the frequency in which the acoustic model was trained should match. An acoustic model may have very poor detection rate when tested on recordings of different sampling frequency than the one used during its training.

- It seems that recording sampling frequency does not have an influence on recognition, if we convert the sampling frequency of the file to match the frequency of the acoustic model. For example, if we record an audio on 44kHz and we convert it to 16kHz and then test it with a 16kHz acoustic model, the recognition accuracy will be the same as if it was originally recorded on 16kHz.

- The English model is very vulnerable to accent. Since, it is trained on US English accent, it has low accuracy when tested on British accent. This should be taken care of in future work, by ensuring that different accents are covered, for all models, but especially the English one.

- There is inter-speaker variability on the results, with the gender having significant effect on the results. For example, the Spanish model needs to be further trained, by using more female subjects.

7.3 Text analysis

In this section, we provide an evaluation of the UD-based and the Penn-Treebank-based analysis pipelines.

For the UD-based pipeline, we report on the evaluation of the Part-of-Speech tagging (Table 31, Table 33, Table 35, Table 37), on the syntactic dependency parsing (

Table 32, Table 34, Table 36, Table 38), and the deep analysis (Table 39). For this, we use the official UD test sets as provided in the CoNLL 2017 shared task[50]. For the evaluation of the deep analysis, we annotated manually about 900 deep tokens (~75 sentences) in English and Spanish at this point; further evaluations, including coverage for Italian and Greek will be carried out during the second half of the project. For the dependency relation assignment, we provide both the results of the parser only, using gold standard features, and of the whole pipeline, that is, using the features predicted by the previous modules, in order to reflect the accuracy in a real-life setting.

Table 31: Results of the evaluation of the UD-based PoS tagging (Greek)

PoS tag	Recall	Precision
NOUN	0.9321739	0.8355417
PUNCT	0.9313815	1.0
DET	1.0	0.96156085
ADJ	0.8653396	0.6798528
AUX	0.91150445	0.9809524
ADV	0.9111111	0.8523908
PART	0.9897698	1.0
SCONJ	0.962963	0.9017341
VERB	0.9043977	0.87108654
CCONJ	0.9498681	1.0
ADP	0.7048611	0.9902439
PRON	0.8898129	0.9861751
PROPN	0.48916408	0.8586956
X	0.33070865	0.56
NUM	0.67021275	0.9402985

Table 32: Results of the evaluation of the UD-based dependency parsing (Greek)

	UAS	LAS
Gold PoS, Lemma and Feats	84.03	77.61
Predicted PoS, Lemma and Feats	74.05	64.90

Table 33: Results of the evaluation of the UD-based PoS tagging (English)

PoS tag	Recall	Precision
PRON	0.9791474	0.97869384
SCONJ	0.7209302	0.8857143
PROPN	0.7817919	0.7143486
VERB	0.9197438	0.92217606
ADP	0.96432114	0.9284351
NOUN	0.9070668	0.8336299
PUNCT	0.9059884	0.9918929
CCONJ	0.9878214	0.9945504
ADV	0.87510204	0.9061707
ADJ	0.86296517	0.880651
DET	0.9836498	0.98468846

[50] http://universaldependencies.org/conll17/

AUX	0.9839465	0.9558155
PART	0.9873016	0.9242199
NUM	0.6604478	0.8962025
X	0.23021583	0.74418604
SYM	0.59782606	0.9166667
INTJ	0.75	0.9574468

Table 34: Results of the evaluation of the UD-based dependency parsing (English)

	UAS	LAS
Gold PoS, Lemma and Feats	83.79	79.65
Predicted PoS, Lemma and Feats	77.63	71.51

Table 35: Results of the evaluation of the UD-based PoS tagging (Spanish)

PoS tag	Recall	Precision
ADJ	0.908776	0.8142783
ADP	0.9974796	0.9975993
DET	0.9940178	0.9811865
PUNCT	0.8169148	0.99980617
NOUN	0.96965563	0.9229462
PROPN	0.8998785	0.8678228
VERB	0.94282407	0.8844734
NUM	0.8659044	0.9731308
CCONJ	0.99652535	0.9944522
PRON	0.9314602	0.94432455
ADV	0.9456776	0.9648391
AUX	0.91425043	0.94095236
SCONJ	0.91645986	0.93977946
SYM	0.8378378	1.0
PART	0.7222222	0.8125
INTJ	0.53846157	1.0

Table 36: Results of the evaluation of the UD-based dependency parsing (Spanish)

	UAS	LAS
Gold PoS, Lemma and Feats	85.62	80.69
Predicted PoS, Lemma and Feats	79.65	73.73

Table 37: Results of the evaluation of the UD-based PoS tagging (Italian)

PoS tag	Recall	Precision
VERB	0.9321267	0.91049725
DET	0.9953271	0.9861111
PROPN	0.819802	0.82965934
PUNCT	0.8212766	1.0
AUX	0.9876543	0.9546539
NOUN	0.9671339	0.9191548
ADP	0.9939136	0.99512494
ADJ	0.9045454	0.78552634
PRON	0.9393204	0.94621027

SCONJ	0.8181818	0.92045456
ADV	0.9326683	0.9396985
CCONJ	1.0	1.0
NUM	0.877193	0.97402596
X	0.3846154	0.8333333
SYM	1.0	1.0

Table 38: Results of the evaluation of the UD-based dependency parsing (Italian)

	UAS	LAS
Gold PoS, Lemma and Feats	88.00	83.25
Predicted PoS, Lemma and Feats	82.03	75.95

Table 39: Results of the evaluation of the UD-based deep graph-transduction grammars

	LAS
English	79.83
Spanish	67.28

In the following we report the evaluation results for the English-specific, PennTreebank-based analysis pipeline; respective evaluations for Spanish-specific analysis will be carried out during the second half of the project, as part of the investigations into performance trade-off between using generic versus language-specific analysis. More specifically, we report the numbers of the MATE tools parser, which assigns jointly lemmas, parts of speech, morphological features, and dependencies (Table 40). For the analysis of the deep analysis, we annotated manually a gold standard of about 300 sentences (5,000 deep tokens); the precision and recall of the hypernode identification (Table 41) and the labeled and unlabeled attachment scores are provided (Table 42). A formal evaluation of the PredArg structures has not been carried out at this point.

