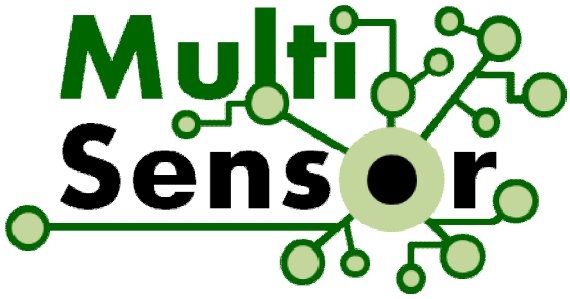

MULTISENSOR

Mining and Understanding of multilinguaL contenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation

FP7-610411

D3.1

Report on context representation and experimental findings

Dissemination level:	Public
Contractual date of delivery:	Month 12, 31/10/2014
Actual date of delivery:	Month 12, 31/10/2014
Workpackage:	WP3 User and context-centric content analysis
Task:	T3.2 Context modelling and representation
Type:	Report
Approval Status:	Final Draft
Version:	1.0
Number of pages:	26
Filename:	D3.1_ContextRepresentation_2014-10-31_v1.0.pdf

Abstract

In this deliverable we report on context representation and describe the set of contextual features that will be used to characterise media items within MULTISENSOR. We present methods for extracting or deriving the values of these features exploiting both the textual content of each media item along with its associated metadata. Also, we describe how these properties can be represented using an ontology and stored in the semantic repository. After enriching each media item with its context, we exploit the contextual features to support context-based search along with similarity-based retrieval. Last, we discuss how these

features can be utilised to support contextual analytics.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	03/10/2014	Proposal of structure and draft content	I. Miliaraki (BM-Y!)
0.2	13/10/2014	First draft	I. Miliaraki (BM-Y!)
0.3	20/10/2014	Internal BM-Y! review	I. Arapakis (BM-Y!)
0.4	24/10/2014	Version for internal review	I. Miliaraki (BM-Y!)
0.5	28/10/2014	Internal review	G. Casamayor (UPF)
1.0	31/10/2014	Final version after feedback from internal review	I. Miliaraki (BM-Y!)

Author list

Organisation	Name	Contact Information
BM-Y!	Iris Miliaraki	irismili@yahoo-inc.com
BM-Y!	Ioannis Arapakis	arapakis@yahoo-inc.com

Executive Summary

This deliverable reports the first results of the Task 3.2 regarding context modelling, extraction and representation. The goal of the contextual model is two-fold, firstly to facilitate the analysis and interpretation of multimedia content consumed by MULTISENSOR and secondly to provide an additional search layer over this content. Both objectives are pursued by selecting a set of contextual features that characterise a media item. These features can be exploited either in an indirect way, e.g., for clustering, or in a direct way, i.e., to provide to the end users context-based search functionality.

We describe what we consider as context for a media item consumed by the MULTISENSOR architecture and then present in detail the properties we select to include in this contextual representation. Driven by the user requirements (outlined in D8.2), we select a set of contextual features to characterise media items in a meaningful and useful manner. For each of these features, we describe in detail how we can extract or derive their values either exploiting the main content of a media item or by harvesting information from the associated metadata. In the last part of this deliverable, we discuss how we make use of these features within MULTISENSOR providing various examples.

Abbreviations and Acronyms

CEP	Content Extraction Pipeline
DC	Dublin Core
DoW	Description Of Work
IR	Information Retrieval
JSON	JavaScript Object Notation
KB	Knowledge Base
NER	Named Entities Recognition
OPS	Operations Repository
PLSA	Probabilistic Latent Semantic Analysis
POS	Part Of Speech
RDF	Resource Description Framework
RIDF	Residual Inverse Document Frequency
SME	Small and Medium-sized Enterprise
SPARQL	SPARQL Protocol and RDF Query Language
TF/IDF	Term Frequency/ Inverse Document Frequency
WP	Work Package

Table of Contents

1 INTRODUCTION

To effectively understand, interpret or even summarise textual and multimedia content (e.g., a news article, a blog post or a video), one should extract its context and the various attributes that comprise it. We use the term *context* to refer to a set of features that characterise the input (i.e., typically, a media item) in various and diverse ways. Contextual features characterizing a media item can include, among others, its author, its creation date, or its categorical topic. In contrast to context extraction, *content* extraction aims to explain and extract information directly associated with the subject of a media item, such as the mentioned entities or events. The selection of the final set of contextual features is guided by the use case requirements (described in D8.2) that establish the importance and interestingness of each feature.

The context analysis task involves the development of methods for the extraction of contextual features and the determination of a representation for them. The extraction is implemented by a context analysis module (described in D7.2), which takes as input textual content, corresponding for example to a news article or a blog post, along with some metadata. This content is previously processed by a number of analysis components such as the named entity identification module and the content extraction module, which identify named entities and concepts, as well as relations between them. The extraction of contents is based on a lingusitic analysis of the texts which is also useful for the context analysis, e.g. the results of the deep dependency-based syntactic analysis of document sentences. The goal of the context analysis component will be to gather, extract and represent the contextual features exploiting part of the output of the analysis modules preceding it. As a final step, for representation purposes, the contextual features will be modelled according to an ontology and stored as RDF triples in the project main semantic repository.

In terms of usage, the goal is to exploit this additional level of information for supporting context-based search and also similarity-based retrieval. Context-based search will allow the end users to search along different contextual dimensions retrieving relevant items (e.g., items published at a specific date or articles written by a specific author). Last, a context-based analysis tool will be designed and targeted to the use cases for enabling a context-aware interpretation of the data.

