MULTISENSOR

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

FP7-610411

D3.3

Techniques for sentiment extraction and social interaction analysis

Dissemination level:	Public
Contractual date of delivery:	Month 22, 31/08/2015
Actual date of delivery:	Month 22, 31/08/2015
Workpackage:	WP3 User and Context-centric Content Analysis
Task:	T3.3 Polarity and sentiment extraction T3.4 Information propagation and social interaction analysis
Type:	Report
Approval Status:	Final draft
Version:	1.0
Number of pages:	47
Filename:	D3.3_SentimentExtractionSocialInteraction_20150831_v1.0.pdf

Abstract

In this deliverable we report on the following tasks: *polarity and sentiment extraction* and *information propagation and social interaction analysis*. First, we describe a set of sentiment features that will be used to characterise news media textual content within MULTISENSOR, and propose methods for extracting and evaluating these features. In addition, we describe algorithms to analyze the network of interactions among social media users, and the propagation of information within the network. Moreover, we describe how these methods

are represented in the MULTISENSOR architecture.
The information in this document reflects only the author’s views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	08/07/2015	Initial draft	Filipa Peleja (BM-Y!)
0.2	21/08/2015	Review	Ioannis Arapakis (BM-Y!)
0.3	24/08/2015	Second draft	Filipa Peleja (BM-Y!)
0.4	24/08/2015	CERTH contribution	Ilias Gialampoukidis, Stefanos Vrochidis (CERTH)
0.5	27/08/2015	Review ready version	David García-Soriano (BM-Y!)
0.6	28/08/2015	Document review	Dimitris Liparas (CERTH)
1.0	30/08/2015	Final document	Filipa Peleja (BM-Y!)

Author list

Organization	Name	Contact Information
BM-Y!	David García-Soriano	davidgs@yahoo-inc.com
BM-Y!	Filipa Peleja	peleja@yahoo-inc.com
BM-Y!	Ioannis Arapakis	arapakis@yahoo-inc.com
CERTH	Ilias Gialampoukidis	heliasgj@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

This document presents the framework and the initial results of the baseline approaches for sentiment (Task 3.3) and social network analysis (Task 3.4) tasks in MULTISENSOR projects. First the deliverable presents the state of the art methods in these two areas (sentiment and social network analysis) and then it reports the application of the proposed methods in the MULTISENSOR framework and use cases.

With respect to the sentiment extraction task, we present the initial results for sentiment classification. Specifically each sentence of an article is associated with a specific sentiment. On the other hand, the social network analysis focuses on community detection and centrality and social influence analysis. In this context we present the initial results of community detection techniques in MULTISENSOR.

Abbreviations and Acronyms

LDA	Latent Dirichlet allocation
RDF	Resource Definition Framework
MPQA	Multi-perspective Question Answering
SME	Small or medim-sized companies

Table of Contents

1 INTRODUCTION

This report presents the methodology followed in MULTISENSOR for the sentiment and social network analysis modules. The implementation will be integrated into the general architecture of MULTISENSOR project. More specifically, this deliverable describes the proposed work for Work Package 3 (WP3): User and Context-centric Content Analysis. However, the work described in this document contributes towards the implementation of tasks T3.3 and T3.4, as described in the project roadmap D7.1 and scheduled to be completed by end of M22. According to roadmap D7.1, T3.3 and T3.4 refer to:

- T3.3: Applying sentiment analysis algorithms for the social web through a robust opinion mining system and by considering inter-dependencies between expressions and opinion holders.
- T3.4: Discovering relevant communities, information propagation, and opinion mining by applying social mining tools.

The analysis of humans’ viewpoints is known as sentiment analysis. Although this field of study had some research prior to year 2000 (Hatzivassiloglou & McKeown, 1997; Wiebe 1990; Wiebe, 1994) this was the year in which sentiment analysis emerged as a very active research area. Sentiment analysis is a field of study that deals with opinion-oriented natural language processing – it is typically applied to the analysis of structured or unstructured free text documents. These text documents represent people’s opinions towards entities such as products, organizations, individuals and their features. For the different levels of text granularity there are numerous approaches for analysing user opinions: *sentiment analysis*, *emotion analysis*, *opinion mining*, *opinion extraction*, *sentiment mining*, *subjectivity analysis*, *affect analysis*, *emotion analysis* and *review mining* (Liu, 2012). In this context, the focus of the sentiment extraction task is to improve existing sentiment analysis algorithms for the social web. This task is guided by the requirements described in T3.3 that define the importance of performing polarity and sentiment extraction.

Opinions include appraisals, thoughts and emotions towards a product or individual. Usually an opinion is given in the form of a review, comment or purchase evaluation. The sentiment analysis of a user’s opinion can be defined as mapping the text to one of the classes (labels) from a predefined set. Usually the classes of the predefined set are “negative” and “positive”, “objective” and “subjective”, or in a scale that goes beyond binary such as “negative”, “positive” and “neutral” or in a range of numbers such as from 1 to 5 or from 1 to 10 (Pang & Lee, 2008). Sentiment words have a natural association to people opinions and opinions tend to target specific people, organizations or products. To this end, we propose to identify sentiment words and measure entities reputation.