Table 40: Results of the evaluation of the PTB-based joint parsing

	UAS	LAS
English	93.67	92.68

Table 41: Results of the evaluation of hypernode identification

	Precision	Recall
English	97.00	99.96

Table 42: Results of the evaluation of the deep-syntactic graph-transduction grammars

	UAS	LAS
English	96.74	91.24

The results of the evaluation of the English-specific pipeline are better than that of the generic UD-based pipeline, but it remains to be seen if this has an actual impact on the overall performance of the analysis component in beAWARE.

The aforementioned evaluations have been carried out using the typical methodology for assessing the performance of dependency parsing, namely in terms of the similarity of the predicted output with a given gold tree in absolute terms, meaning that all errors are counted equal. However, from a linguistic point of view, errors may be less or more severe depending on the predicted labels and the misplacement of branches. For example, confusing a direct object with an indirect object may not be as severe as confusing it with the subject, and hanging a branch far away from its right position may be more grave than hanging it closer.

To address this issue, UPF has been carrying out extensive experiments on linguistics-oriented evaluation of dependency parser. To this end, we extended the two most commonly used dependency parsing metrics, namely Unlabeled Attachment Score --UAS-- and Labeled Attachment Score --LAS--, with penalization coefficients based on linguistically motivated relation hierarchies and on relation importance, and incorporated the notion of distance between the gold and the predicted heads. Thus, 46 new different metrics were proposed, by both combining the different coefficients with UAS and LAS, and by themselves.

With the objective of studying the effect of the different linguistic hypothesis on the evaluation, and with the major goal of assessing which evaluation metric is more indicative of the quality of the dependency parsing, we have conducted both an intrinsic an extrinsic evaluation to see which of the intrinsic metrics correlates better with the extrinsic ones.

We first conduct an intrinsic evaluation, relying on the available data from the CoNLL 2017 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies. We compute UAS, LAS and the proposed new metrics on a subset of the system outputs for English, Spanish and French languages, composed of 12,713, 35,758 and 6,111 tokens respectively. We also conduct an extrinsic evaluation. As no direct downstream application is available, we decided to use the UD-based deep structures. We manually generated deep parsing structures for a subset of the sentences of the CoNLL 2017 system outputs to constitute the gold standard, containing 943, 930 and 938 tokens respectively. Then, we fed the correspondent surface parsing structures into a deep parser, producing deep parsing structures for each candidate system, which were then evaluated with the gold standard generating deep parsing metrics.

To evaluate the results, we are currently studying which of the surface dependency parsing metrics correlated better with the deep recall LAS metric. The results will be reported in the final deliverable.

8 Conclusions and Next Steps

8.1 Conclusions

This deliverable reports on the basic methodological framework for image/video, audio and text analysis including techniques for named entities recognition, concept extraction from textual content, concept linking and relations, speech recognition and multimedia concept detection. The document relies on the work that have done in the first seventeenth months of the beAWAREproject in the Tasks 3.2 and 3.3. It serves the needs and requirements as they have determined in the aforementioned section 2 and extensively described in Deliverable D2.1. Furthermore, the progress of the work that has been made in this period met the initial objectives of the current WP3 and is aligned with the beAWARE's general objectives. Specifically, we have already developed and deployed interoperable modules for multilingual text analysis, for multimedia (image/video) analysis and for automatic speech recognition analysis. These modules interact with the beAWARE framework and support the decision making process for addressing extreme crisis events. However, this effort need to be refined and enhanced as described in the following section.

8.2 Next Steps

8.2.1 Image and video analysis

Most of the visual analysis components that we described previously currently incorporate basic approaches so as to extract and deliver valuable information to the beAWARE decision support system. In order to deal with more challenging scenarios and to provide more solid analysis results we will continue to enchance our methods and techniques and then evaluate them as well so as to track the progress.

For the modules related to fire and flood detection from social media images, namely the EmC, EmL and ObD, the next steps include further and more customized training of the deep CNN architectures that are deployed, with more data related to fire and flood scenarios, so as to better tailor our algorithms to respond well in those types of challenges. More specifically, an approach for object detection based on occluded models would greatly aid that purpose since most of the targets we are trying to detect are occluded from flood or fire pixels. In order to get better segmentation results from the EmL, a bigger dataset will be compiled to train the Deeplab model. Moreover, to properly extract severety levels according to CAP, more sophisticated rules will be designed.

For the modules of the task related to fire and flood detection from video samples we plan to explore the option of evaluating the deep architectures that were used in 3.1.1 . To

modify them accordingly for video analysis, a keyframe extractor will be implemented that will select the most representative video frames and the analysis will be performed in or arround those moments. The algorithms are expected to get a speed boost as well.