This document is divided into a section describing context as a term (Section 2), how context is perceived in MULTISENSOR and which contextual features are considered (Section 3), how these features are extracted or derived from the input (Section 4), how these features are exploited (Section 5), and a closing section, where we summarise about the work done so far (Section 6).

2 CONTEXT: USAGES AND INTERPRETATIONS

2.1 Context as a term

The term "**context**" has various and diverse interpretations depending on the setting it is being considered. A definition provided by Dey and Abowd (2000) defines "context" as follows:

"any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves."

A more generic definition comes from the Oxford dictionary:

"the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood and assessed".

In agreement with the former definitions, we formally define our concept of context in Section 3.

2.1.1 Context in other areas

Information retrieval: The main research area where context-awareness has been applied is **information retrieval** (IR) (Goker et al., 2009). A popular example of context-aware IR systems are Web search engines, where instead of returning identical results to all users, the user context (e.g., her location or her interests) is exploited as prior knowledge for personalizing the results. Therefore, in IR systems, the object in interest is the user and the context is used to characterise this user.

Contextual analytics: Another popular area is **contextual text mining** (Mei, 2010) where a text document is often associated with various kinds of context (e.g., temporal or spatial characteristics) and an analysis is performed to study the correlations among documents sharing the same or similar contextual characteristics. In the latter case, the goal is towards an analytical study and there is no direct interaction between a user and an application.

3 CONTEXT IN MULTISENSOR

In MULTISENSOR, we consume published media items either from traditional media (print, online, broadcast) or from social media sources (such as tweets). To enable an analysis and interpretation of such content, apart from its main content (e.g., in the case of news articles, the main content is the text enriched with photos or videos accompanying it in the article), analysed by other modules such as the NER module and the event detection module (see D7.2), we need to consider also a number of other dimensions characterizing each item, i.e., its context.

In this chapter, we describe what we consider context within MULTISENSOR. To select the features constituting the context of a media item, we review the core user requirements and then couple them with the corresponding contextual features. After proposing and describing the list of desired contextual features, we describe their ontological representation based on the Dublin Core ontology[1].

3.1 Goals

The objects of interest within MULTISENSOR are the media items consumed by the different modules, such as news articles (from print or online media) and other multimedia content. We want to extract, describe and analyse information characterizing these items, i.e., their context, having as ultimate goal to meet the requirements of the use cases. Before describing in detail the different user requirements, we first outline the high-level scientific objectives pursued by MULTISENSOR and related to the contextual analysis component, as stated in the DoW:

- To perform a user- and **context-centric analysis of heterogeneous multimedia and multilingual content**.
- To enable the semantic integration and **context-aware interpretation over the spatiotemporal and psychological dimension of heterogeneous and spatiotemporally distributed multimedia and multilingual data** as audio, video, text and social content interaction.

As can be seen by the objectives, the goal with respect to context-awareness is **towards an analysis and an interpretation** of heterogeneous multilingual content.

More specifically, the contextual component aims to:

- provide **a set of contextual indicators** characterizing the content items and a framework for measuring their impact in the context of our use cases, and to
- provide **models and representation techniques** to be used in effective context-based search.

[1] Dublin Core (DC) ontology: http://dublincore.org/

3.2 Motivation and user requirements

In this section, we review the user requirements (described in D8.2) targeted by the contextual analysis module. Later, guided by the user requirements we select the set of contextual features considered within MULTISENSOR.

3.2.1 Use Case 1 (scenario 1): Journalism

From a journalistic point of view, the contextual module is associated directly or indirectly with the following requirements:

- **User-defined search**: search according to *specific keywords*, search in *specific media sources or media types*, search in *specific languages*, search according to *location*, search within *temporal dimensions,* and item similarity search where the user should be able to select an article/item and retrieve relevant items
- **Context extraction**: analysing *background information about author*, *date of content creation*, *geographic provenance of content* & identifying content and information that is related to specific extracted entities

First, to enable user-defined search on top of the collection of media items stored in the repository of MULTISENSOR, we need to describe these items with the relevant contextual features (i.e., keywords, media source, language, location, time). While keywords can be supported also by a full-text search, in case of textual content or descriptions accompanying multimedia items, discovery of the most important keywords can result in more efficient and effective search. In addition, we can exploit contextual features to support also similarity-based recommendations. For example, two news articles may be considered similar not only if they have textual similarities, but also if they are written by the same author or published by the same media source.

The second requirement specifically targets context extraction by focusing on the analytical functionality of the contextual model. To enable this analysis, the contextual features required include author, date of content creation and geographic provenance of content.

3.2.2 Use case 1 (scenario 2): Commercial media monitoring

From a media monitoring point of view, the contextual module is associated directly or indirectly with the following requirements:

- **User-defined search**: search according to *specific keywords*, search in *specific media sources or media types*, search in *specific languages*, search *according to location*, search within *temporal dimensions*, search by *topic*
- **Contributor analysis**: creation of a database of *authors/contributors*, detection of *personal information about specific contributors* (age, group, nationality etc.)
- **Further analysis**: analysis by metadata

Similar to the journalistic scenario, user-defined search entails associating each media item with a set of contextual features (i.e., keywords, media source, language, location, time). In addition, it is necessary to extract the authors or creators of media items such that a database of authors can be created. Note, however, that this also corresponds to social content such as tweets for which a different procedure may be followed (addressed in MULTISENSOR by Task 3.4 focusing on information propagation and social interaction

analysis). Finally, for supporting further analysis we will consider additional contextual features derived from the metadata accompanying a media item.