The polarity and sentiment extraction task involves the development of methods for extraction of sentiment words and the determination of their polarity and weight. The sentiment words extraction will be implemented by the sentiment analysis service, which takes as input textual content of news articles. This content is previously pre-processed for sentence level extraction, part-of-speech syntactic analysis, named entities detection and finally, stop words removal. This linguistic analysis is highly important for the sentiment extraction module, e.g. the association of named entities and sentiment words. In this module, we aim to exploit the overall sentiment, sentiment by sentence level and sentiment targets within news articles. As a final step we will model the output of our module according to the project's main semantic repository, meaning that the output will be stored as a RDF object.

In terms of usage, the goal is to exploit this additional level of information for a better understanding of news articles content. Polarity, sentiment words, and opinion target(s) extraction will allow the end users to identify the most sentimental sections of the news article and the sentiment expressions that had the highest contribution.

The Social Network Analysis task (T3.4) aims to understand the patterns of activity in online social media (tweets, blogs, web pages, etc.). This includes building enriched user profiles with information concerning about interests, level of activity and communication patterns with other users in the social network. These interactions among users are then modelled as a graph, allowing us to leverage existing graph mining techniques such as the discovery of relevant communities, the study the propagation of information throughout the network, and the detection of users that are highly influential or play a central role within the community. This analysis is performed in a query-dependent manner (the input is a list of keywords of interest, used to define the topic of interest).

The rest of the deliverable is organised as follows. Section 2 provides an overview of the state-of-the-art sentiment analysis techniques. Section 3 discusses how sentiment analysis is perceived in MULTISENSOR and which sentiment features are considered. In Section 4, we discuss sentiment analysis extraction techniques and present some initial experimental results. Section 5 reviews several state-of-the-art social network analysis techniques and in Section 6 we present the proposed framework for MULTISENSOR, as well as preliminary experimental results. Finally, some concluding remarks are provided in Section 7.

2 SENTIMENT ANALYSIS

Opinionated text, also known as subjective text, is text consisting of words, phrases and/or sentences that express sentiment. The difference between these terms and factual terms is in the notion of *private state*. As (Quirk et al., 1985) define it, a *private state* is a state that is not open to objective observation or verification: "*a person may be observed to assert that God exists, but not to believe that God exists. Belief is in this sense 'private'.*" (p. 1181). More recently, the term *subjectivity* for this concept has been adopted by the respective research community (Liu, 2012; Wilson & Wiebe, 2005). Although this research domain has received much attention, the problem is still far from being completely solved (Liu, 2012). One of the main challenges is that opinionated language varies over a broad range of discourse, a system with a fixed vocabulary will not be enough to represent user's opinion (Wilson et al., 2004).

2.1 Background

Sentiment analysis enfolds various techniques for detecting words that express a positive and negative feeling or emotion (Liu, 2012). These words are commonly known as *sentiment words* or *opinion words*. Beyond words, n-grams (contiguous sequence of *n* words) and idiomatic expressions are commonly used as sentiment words. Such examples are the expressions "terrible", "quite wonderful" and "break a leg". Sentiment words can be used to predict sentiment classes for users' opinions and these words have proved to be useful in sentiment analysis tasks (Liu, 2012). For this reason, the past decade has witnessed a surge of research on numerous algorithms that compile lists of sentiment words – the so called *sentiment lexicons* (Baccianella et al., 2010; Ding et al., 2008; Esuli & Sebastiani, 2006; Rao & Ravichandran, 2009; Takamura et al., 2005; Velikovich et al., 2010). In this context, topic models gained popularity as a tool for automatic corpus summarization and document browsing on large-scale data. These models have been integrated into the context of online commerce, allowing to identify important pieces of information (i.e. sentiment expressions) (Moghaddam & Ester, 2011; Ramage et al., 2009; Titov & McDonald, 2008).

2.2 Levels of Granularity of Sentiment Analysis

Initial work in sentiment analysis aimed at identifying overall positive or negative polarity within full documents (e.g. reviews). Later works identified that sentiment does not occur only at document-level, nor is limited to a single valence or target (Cambria et al., 2013). Hence, sentiment analysis has been investigated at four granularity levels: document-, sentence-, word- and entity- or aspect-level. Usually entity- or aspect-level involves extracting product features that are used to express an opinion (Hu & Liu, 2004a; Hu & Liu, 2004b; Popescu & Etzioni, 2005). Finding semantic orientation at word- or phrase- level differs from entity- or aspect- level since it is related to specific word families that are mostly used to express a sentiment. At word-level researchers have mainly used two methods to automatically annotate sentiment: dictionary-based and corpus-based (Figure 1). Others have also chosen to manually annotate at word-level, however, relying on a manual approach is highly time consuming and subjective (Liu, 2012; Ding et al., 2008).