For our traffic analysis and management schemes, we will continue to explore better techniques for automatic camera calibration so as to improve our speed estimation results. In addition, we will also create and evaluate a sophisticated describtor, that will encode motion information patterns from the vehicles found in the scene based on optical flow. A new model will be then trained in order to estimate traffic levels of highways based on motion patterns.

Finally, new components that will incorporate some or all of the aforementioned techniques will be gradually constructed in order to analyse footage from UAVs. The algorithms will be properly modified and new models will be trained in order to accommodate the analysis of UAV captured footage. Additionally, some techniques such as the EmL flood segmentation model will be evaluated in real time camera feed from a dedicated static camera that will be placed in Vicenza so as to monitor water levels in a critical city location and warn about possible flooding situations.

8.2.2 Audio analysis

In order to improve recognition accuracy and make speech recognition more robust CERTH will continue adapting acoustic models, as new recordings become available. We will also continue expanding dictionaries in order to include all possible location names, keywords and missing words. Future work will also focus on the implementation of more advanced denoising techniques in order to improve recognition accuracy in noisy environemts and on more advanced automatic punctuation techniques, in order to facilitate the textual analysis. Finally, regarding the English model, more emphasis should be given to the different accents, in order to ensure that all accents are covered.

8.2.3 Text analysis

As far as text analysis concerned, and as already sketched above, next steps include, among others, the incorporation of a more elaborate strategy for location mentions candidate selection and their disambiguation using OpenStreetMap data as the underlying reference knowledge base. Semantic abstraction, intra- and across the different languages is another direction, along which investigations have already commenced via the integration of linking against BabelNet, and the preliminary work into the definition of a common reference conceptual structure model to which the predArg representations extracted in different languages will be mapped. Moreover, we will continue the ongoing study on new parsing evaluation metrics, so as to improve the correlation between intrinsic parser evaluation and

the analysis accuracy in downstream applications, and eventually identify the best parsing framework for the purpose of beAWARE. In addition, enhancements to meet the scope and coverage as entailed by the incremental addition of new use cases for the planned pilots will be catered for as needed, including tweet normalization and adaptations for spoken language parsing. Last but not least, we will continue working on the compilation of annotated copora to be used as training data for the planned statistical parsing (and generation, as will be described in D5.4) investigations.

9 References

Hinton, G., Deng, L., Yu, D., Mohamed, A.-r., Jaitly, N., Senior, A., . . . Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82 - 97.

[Morgan, N., Zhu, Q., Stolcke, A., Sönmez, K., Sivadas, S., Shinozaki, T., . . . Athineos, M. (2005). Pushing the envelope—Aside [speech recognition]. *IEEE SIGNAL PROCESSING MAGAZINE [, 22*(5), 81 - 88.

Abdulla, W., Chow, D., & Sin, G. (2003). Cross-words reference template for DTW-based speech recognition systems. *Conference on Convergent Technologies for Asia-Pacific Region. 4.* Bangalore: IEEE.

Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., & Tao, K. (2012). Semantic + filtering + search = Twitcident. Exploring information in social web streams. *Hypertext*, (pp. 285–294).

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2012, Nov). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(11), 2274-2282.

Ahmad, K., Konstantin, P., Riegler, M., Conci, N., & Holversen, P. (2017). Cnn and gan based satellite and social media data fusion for disaster detection. *Working Notes Proc. MediaEval Workshop*, (p. 2).

Ahmad, S., Ahmad, K., Ahmad, N., & Conci, N. (n.d.). Convolutional Neural Networks for Disaster Images Retrieval.

Ahonen, T., Hadid, A., & Pietikainen, M. (2006, Dec). Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2037-2041.

Alam, M., Recupero, D., Mongiovì, M., Gangemi, A., & Ristoski, P. (2017). Event-based knowledge reconciliation using frame embeddings and frame similarity. *Knowl.-Based Syst, 135*, 192-203.

Al-Olimat, H. S., Thirunarayan, K., Shalin, V. L., & Sheth, A. P. (2017). Location Name Extraction from Targeted Text Streams using Gazetteer-based Statistical Language Models. *CoRR, abs/1708.03105*.

Amin, T. B., & Mahmood , I. (2008). Speech Recognition Using Dynamic Time Warping. *2nd International Conference on Advances in Space Technologies. 2*, pp. 74-79. Islamabad: Proceedings of ICAST.

Anusuya, M. A., & Katti, S. K. (2009). Speech Recognition by Machine: A Review. *International Journal of Computer Science and Information Security, 6*(3), 181-205.

Aparicio, J., Taulé, M., & Martí, M. (2008). AnCora-Verb: A lexical resources for the semantic annotation of corpora. *International Conference on Language Resources and Evaluation (LREC).* Marrakesh, Morocoo.

Augenstein, I., Pado, S., & Rudolph, S. (2012). LODifier: Generating Linked Data form unstructured text. *Proceedings of ESWC'12* (pp. 210-224). Springer.

Avalhais, L. P., Rodrigues, J., & Traina, A. J. (2016). Fire detection on unconstrained videos using color-aware spatial modeling and motion flow. *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, (pp. 913-920).

Avgerinakis, K., Briassouli, A., & Kompatsiaris, Y. (2016). Activity detection using Sequential Statistical Boundary Detection (SSBD). *Computer Vision and Image Understanding, 144*, 46-61.