3.2.3 Use case 2: SME internationalisation

With respect to the use case of SME internationalisation, the contextual module is associated directly or indirectly with the following requirements:

- **User-defined search**: search according to *specific keywords*, search in *specific languages*, search by *topic* & item similarity search where the user should be able to select an article/item and retrieve relevant items
- **Correlation:** suggestion of similar pieces of information from different sources (especially videos, images and multimedia items)

As in the previous scenarios, user-defined search entails the characterisation of each media item with the proper contextual features (i.e., keywords, topic, language). Contextual features can be exploited to support item similarity, both in terms of suggesting related items when the user is searching for information and in terms of the correlation requirement.

3.3 Contextual module

Before describing what is the context of each type of media item, we provide an overview of the content extraction pipeline (CEP) involving all main components of MULTISENSOR including the contextual module. The CEP pipeline, introduced in D7.2, is depicted in Figure 1, highlighting the contextual component.

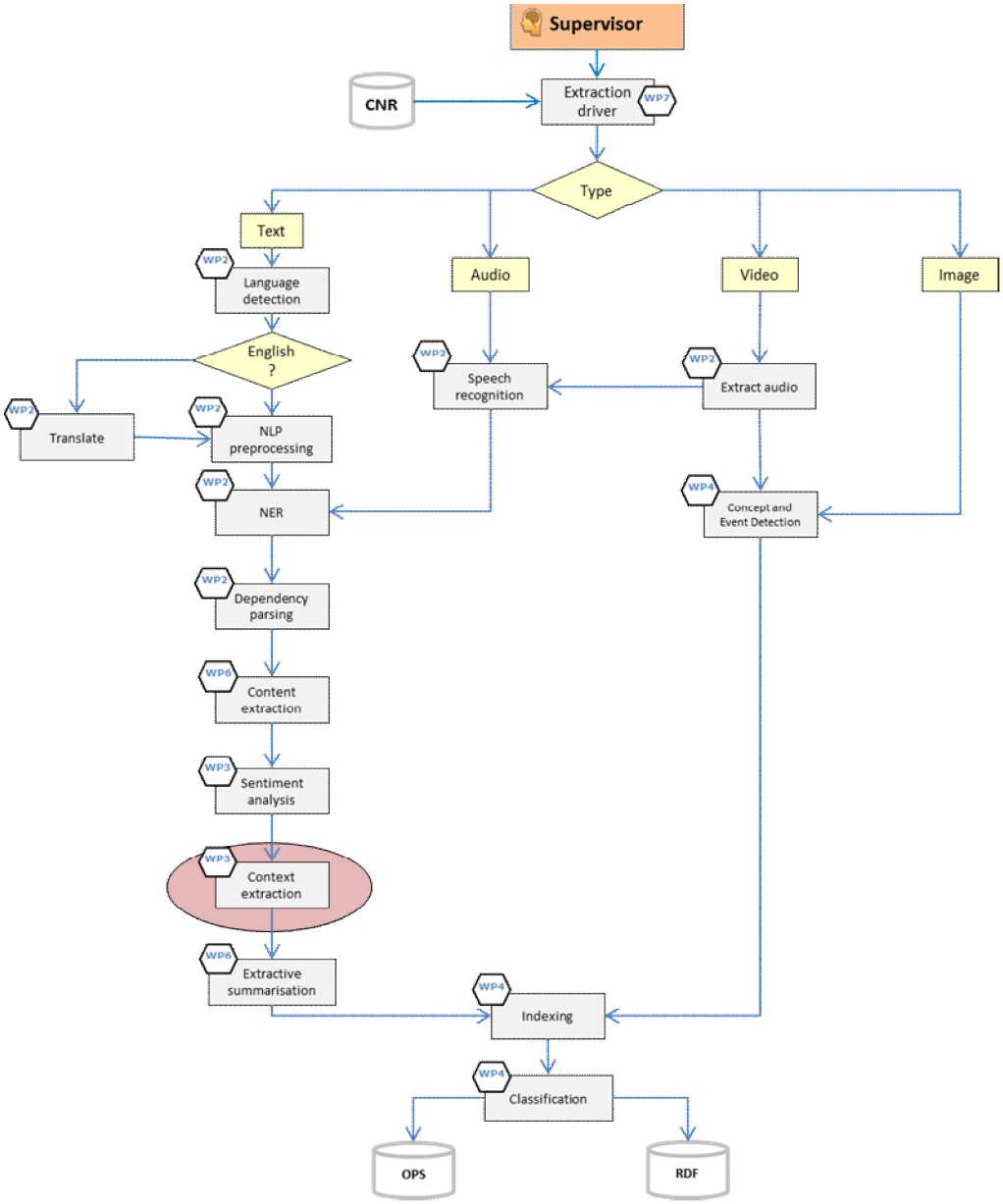

Figure 1: CEP pipeline

As we see in the Figure, taken from D7.2, the context extraction module consumes the output of various different modules such as NER, content extraction, and sentiment analysis, appearing earlier in the processing pipeline. The contextual model takes advantage of this output to extract more effectively the context of each media item.

Figure 2 depicts the architecture and interactions among the three main modules of WP3 cooperating to perform user and context-centric analysis of the input content. As depicted also in the architectural pipeline, first the polarity and sentiment extraction module processes the input object and then it is then passed for processing to the contextual module. While, as described in Section 4, sentiment information is not currently exploited, we expect that it may be proven useful as we elaborate on the contextual feature set.