Avgerinakis, K., Giannakeris, P., Briassouli, A., Karakostas, A., Vrochidis, S., & Kompatsiaris, I. (2017). Intelligent traffic city management from surveillance systems (CERTH-ITI). *IEEE Smart World 2017, NVIDIA AI city challenge.*

Avgerinakis, K., Giannakeris, P., Briassouli, A., Karakostas, A., Vrochidis, S., & Kompatsiaris, I. (2017). LBP-flow and hybrid encoding for real-time water and fire classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 412-418).

Avgerinakis, K., Moumtzidou, A., Andreadis, S., Michail, E., Gialampoukidis, I., Vrochidis, S., & Kompatsiaris, I. (2017). Visual and textual analysis of social media and satellite images for flood detection@ multimedia satellite task MediaEval 2017. *Multimedia satellite task MediaEval 2017.*

Bilmes, J. A. (2006). What HMMs can do. *IEICE - Transactions on Information and Systems, E89-D*(3), 869-891.

Bischke, B., Bhardwaj, P., Gautam, A., Helber, P., Borth, D., & Dengel, A. (2017). Detection of flooding events in social multimedia and satellite imagery using deep neural networks. *Working Notes Proc. MediaEval Workshop*, (p. 2).

Bohnet, B., & Nivre, J. (2012). A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. *EMNLP-CoNLL*, (pp. 1455-1465). Jeju Island Korea.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M., Maynard, D., & Aswani, N. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. *RANPL*, (pp. 83-90).

Bosco et al. (2000). Building a Treebank for Italian: a Data-driven Annotation Schema., (pp. 99-105). Athens.

Buchholz, S., & Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006*, (pp. 149--164). New York City.

Burton, S., Tanner, K., Giraud-Carrier, C., West, J., & Barnes, M. (2012). "Right Time, Right Place" Health Communication on Twitter: Value and Accuracy of Location Information. *Journal of Medical Internet Research, 14*(6).

Celik, T., & Demirel, H. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal, 44*, 147-158.

Chan, A. B., & Vasconcelos, N. (2008, May). Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(5), 909-926.

Chan, A. B., & Vasconcelos, N. (2009). Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(10), 1862-1879.

Chen, C.-S., Yeh, C.-W., & Yin, P.-Y. (2009). A novel Fourier descriptor based image alignment algorithm for automatic optical inspection. *Journal of Visual Communication and Image Representation, 20*, 178-189.

Chen, J., Zhao, G., Salo, M., Rahtu, E., & Pietikainen, M. (2013). Automatic dynamic texture segmentation using local descriptors and optical flow. *IEEE Transactions on Image Processing, 22*(1), 326-339.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *CoRR, abs/1606.00915*. Retrieved from http://arxiv.org/abs/1606.00915

Cheng, K. W., Chen, Y. T., & Fang, W. H. (2015, 12). Gaussian Process Regression-Based Video Anomaly Detection and Localization With Hierarchical Feature Representation. *IEEE Transactions on Image Processing, 24*, 5288-5301.

Chino, D. Y., Avalhais, L. P., Rodrigues, J. F., & Traina, A. J. (2015). Bowfire: detection of fire in still images by integrating pixel color and texture analysis. *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*, (pp. 95-102).

Corcoglioniti, F., Rospocher, M., & Aprosio, A. (2016). A 2-phase Frame-based Knowledge Extraction Framework. *Proc. of ACM Symposium on Applied Computing*, (pp. 354-361). Pisa.

Daiber, J., Jakob, M., Hokamp, C., & Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. *I-SEMANTICS*, (pp. 121-124).

Dao, M. S., Pham, Q. N., Nguyen, D., & Tien, D. (2017). A domain-based late-fusion for Disaster Image Retrieval from Social Media.

de Souza, C. R., Gaidon, A., Vig, E., & López, A. M. (2016). Sympathy for the Details: Dense Trajectories and Hybrid Classification Architectures for Action Recognition. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision -- ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11--14, 2016, Proceedings, Part VII* (pp. 697-716). Cham: Springer International Publishing.

Del Gratta, R. (2015). Converting the PAROLE SIMPLE CLIPS lexicon into RDF with lemon. *Semantic Web, vol. 6, no. 4*, 387-392.

Deng, L., & Li, X. (2013). Machine Learning Paradigms for Speech Recognition: An Overview. *IEEE Transactions on Audio, Speech, and Language Processing, 21*(5), 1060-1089.

Derczynski, L., Ritter, A., Clark, S., & Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. *RANLP*, (pp. 198-206).

Derpanis, K. G., Lecce, M., Daniilidis, K., & Wildes, R. P. (2012, June). Dynamic scene understanding: The role of orientation features in space and time in scene classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1306-1313).

Dimitropoulos, K., Barmpoutis, P., Kitsikidis, A., & Grammalidis, N. (2017). Classification of Multidimensional Time-Evolving Data using Histograms of Grassmannian Points. *IEEE Transactions on Circuits and Systems for Video Technology, PP*(99), 1-1.

Doretto, G., Chiuso, A., Wu, Y. N., & Soatto, S. (2003, Feb). Dynamic Textures. *International Journal of Computer Vision, 51*(2), 91-109.

Dubois, S., Peteri, R., & Menard, M. (2015, May). Characterization and recognition of dynamic textures based on the 2D+T curvelet transform. *Signal, Image and Video Processing, 9*(4), 819-830.

Dubská, M., Herout, A., & Sochor, J. (2014). Automatic Camera Calibration for Traffic Understanding. *BMVC*, *4*, p. 8.

Dubská, M., Herout, A., Juránek, R., & Sochor, J. (2015). Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems, 16*, 1162-1171.