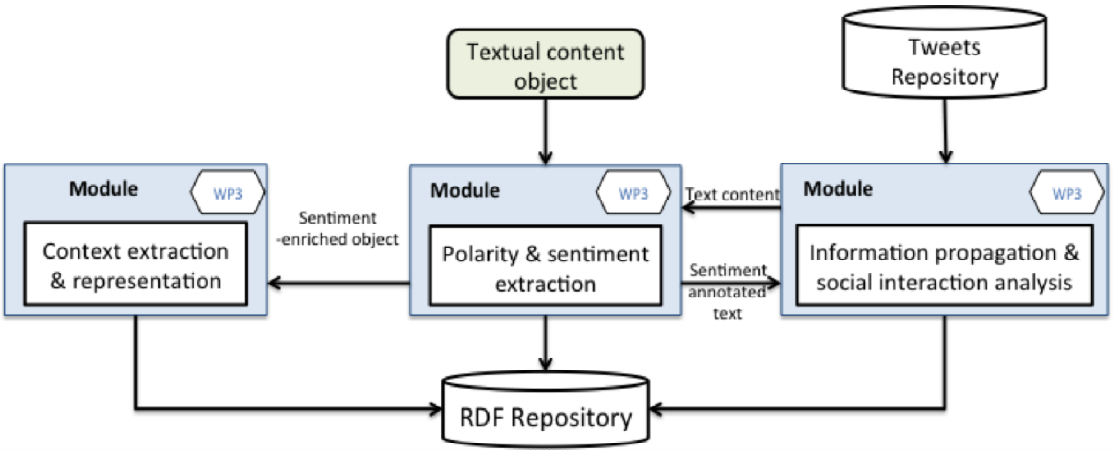

Figure 2: WP3 architecture

3.4 List of contextual features

Guided by the high-level scientific objectives and the specified user requirements described in Section 3.2, we propose a list of contextual features characterizing the media items consumed by MULTISENSOR. In principle, any kind of metadata (explicitly or implicitly provided) associated with a media item can be regarded as potential contextual features. As indicated in the DoW, the focus will be mainly on online content (e.g., news articles or blogs), but other types of content (e.g., printed media) are also foreseen. Contrast to online content, in the latter case of printed content, we expect that only its main part, without any additional metadata, is provided.

Given a media item, we propose using the following contextual features:

- **Author**: an entity responsible for the creation of the item content
- **Source**: an entity responsible for making the item available
- **Title**: a name given to the media item
- **Keywords**: a set of phrases describing the topic of the item
- **Genre**: the style or type of the item
- **Category**: a classification of the item according to its content
- **Date**: a date associated with the creation or availability of the item
- **Location**: a location indicating where the item was created
- **Literary style**: a metric of language formality
- **Language**: the language of the content of item

An example of a media item along with its considered context is depicted in Figure 3.

Figure 3: An article and its contextual representation

In addition to the features produced or explicitly extracted by the contextual module, we will eventually consider also the following additional features contributed by other modules within the MULTISENSOR architecture:

- **Topic**: a topic from an ontology characterizing the item (provided by the topic-detection module)
- **Sentiment**: the sentiment characteristics of the item (provided by the sentiment extraction module)
- **Event(s)**: the events referenced within the content of the item (provided by the event detection module and applicable only to images and videos)
- **Concept(s)**: the concepts identified within the content of the item (provided by the concept detection module and applicable only to images and videos)
- **Entities**: the entities mentioned within the content of the item (provided by the named-entity extraction module)

Some of the above properties are textual and therefore can be represented using a plain string, e.g., title, while other like author and event correspond to entities that can be associated with a number of additional properties. Such properties can either be explicit (e.g., author occupation) or implicit (e.g., author experience). This second level of contextual features will be studied at latter stages of the project and we currently assume a simple form of these entities.

Contextual features can be classified as follows:

Metadata features: this refers to features like the time the article was written, the media that published the article, the author, a list of keywords and a topic. These features should be provided as part of the input.

Extracted features: This includes features like events described in the content, the sentiment of the article, fine-grained topics, or the literary style of an article. These features are extracted either by other components or by the context extraction component itself.

Collective features: This includes features corresponding to an entity like the author who wrote the article and possesses her own context or the events described by the article which are also represented by their own context. This set of more advanced features will be studied at a latter stage of MULTISENSOR.

3.5 Context representation

In this section, we describe how context can be represented in order to enable storage within the ontological repository. At the end of the processing pipeline (described in D7.2 and depicted also in Figure 1), the contents of the JSON container are stored in repositories for later use by the client applications. There are a variety of data formats generated by the analysis services: RDF data is written to the RDF Repository, and structured and literal data is written to one or more databases in the Operations Repository (OPS). With respect to the contextual component, all information regarding the context of an item will be represented in RDF and stored in the RDF repository. Client applications in this case include mainly context-based search and similarity-based retrieval.

3.5.1 Ontological representation

We propose to use the Dublin Core ontology[2] that contains a number of metadata elements for classifying documents and other electronic resources. The vocabulary of Dublin Core can be used to express contextual features. An overview of the main elements included in the Dublin Core ontology in a schematic way along with an example instantiation is depicted in Figure 4 and Figure 5.

An explanation of the different terms provided by the ontology are described in the specification of Dublin Core and can be matched directly with some of the contextual features. Note that all DC properties are optional and repeatable, meaning that they can be defined more than once per each document resource.

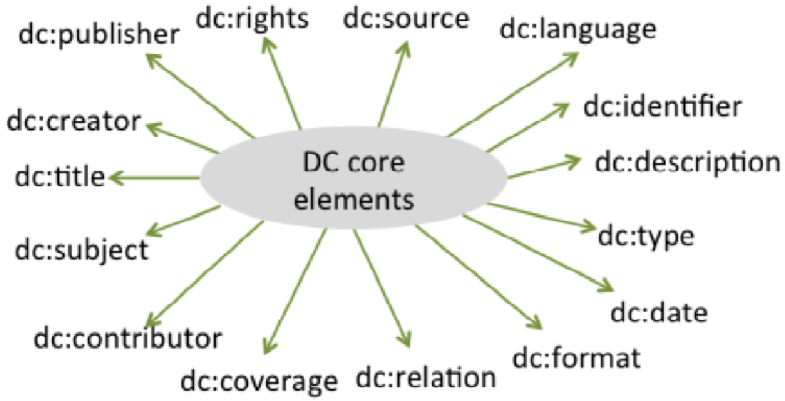

Figure 4: Dublin Core elements.

[2] Dublin Core metadata initiative, http://dublincore.org

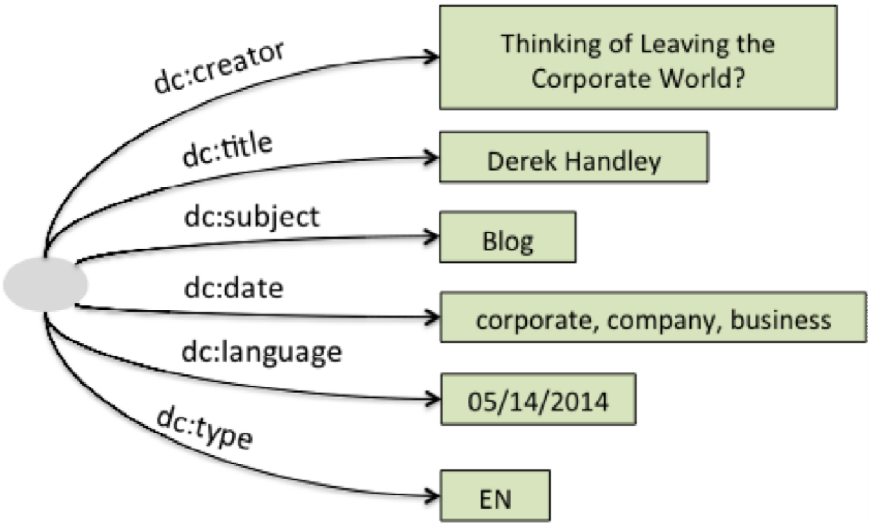

Figure 5: Dublin Core example.

Contextual features are mapped to the Dublin Core Ontology elements as follows: author is mapped to **dc:creator**, source is mapped to **dc:source**, title is mapped to **dc:title,** keywords are mapped to **dc:subject**, language is mapped to **dc:language**, genre is mapped to **dc:type**, and finally date is mapped to **dc:date**. Finally, additional features not supported directly by the schema, such as literary style, will be added by extending the schema.

4 CONTEXT EXTRACTION

In this chapter, we will discuss how we extract or derive the set of contextual features associated with a media item. Depending on the feature, we may exploit a different method based on the main content of the item or on the metadata accompanying the item. In some cases, a contextual feature may be computed by another component (e.g., the topic detection module or the entity extraction module preceding the contextual module in the pipeline of Figure 1). Given the variety and the diversity of the contextual features, the extraction methods also differ. In the following, we describe our basic methods for extracting or deriving the value of each of our proposed features.

4.1 Related work

There is a plethora of related work that has focused on extracting different types of structured information from web pages corresponding to news articles and many other types of pages such as blogs, forums and so on. Arasu and Molina (2003), where among the first who studied the problem of extracting structured data from Web pages by deducing the template of the Web page and exploiting this to extract the values.

More recently, considering the task of author extraction from Web pages, Kato et al. (2008) focus on linguistic features while Changuel et al. (2009) describe a machine learning approach based on C4.5 decision tree exploiting both structure and content information. Zheng et al. (2007) describe an alternative approach where the visual features of the web pages are taken into account for extracting author metadata. The latter approach mainly focuses on retrieving apart from the name of the author, additional information such as its email address or affiliations.

With respect to commercial tools focused on these tasks, tools such as Alchemy[3] and DIFFBOT[4] are capable of extracting a number of fields from web pages, such as author, keywords, entities and sentiment, using sophisticated machine learning techniques.

4.1.1 Author extraction

Online content consumed by MULTISENSOR such as news articles or blogs is commonly related with an author entity primarily responsible for creating this item. This entity may be a person or alternatively an organisation. The author can be specified within the metadata of the item using various possible schemas (e.g., using Dublin Core's *creator* property) or mentioned in its textual content (e.g., with a phrase such as "Written by …"). Since we want to support a plethora of online sources in MULTISENSOR, we address both cases.

Method details

Let us consider a media item for which we want to extract or derive its author entity. First, if there is available metadata for this media item, we check whether this metadata includes the author information. We achieve this by checking for specific properties derived from a

[3] Alchemy API http://www.alchemyapi.com/

[4] Diffbot extraction tool http://www.diffbot.com/products/

sample of items stored in the repository. A non-exhaustive list of such structural properties is included in Table 1.

Id	Schema	Property
1	Dublin Core	`dc:creator`
2	HTML5 rel attribute	`rel="author"`
3	schema.org	`itemprop="author"`
4	vCard ontology	`author vcard`
5	HTML meta tag	`meta name ="author"`

Table 1: Structural properties for author information

If no metadata are associated with the media item or author information cannot be harvested from the metadata, we need to investigate whether we can extract the author information from the main content, i.e., textual content.

To achieve this, we use a rule-based approach inspired by the following observations, described first by Changuel et al. (2009):

1. The author should be represented by a named entity corresponding either to a person or an organisation
2. The author is often placed in the beginning of the main content or at the end of the main content
3. Next to the author name we can find a mention of the document creation date, author contact information such as her email or twitter account, and the name of the organisation she belongs to.
4. Phrases surrounding the author name use a rather restricted vocabulary such as "written by", "created by", etc.