Farzindar, A. K. (2015). A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence, 31*(1), 132-164.

Feichtenhofer, C., Pinz, A., & Wildes, R. P. (2014, June). Bags of Spacetime Energies for Dynamic Scene Recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Filipiak, P., Golenko, B., & Dolega, C. (2016). NSGA-II Based Auto-Calibration of Automatic Number Plate Recognition Camera for Vehicle Speed Measurement. *European Conference on the Applications of Evolutionary Computation*, (pp. 803-818).

Foster, J., Çetinoglu, Ö., Wagner, J., Le Roux, J., Hogan, S., Nivre, J., . . . Van Genabith, J. (2011). #hardtoparse: POS tagging and parsing the Twitterverse. *5th AAAI Conference on Analyzing Microtext.*

Fotopoulou, A., & al., e. (2014). Encoding MWEs in a conceptual lexicon. *Worhshop on Multi-word Expressions (MWE)*, (pp. 43-47).

Fritz, M., Leibe, B., Caputo, B., & Schiele, B. (2005). Integrating representative and discriminant models for object category detection. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, *2*, pp. 1363-1370.

Furui, S. (1991). Vector-quantization-based speech recognition and speaker recognition techniques. *Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems and Computers. 2*, pp. 954-958. IEEE.

Gangemi, A., & Presutti, V. (2009). Ontology Design Patterns. In S. S. Studer (Ed.). (pp. 221-243). Springer Verlag.

Gangemi, A., Presutti, V., Recupero, D., Nuzzolese, A. D., & Mongiovì, M. (2017). Semantic Web Machine Reading with FRED. *Semantic Web, 8*(6), 873-893.

García-Miguel, J., & al., e. (2010). ADESSE. A Database with syntactic and semantic annotation of a corpus of Spanish. *International Conference on Language Resources and Evaluation (LREC).* Valetta, Malta.

Gelernter, J., & Mushegian, N. (2011). Geo-parsing Messages from Microtext. *Transactions in GIS, 15*(6), 753–773.

Ghaemmaghami, M. P., Razzazi, F., Sameti, H., Dabbaghchian, S., & BabaAli, B. (2009). Noise reduction algorithm for robust speech recognition using MLP neural network. *Asia-Pacific Conference on Computational Intelligence and Industrial Applications. 1*, pp. 377-380. Wuhan: IEEE.

Giouli, V., & Fotopoulou, A. (2012). Emotion verbs in Greek. From lexicon-grammar tables to multi-purpose syntactic and semantic lexica. *15th EURALEX International Congress.* Oslo, Norway.

Girshick, R. (2015). Fast r-cnn. *arXiv preprint arXiv:1504.08083*.

Goutsos, D. (2010). The Corpus of Greek texts: a reference corpus for modern Greek. *Corpora, 5*(1), 29-44.

Graham, M., Hale, S., & Gaffney, D. (2014). Where in the World Are You? Geolocation and Language Identification in Twitter. *The Professional Geographer, 66*(4), 568–578.

Grossberg, S., & Huang, T.-R. (2009). ARTSCENE: A neural system for natural scene classification. *Journal of Vision, 9*(4), 6.

Hatzigeorgiu, N. et al. (2000). Design and Implementation of the Online ILSP Greek Corpus. *LREC.*

He, X. C., & Yung, N. H. (2007). A novel algorithm for estimating vehicle speed from two consecutive images. *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*, (pp. 12-12).

He, X., Deng, L., & Chou , W. (2006). A Novel Learning Method for Hidden Markov Models in Speech and Audio Processing. *IEEE 8th Workshop on multimedia signal processing.* Victoria.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*, 583-596.

Hospedales, T., Gong, S., & Xiang, T. (2012, 7 01). Video Behaviour Mining Using a Dynamic Topic Model. *International Journal of Computer Vision, 98*, 303-323.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., . . . others. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *IEEE CVPR.*

Hughes, A., & Leysia, P. (2009). Twitter adoption and use in mass convergence and emergency events. *Int. Journal of Emergency Management, 6*(3), 248–260.

Ikawa, Y., Enoki, M., & Tatsubori, M. (2012). Location inference using microblog messages. *Proc. of WWW. ACM*, 687–690.

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015). Processing Social Media Messages in Mass Emergency: A Survey. *ACM Comput. Surv. , 47*(4), 67:1--67:38.

Inkpen, D., Liu, J., Farzindar, A., Kazemi, F., & Ghazi, D. (2017). Location detection and disambiguation from twitter messages. *J. Intell. Inf. Syst., 49*(2), 237-253.

Jadhav, A. V., & Pawar, R. V. (2012). Review of various approaches towards speech recognition. *International Conference on Biomedical Engineering (ICoBE)* (pp. 27-28). Penang: IEEE.

Jeong, H., Yoo, Y., Yi, K. M., & Choi, J. Y. (2014, 8 01). Two-stage online inference model for traffic pattern analysis and anomaly detection. *Machine Vision and Applications, 25*, 1501-1517.

Ji, H., Yang, X., Ling, H., & Xu, Y. (2013, Jan). Wavelet Domain Multifractal Analysis for Static and Dynamic Texture Classification. *IEEE Transactions on Image Processing, 22*(1), 286-299.

Jiang, F., Yuan, J., Tsaftaris, S. A., & Katsaggelos, A. K. (2011). Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding, 115*, 323-333.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In H.-J. K. Joakim Nivre (Ed.), *NODALIDA 2007 Proceedings*, (pp. 105-112). Tartu.

Kaltsa, V., Briassouli, A., Kompatsiaris, I., Hadjileontiadis, L. J., & Strintzis, M. G. (2015, July). Swarm Intelligence for Detecting Interesting Events in Crowded Environments. *IEEE Transactions on Image Processing, 24*(7), 2153-2166.

Kamp, H., & Reyle, U. (1993). *From Discourse to Logic.* Dordrecht: Kluwer Academic Publishers.

Karpagavalli, S., & Chandra, E. (2016). A Review on Automatic Speech Recognition Architecture and Approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 9*(4), 393-404.

Kingsbury, P., & Palmer, M. (2002). From TreeBank to PropBank. *International Conference on Language Resources and Evaluation (LREC).* Las Palmas.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., & Smith, N. (2014). A Dependency Parser for Tweets. *EMNLP*, (pp. 1001–1012).

Kuettel, D., Breitenstein, M. D., Gool, L. V., & Ferrari, V. (2010, 6). What's going on? Discovering spatio-temporal dependencies in dynamic scenes. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 1951-1958).

Lan, J., Li, J., Hu, G., Ran, B., & Wang, L. (2014). Vehicle speed measurement based on gray constraint optical flow algorithm. *Optik-International Journal for Light and Electron Optics, 125*, 289-295.

Lenci, A., Montemagni, S., Venturi, G., & Cutrulla, M. G. (2012). Enriching the ISST-TANL Corpus with Semantic Frames. *LREC*, (pp. 3719-3726).

Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., & Lee, B.-S. (2012). TwiNER: named entity recognition in targeted twitter stream. *In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (pp. 721-730). Association for Computing Machinery (ACM).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, (pp. 740-755).

Lingad, J., Karimi, S., & Yin, J. (2013). Location Extraction from Disaster-related Microblogs. *WWW '13 22nd International World Wide Web Conference IW3C2*, 1017–1020.

Lishuang, Z., & Zhiyan, H. (2010). Speech Recognition System Based on Integrating Feature and HMM. *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, *3*, pp. 449-452.

Liu, L., Zhao, L., Long, Y., Kuang, G., & Fieguth, P. (2012). Extended local binary patterns for texture classification . *Image and Vision Computing , 30*(2), 86-99.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *European conference on computer vision*, (pp. 21-37).

Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., & Smith, N. A. (2018). Parsing Tweets into Universal Dependencies. *CoRR, abs/1804.08228*.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 3431-3440).

Lopez-Fuentes, L., Weijer, J., Bolaños, M., & Skinnemoen, H. (n.d.). Multi-modal Deep Learning Approach for Flood Detection.

Lyding, V. et al. (2014). The PAISÀ Corpus of Italian Web Texts. *Proceedings of the 9th Web as Corpus Workshop (WaC-9), Association for Computational Linguistics* , (pp. 36-43). Gothenburg.

MacEachren, A. M., Jaiswal, A., Robinson, A. C., Pezanowski, S., Savelyev, A., Mitra, P., . . . Blanford, J. (2011). Senseplace2: Geotwitter analytics support for situational awareness. *In Proc. of VAST. IEEE*, (pp. 181–190).

Manning, C. D., Surdeanu, M., & Baue, J. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguis: System Demonstrations*, (pp. 55-60).

Marimon, M., & Bel, N. (2015). Dependency structure annotation in the IULA Spanish LSP Treebank. *Language Resources and Evaluation, 49*(2), 433-454.

Marszalek, M., Laptev, I., & Schmid, C. (2009, June). Actions in context. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2929-2936).

Martínez Alonso, H., & Zeman, D. (2016). Universal Dependencies for the Ancora Treebanks. *Procesamiento del Lenguaje Natural*(57), 91-98.

McDonald, R., Lerman, K., & Pereira, F. (2006). Multilingual Dependency Parsing with a Two-Stage Discriminative Parser. *Tenth Conference on Computational Natural Language Learning (CoNLL-X) .*

Mel'cuk, I. (1988). *Dependency syntax.* State University of New York Press, Albany, NY.

Mettes, P., Tan, R. T., & Veltkamp, R. C. (2017). Water detection through spatio-temporal invariant descriptors. *Computer Vision and Image Understanding, 154*, 182-191.

Mettes, P., Tan, R. T., & Veltkamp, R. C. (2017). Water detection through spatio-temporal invariant descriptors. *Computer Vision and Image Understanding, 154*, 182-191.

Meyers, A., MacLeod, C., Szekely, R., Zelinska, V., & Young, B. G. (2004). The NomBank project: An interim report. *HLT-NAACL Workshop on Frontiers in corpus annotation.* Boston, US.

Mille, S., & Wanner, L. (2010). Syntactic Dependencies for Multilingual and Multilevel Corpus Annotation. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10).* Valetta.

Montemagni et al. (2003). The syntactic-semantic treebank of Italian. An overview. *Linguistica Computazionale XVI-XVII*, 461-492.

Mumtaz, A., Coviello, E., Lanckriet, G. R., & Chan, A. B. (2013, July). Clustering Dynamic Textures with the Hierarchical EM Algorithm for Modeling Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(7), 1606-1621.

Mumtaz, A., Coviello, E., Lanckriet, G. R., & Chan, A. B. (2015, April). A Scalable and Accurate Descriptor for Dynamic Textures Using Bag of System Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(4), 697-712.

Nadeau, D., & Sekine, S. (2009). A survey of Named Entity recognition and classification. *In SekineS.RanchhodE. (Eds.), Named entities: Recognition, classification and use*, 3-25.