Based on the above observations and on the keywords surrounding text commonly indicating author names, we design a method for extracting author names included in the main textual content of a media item (typically articles or blog posts). Note that the tagging of named entities is performed by the respective module, i.e., the named entity recognition module.

In order to create our rule knowledge base (KB) we exploit a sample of media items, retrieved from the MULTISENSOR news repository, originally crawled and made available by pressrelations. We start by manually checking and annotating each item with the name of the associated author entity. Then, we make use of these annotations to learn both structural patterns from the metadata and textual patterns from the main content. These patterns, along with additional information, such as their position within the content, are exploited to create our rule knowledge base (KB). Finally, we apply these rules to unseen items and extract automatically their author. A non-exhaustive list of such textual patterns is depicted in Table 2 including both specific ("Written by AUTHOR_NAME") and more general patterns ("by AUTHOR_NAME"). Currently, we consider this to be a tentative list, since the sample of items considered may not be representative enough, and our plan is to continuously expanding this list as a more diverse collection of items is included in the MULTISENSOR repository.

While we prioritise the matching against the more specific patterns, matching against the general ones can lead to many false positives (i.e., recognise the wrong entity as the author). Dealing with such cases and quantifying their impact is subject of future work.

Note that placeholder "AUTHOR_NAME" corresponds to a named entity identified by the named entity recognition module preceding the contextual module in the architecture pipeline (see Figure 1).

Id	Textual patterns	Example
1	`Posted by: AUTHOR_NAME`	*Posted by: ESI Africa*
2	`Posted by AUTHOR_NAME, DATE`	*Posted by Ty Velde April 9, 2014 11:39 PM*
3	`Written by AUTHOR_NAME`	*Written by Jordan Finney*
4	`Compiled by AUTHOR_NAME`	*Compiled by Michal Bacia*
5	`Follow AUTHOR_NAME on Twitter:`	*Follow Courtney Carter on Twitter:*
6	`Contact: AUTHOR_NAME`	*Contact: Kathryn Cosse*
7	`By AUTHOR_NAME`	*By Michael Turpin*
8	`By AUTHOR_NAME, Ph.D.`	*By Romeo Vitelli, Ph.D.*
9	`AUTHOR_NAME is a * who blogs at`	*Greg Smith is a psychiatrist who blogs at*
10	`-- AUTHOR_NAME`	*-- Robert Preidt*

Table 2: Textual patterns for author extraction

4.1.2 **Keyword extraction**

To improve the relevancy of the retrieval, the keyword extraction task aims to select a small set of words or phrases that can describe the meaning or the content of a media item. Similar to other contextual features, keywords, or more generally keyphrases, may be included explicitly as part of the metadata associated with the item (e.g., using Dublin Core's *subject* property). These can either be keywords specified manually by a human editor (in this case, keywords usually capture with high precision the topic of the item) or they can be generated automatically, e.g., from the title of the item (each word in the title can correspond to a different keyword). In the latter case, the usefulness of the keywords is reduced since titles do not necessarily consist of a good set of keywords. For example, in Figure 3, the title of the article, "Thinking of leaving the safety of the corporate world", contains among others very abstract keywords such as "thinking" or "world".

Alternatively, if no such list is provided, one can exploit importance metrics such as TF-IDF to extract a set of keywords from the textual content (Matsuo and Ishizuka, 2004.). The TF-IDF metric (short for term frequency – inverse document frequency) is a statistic reflecting the importance of a word within a document of a corpus. Words appearing relatively often within a document (term frequency) and rarely in the other documents (inverse document frequency) of the corpus receive a higher score since the words are considered the most informative ones. Alternatively, other scores such as RIDF, Weirdness or C-value (Knoth et al., 2009) can also be applied.

Method details

Our basic approach for extracting keywords from textual content works as follows:

1. First, pre-compute TF/IDF statistic for a collection of words, n-grams or sequences of words using a representative subset of the corpus.
2. Exploit the TF/IDF statistic along with other relevance criteria (commonly used in search engine ranking), such as the location of each word within a web page or the appearance of a word in the title, to rank the candidate terms.
3. Select as keywords the top-k candidate terms based on the relevance score computed previously.

Given that the media contents stored in the news repository of the MULTISENSOR project can be categorised into separate classes, either using the "category" contextual feature or the topic-detection module, we can improve the effectiveness of TF/IDF by considering each sub-collection of documents separately. A similar observation is described by Yang et al. (2011), where the authors pointed out that the relevancy of a word is domain-dependent. For example, the word "gun" found in articles concerning military news shouldn't always be selected as a keyword, while the same word if found in social news should be definitely selected since its importance is significantly increased.

In the case of social content, extracting keywords becomes more complicated. Li et al. (2013) studied this problem by considering several intuitive features such as the capitalisation of words (e.g., if a word is capitalised or starts with a capital letter) or their linguistic features (e.g., if a word is a noun). Our plan is to investigate keyword extraction from social content separately. Evaluating the effectiveness and efficiency of the generic approach is left as future work.

The literary style of a document refers to the way a document is written. In the context of MULTISENSOR, we aim to characterise the style of any kind of textual content. While the notion of literary style captures many different characteristics of how language is used within a document, our focus will be on language formality.

Formality is considered an important dimension for characterizing the language style of document. Texts of different genres will consequently have different degrees of formality, e.g., a scientific article versus a blog post versus a tweet. If we consider the journalistic use case, a journalist investigating a specific topic may be interested in discovering different views on this topic described either in a formal or an informal way (possibly corresponding to expert's and non-expert's opinions).