Navigli, R., & Ponzetto, S. (2010). BabelNet: Building a very large multilingual semantic network. *48th Annual meeting of the Association for Computational Linguistics (ACL).* Uppsala, Sweden.

Navigli, R., & Ponzetto, S. (2012, December). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence. *Artificial Intelligence, 193*, 217-250.

Navigli, R., Camacho-Collados, J., & Raganato, A. (2017). Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. *EACL (1)*, (pp. 99-110).

Nister, D., & Stewenius, H. (2006). Scalable Recognition with a Vocabulary Tree. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, *2*, pp. 2161-2168.

Nivre,J., et al. . (2016). Universal Dependencies v1: A Multilingual Treebank Collection. *Proceedings of LREC.* Portoroz.

Nogueira, K., Fadel, S. G., Dourado, Í. C., Werneck, R. d., Muñoz, J. A., Penatti, O. A., . . . Torres, R. d. (n.d.). Data-Driven Flood Detection using Neural Networks.

Nurhadiyatna, A., Hardjono, B., Wibisono, A., Sina, I., Jatmiko, W., Ma'sum, M. A., & Mursanto, P. (2013). Improved vehicle speed estimation using Gaussian mixture model and hole filling algorithm. *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, (pp. 451-456).

Oliva, A., & Torralba, A. (2001, May). Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision, 42*(3), 145-175.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, N. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. *HLTCon*, (pp. 380-390).

Paliwal K.K. (1990). Neural net classifiers for robust speech recognition under noisy environments. *International Conference on Acoustics, Speech, and Signal Processing*, *1*, pp. 429-432. Albuquerque.

Palmero, A., & Moretti, G. (2016). Italy goes to Stanford: a collection of CoreNLP modules for Italian. *CoRR, abs/1609.06204.*

Papageorgiou, H., & al., e. (2006). Adding multi-layer semantic to the Greek Dependency Treebank. *LREC.* Genoa, Italy.

Patel , I., & Rao, Y. S. (2010). Speech Recognition Using Hidden Markov Model with MFCC-Subband Technique. *International Conference on Recent Trends in Information, Telecommunication and Computing (ITC)*, (pp. 168-172).

Peris, A., & Taulé, M. (2011). AnCora-Nom: A Spanish lexicon for deverbal nominalizations. *Procesamiento del Lenguage Natural, vol. 46*, 11-18.

Petasis, G., & al., e. (2001). A Greek morphological lexicon and its exploitation by natural language processing applications. *Panhellenic Conference on Informatics*, (pp. 401-419).

Peteri, R., Fazekas, S., & Huiskes, M. J. (2010). DynTex: A comprehensive database of dynamic textures. *Pattern Recognition Letters, 31*(12), 1627-1632.

Piciarelli, C., Micheloni, C., & Foresti, G. L. (2008, 11). Trajectory-Based Anomalous Event Detection. *IEEE Transactions on Circuits and Systems for Video Technology, 18*, 1544-1554.

Povey, D., & Ghoshal, A. (2011). The Kaldi Speech Recognition Toolkit. *Proceedings of ASRU*, (pp. 1-4). Hawaii .

Prokopidis et al. (2005). Theoretical and Practical Issues in the Construction of a Greek Dependency Treebank. In S. K. Montserrat Civit (Ed.), *Prokopidis et al,. Theoretical and Practical Issues in the CoProceedings of The Fourth Workshop on Treebanks and Linguistic Theories*, (pp. 149-160). Barcelona.

Prokopidis, P., & Papageorgiou, H. (2017). Universal Dependencies for Greek. *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, (pp. 102-106). Gothenburg.

Purohit, H., & Sheth, A. (2013). Twitris v3: From citizen sensing to analysis, coordination and action. *ICWSM.*

Qi, X., Li, C.-G., Zhao, G., Hong, X., & Pietikäinen, M. (2016). Dynamic texture and scene classification by transferring deep image features. *Neurocomputing, 171*, 1230-1241.

Qian, X., Hua, X.-S., Chen, P., & Ke, L. (2011). PLBP: An effective local binary patterns texture descriptor with pyramid representation. *Pattern Recognition, 44*(10), 2502-2515.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, (pp. 91-99).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, (pp. 91-99).

Ritter, A., Clark, S., Mausam, & Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. *EMNLP*, (pp. 1524-1534).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, (pp. 234-241).

Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., . . . Bogaard.T. (2016). Building event-centric knowledge graphs from news. *J. Web Sem*, 132-151.

Rossi, L., Akhloufi, M., & Tison, Y. (2011). On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. *Fire Safety Journal, 46*, 9-20.

Rudz, S., Chetehouna, K., Hafiane, A., Laurent, H., & Séro-Guillaume, O. (2013). Investigation of a novel image segmentation method dedicated to forest fire applications. *Measurement Science and Technology, 24*, 075403.

Ruimy, N., & al., e. (2002). CLIPS, a multi-level Italian computational lexicon: a glimpse to data. *LREC.* Palmas, spain.

Saleemi, I., Shafique, K., & Shah, M. (2009, 8). Probabilistic Modeling of Scene Dynamics for Applications in Visual Surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*, 1472-1485.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation, 13*, 1443-1471. doi:10.1162/089976601750264965

Schuler, K. (2005). *VerbNet: A broad coverage, comprehensive verb lexicon.* Pennsylvania, US: University of Pennsylvnia.

Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing, 27*(6), 803-816.

Sharnagat, R. (2014). *Named entity recognition: A literature survey.* Bombay: Indian Institute of Technology.

Shroff, N., Turaga, P., & Chellappa, R. (2010, June). Moving vistas: Exploiting motion for describing scenes. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 1911-1918).

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR, abs/1409.1556*. Retrieved from http://arxiv.org/abs/1409.1556

Smith, J. R., Lin, C.-Y., & Naphade, M. (2002). Video texture indexing using spatio-temporal wavelets. *Proceedings. International Conference on Image Processing*, *2*, pp. II-437-II-440 vol.2.

Smith, N., & Gales, M. J. (2002). Using SVMS and discriminative models for speech recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, *1*, pp. 77-80. Proc. ICASSP, vol. 1, (2002), pp.77 -80.

Sochor, J., Juránek, R., & Herout, A. (2017). Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding, 161*, 87-98.

Stolcke, A., Luciana Ferrer, L., Kajarekar, S., Shriberg, E., & Venk, A. (2005). MLLR transforms as features in speaker recognition. *Proceedings of the 9th European Conference on Speech Communication and Technology*, (pp. 2425-2428).

Taulé, M., & al., e. (2011). AnCora-Net: Integración multilingüe de recursos lingüísticos semánticos. *Procesamiento del Lenguaje Natural, vol. 47*, 153-160.

Taulé, M., Martí, M., & Recasens, M. (2008). Ancora: Multilevel Annotated Corpora for Catalan and Spanish. *Taulé, M., M.A. Martí, M. Recasens "AnProceedings of 6th International Conference on Language Resources and Evaluation.* Marrakesh.

Ting, C.-M., Salleh, S.-H., Tan, T.-S., & Ariff, A. K. (2007). Text independent Speaker Identification using Gaussian mixture model. *International Conference on Intelligent and Advanced Systems* (pp. 194-198). Kuala Lumpur: IEEE.

Tonon, A., Cudré-Mauroux, P., Blarer, A., Lenders, V., & Motik, B. (2017). ArmaTweet: Detecting Events by Semantic Tweet Analysis. *ESWC (2)*, (pp. 138-153).

Tzortzi, K., & Markantonatou, S. (2014). Development of a conceptual lexicon with ontological techniques. *Terminology and Ontology: Theories and Applications.* Chambery, France.

Ueffing, N., Bisani, M., & Vozila, P. (2013). Improved models for automatic punctuation prediction for spoken and written text. *14th Annual Conference of the International Speech Communication Association INTERSPEECH*, (pp. 3097-3101). Lyon.

Usbeck, R., Ngonga Ngomo, A.-C., Röder, M., Gerber, D., Coelho, S. A., Auer, S., & Both, A. (2014). AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. *ECAI*, (pp. 1113-1114).

Varadarajan, J., Emonet, R., & Odobez, J.-M. (2013, 5 01). A Sequential Topic Model for Mining Recurrent Activities from Long Term Video Logs. *International Journal of Computer Vision, 103*, 100-126.

Vieweg, S., Hughes, A., Starbird, K., & Palen, L. (2010). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. *CHI Conference on Human Factors in Computing Systems*, (pp. 1079-1088). Atlanta.

Wang, L., & He, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition, 23*(8), 905-910.

Wang, X., Ma, X., & Grimson, W. E. (2009, 3). Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*, 539-555.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., . . . Lyu, S. (2015). Detrac: A new benchmark and protocol for multi-object tracking. *arXiv preprint arXiv:1511.04136*.

Xu, Y., Huang, S., Ji, H., & FermГOller, C. (2012). Scale-space texture description on SIFT-like textons. *Computer Vision and Image Understanding, 116*(9), 999-1013.

Xu, Y., Quan, Y., Zhang, Z., Ling, H., & Ji, H. (2015). Classifying dynamic textures via spatiotemporal fractal analysis. *Pattern Recognition, 48*(10), 3239-3248.

Yang, W., Gao, Y., & Cao, L. (2013). TRASMIL: A local anomaly detection framework based on trajectory segmentation and multi-instance learning. *Computer Vision and Image Understanding, 117*, 1273-1286.

Yantorno, R. E., Iyer, A. N., & Sha, J. K. (2004). Usable speech detection using a context dependent Gaussian mixture model classifier. *Proceedings of the International Symposium on Circuits and Systems. 5*, pp. V-619- V-623. Vancouver: IEEE.

Zaharia, T., Segarceanu, S., & Cotescu, M. (2010). Quantized Dynamic Time Warping (DTW) algorithm. *8th International Conference on Communications (COMM)* (pp. 91-94). Bucharest: IEEE.

Zhang, J., Wang, X., & Lv, M. (2013). Flame image segmentation algorithm based on background subtraction. *PIAGENG 2013: Image Processing and Photonics for Agricultural Engineering*, *8761*, p. 876112.

Zhao, G., & Pietikainen, M. (2006). Local Binary Pattern Descriptors for Dynamic Texture Recognition. *18th International Conference on Pattern Recognition (ICPR'06)*, *2*, pp. 211-214.

Zhao, G., Ahonen, T., Matas, J., & Pietikainen, M. (2012, April). Rotation-Invariant Image and Video Description With Local Binary Pattern Features. *IEEE Transactions on Image Processing, 21*(4), 1465-1477.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, (pp. 487-495).

Zulfiqar, A., Muhammad , A., & Enriquez, M. (2009). A Speaker Identification System using MFCC Features with VQ Technique. *Third International Symposium on Intelligent Information Technology Application* (pp. 115-118). Shanghai: IEEE.