Heylighen and Dewaele (1999) defined the formality of a language and proposed ways to measure it. According to the authors, a formal style is characterised by detachment, accuracy, rigidity and heaviness; an informal style is more flexible, direct, implicit, and involved, but less informative.

Given these principles, the authors proposed an empirical metric, called the Formality score (F-score), based on the frequencies of different word classes. In particular, nouns, adjectives, articles, and prepositions are more frequent in formal styles, while pronouns, adverbs and verbs are more frequent in informal styles. F-score as defined by Heylighen and Dewaele (1999) is computed as follows:

F = (noun frequency + adjective freq. + preposition freq. + article freq. – pronoun freq. – verb freq. – adverb freq. – interjection freq. + 100)/2

The frequencies correspond to percentages of words belonging to a particular category with respect to the total number of words. As a result, F-score is a percentage between 0 and 100%. The more formal the language of the text, a higher F value is expected. As an example of how the F-score differentiates across different genres of content we include measurements from Heylighen and Dewaele (1999) in Figure 6.

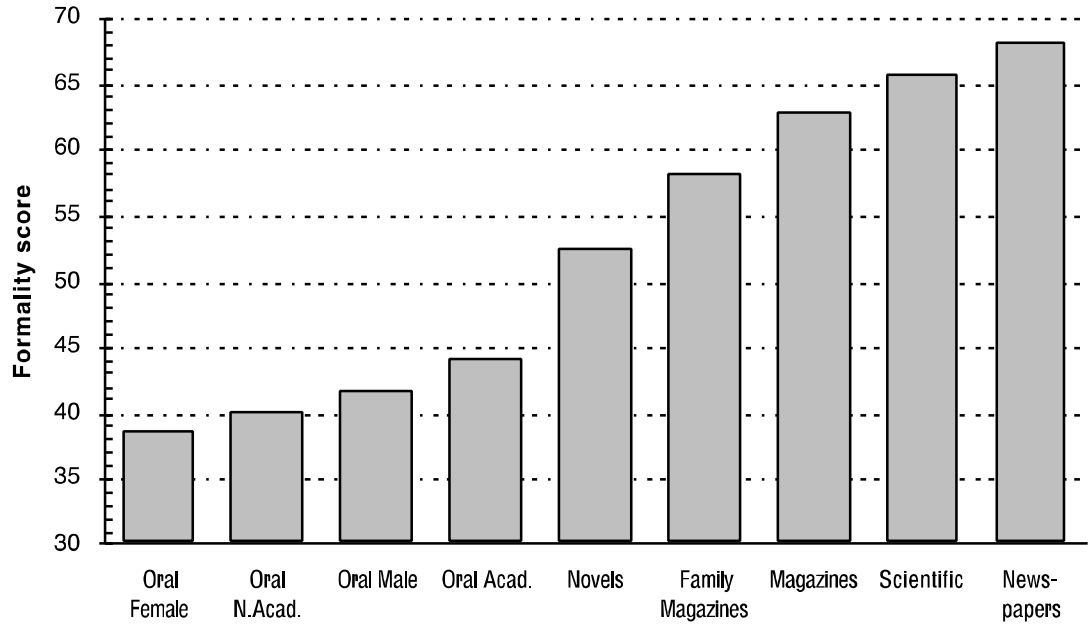

Figure 6: Formality scores reported by Heylighen and Dewaele (1999)

Method details

We currently exploit the F-score as defined above to assess the formality of textual content. To allow easier interpretation, formality scores are mapped to three formality classes: formal, semi-formal and informal. This is currently achieved by equally dividing the range of F-score into three equal sub-ranges. To compute the formality score, POS tagging should be performed a priori. POS tags are provided by the dependency parsing service (WP2).

4.1.3 Extracting other contextual features

We have described our methods for extracting the author or creator of a media item, a list of keywords describing the subject or topic associated with a media item, and the literary style of some textual content of a media item. The remaining contextual features are either harvested from the metadata accompanying the media item by checking the corresponding properties. In some cases, properties are also retrieved from databases and provided by the crawler along with the media items. In the following we describe how we populate each of the remaining features.

Source & Location

MULTISENSOR news repository specifies the source for each media item it stores. This source information is commonly retrieved from a database mapping base URLs to their media sources. This database also contains information about the location of each media source.

Title

This is commonly supplied by the crawler which first extracts headlines and links from overview pages (e.g., the main page of Guardian: http://www.theguardian.com/uk) and then crawls the content of single web pages.

Date

We assume that the date of creating a media item is the same as the date of crawling. This is reasonable for new media items given that crawling mostly takes place on a daily basis

Language

Language is detected and provided by the language detection module (WP2).

Genre & Category

Genre and categorical information are retrieved from the metadata. If no metadata exist, simple heuristics can be exploited to detect genre. In the case of the category, if not found in the metadata information of the article, the respective topic-based classification module will provide the necessary input.

5 EXPLOITING CONTEXT

In this chapter, we sketch the different ways in which contextual features will be exploited within MULTISENSOR either to enable context-based search or in terms of performing contextual analytics.

5.1 Context-based search

Contextual modeling will be used mainly for enabling customised search based on the different contextual features. After enriching each media item with its set of contextual features, the representation will be stored in the RDF repository enabling context-based search allowing either selection operations over single contextual features or join operations over a combination of contextual features.

Examples of such queries expressed in natural language are the following:

1. Which articles where written by author *Greg Smith*?
2. Which articles were published by a media source based in Germany on January 2014?
3. Which articles categorised under "Economy" published by a media source based in Germany have a positive sentiment?
4. Which articles referring to Angela Merkel have an informal literary style?

We expect that these queries will be expressed in SPARQL (potentially using a set of predefined templates). The representation of the contextual features will be based on the Dublin Core ontology, described in Section 3.5.

In addition, similarity-based retrieval will be supported exploiting among other properties, the contextual ones. Alternatively, the latter can also be supported through the clustering module.

5.2 Contextual analytics

Contextual text mining is concerned with extracting topical themes from a text collection with context information (e.g., time and location) and comparing or analysing the variations of themes across the different contexts. The goal is to reveal interesting content patterns in such contextualised text data.

Assuming a text document associated with the following contextual features: time written, location produced, author and source. The contents of the text documents with the same or similar context are often correlated in some way. For example, news articles written in the period of some major event all tend to be influenced by the event in some way, and blogs written by the same blogger tend to share similar topics.

In previous works, Mei and Zhai (2009) extend the probabilistic latent semantic analysis model (PLSA) by introducing contextual variables that model the context of a document. Such a model can be applied to various mining tasks like temporal text mining, spatiotemporal text mining, author-topic analysis and cross-collection comparative analysis.

Example contextual analytics are the following:

1. Author-topic evolutionary analysis over time
2. Media source-topic evolutionary analysis over time
3. Keyword tag clouds across different locations

5.3 Usage of context in other modules

Apart from exploiting the contextual features to support customised search or contextual analytics for the end user, features will also be used in other MULTISENSOR tasks. We briefly describe how the topic-based classification task and the clustering task can make use of the contextual features.

5.3.1 Topic-based classification task

The main modality exploited in the classification task is text. More specifically n-gram textual features are extracted from the articles. Apart from these textual features, the extracted contextual features can also be represented with a bag-of-word model and investigate whether they could further improve the classification performance of our model (either in an early-fusion manner e.g. concatenation with the N-gram textual features or in a late-fusion manner). In other words, the motivation for the extraction of the contextual features, with respect to the classification task, is the investigation of whether they can assist the main textual modality (n-gram features) in the classification task.

5.3.2 Clustering task

In the case of clustering, the extraction of specific features relevant to characteristics such as location, time or formality will allow for multidimensional and multimodal clustering. As an example, spatiotemporal clustering based on authors or contributors, which will certainly allow for event recognition, identification of relevant events (i.e., occurring in different periods or countries), identifying articles by similar authors and so on.

6 CONCLUSIONS

In this deliverable we reported on the contextual representation of the media items consumed by MULTISENSOR. We described a number of contextual features inspired by the requirements of the use cases and discussed our basic approaches for extracting or deriving these features from the text and the available metadata. We presented how these features can be represented using an ontology exploiting the widely used Dublin Core schema, such that the information can be stored in the semantic repository and queried in an effective an efficient way. We concluded our report by discussing how users will exploit the contextual information from the items to perform context-based search and similarity-based retrieval. In addition, we described other modules within the MULTISENSOR architecture that will take advantage of the context to improve their own functionality. Ongoing and future work will focus on extending and improving the extraction methods of the contextual features aiming to increase both precision and recall.

7 REFERENCES

Arasu, A., and Garcia-Molina, H. 2003. “Extracting structured data from web pages”, In Proceedings of the 2003 ACM SIGMOD international conference on Management of data.

Changuel, S., Labroche, N., and Bouchon-Meunier, B. 2009. “Automatic web pages author extraction”, Flexible Query Answering Systems, pp. 300-311.

Dey, Anind K., and Gregory D. Abowd. 2000. “The context toolkit: Aiding the development of context-aware applications”, Workshop on Software Engineering for wearable and pervasive computing.

Goker, A., Myrhaug, H., and Bierig, R. 2009. “Context and information retrieval. Information Retrieval: Searching in the 21st Century”, p. 131-57.

Heylighen, F., and Dewaele, J.-M. 1999. “Formality of language: definition, measurement and behavioral determinants”, Interner Bericht, Center “Leo Apostel”, Vrije Universiteit Brüssel.

Kato, Y., Kawahara, D., Inui, K., Kurohashi, S., and Shibata, T. 2008. “Extracting the author of web pages”, In Proceedings of the 2nd ACM workshop on Information credibility on the web, p. 35-42.

Knoth, P., Schmidt, M., Smrz, P., and Zdrahal, Z. 2009. "Towards a framework for comparing automatic term recognition methods". Technical Report.

Li, Haiying, Zhiqiang Cai, and Arthur C. Graesser. 2013. “Comparing Two Measures for Formality”. FLAIRS Conference.

Li, Z., Zhou, D., Juan, Y. F., and Han, J. 2010. “Keyword extraction for social snippets”, In Proceedings of the 19th international conference on World wide web, p. 1143-1144.

Matsuo, Y., and Ishizuka, M. 2004. “Keyword extraction from a single document using word co-occurrence statistical information”, *International Journal on Artificial Intelligence Tools*, *13*(01), p. 157-169.

Mei, Q. 2010. “Contextual text mining”, Doctoral dissertation, University of Illinois at Urbana-Champaign.

Mei, Q., and Zhai, C. 2005. “Discovering evolutionary theme patterns from text: an exploration of temporal text mining”, In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, p. 198-207

Yang, Y., He, L., and Qiu, M. 2011. “Exploration and improvement in keyword extraction for news based on TFIDF”, Energy Procedia, 13, p. 3551-3556.

Zheng, S., Zhou, D., Li, J., and Giles, C. L. 2007. “Extracting author meta-data from web using visual features”, In Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on Data mining, p. 33-40.