In comparison to document- and sentence-level, entity- or aspect-level allows a finer-grain analysis (Liu, 2010). The latter involves extracting product features that users dislike and like

(Hu & Liu, 2004b) while document- or sentence-level sentiment words tend to be used, in order to predict sentiment classes for users opinions (Liu, 2012). LDA-based models are considered state-of-the-art for aspect-based sentiment analysis, in which topic models have proved to be successful when applied to reviews websites such as IMDb or TripAdvisor (Moghaddam & Ester, 2012; Lim & Buntine, 2014). On the document-level approach, a state-of-the-art approach was introduced by (Turney & Littman, 2002). They implemented an unsupervised learning algorithm to evaluate reviews' polarity. For each review, the authors compute the average polarity of its constituent words or phrases. Other works (Pang et al., 2002; Heerschop et al., 2011) have also addressed the sentiment analysis task by using a document-level approach. A common use of sentence-level sentiment analysis is to capture opinionated sentences (Wiebe et al., 1999). To this end, the goal is to distinguish between sentences that express factual information (objective) and sentences that express an opinion (subjective) (Hatzivassiloglou & Wiebe, 2000). Even so, all these methods require an understanding of "how and which" words express human preferences.

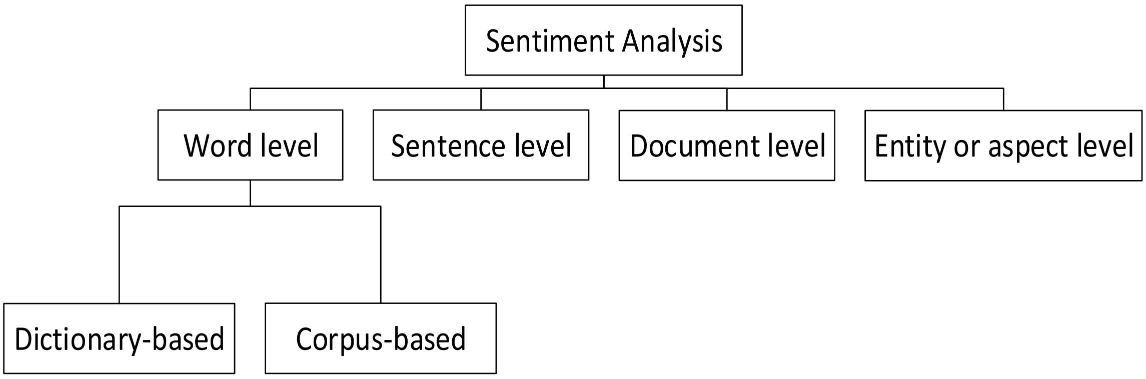


Figure 1. Sentiment analysis granularity levels.

Dictionary-based

Dictionary-based methods are the most straightforward approach to obtaining a sentiment lexicon. These methods use a seed list of words or use them in a bootstrap approach to discover new words (Liu, 2012). The strategy is to use a thesaurus or lexical database (e.g. WordNet) as a seed list of words (Ding et al., 2008). A common assumption in such techniques is that semantic relations transfer sentiment polarity to associated words (Kamps et al., 2004; Hu & Liu, 2004a). For instance, using the synonyms' semantic relation, the sentiment word "lovely" will transfer its positive polarity to its synonyms "adorable", "pretty" etc. (Bross & Ehrig, 2013). Others have chosen to use pre-compiled lists of sentiment words with similar techniques (Hu & Liu, 2004b). The former pre-compiled lists are known as sentiment lexicons. Previous work has made available to the research community numerous sentiment lexicons like SentiWordNet[1], General Inquirer[2], Urban Dictionary[3], Twitrratr [4] and Multi-perspective Question Answering (MPQA)[5]. A sample of

[1] http://sentiwordnet.isti.cnr.it/

[2] http://www.wjh.harvard.edu/~inquirer/

[3] http://www.urbandictionary.com/

[4] https://twitter.com/twitrratr/

generic positive seed of words can be words such as “good”, “nice” and “excellent” and a negative set contain words such as “bad”, “awful” and “horrible”. Usually dictionary-based methods observe the sentiment word occurrence and weaken its sentiment transfer by observing the words proximity. Furthermore, they take into account negation and/or neutralization tokens. The scope of negation aims at detecting polarity changes and neutralization overrides the sentiment polarity effect. Indications of these tokens are words such as “not”, “although”, “never” and “would”, “should” and “hope” for negation and neutralization respectively.

Corpus-based

Corpus-based methods have proven to be more successful compared to approaches that employ pre-built sentiment dictionaries. As reported by (Aue & Gamon, 2005), the later approaches usually fail to generalize. Corpus-based methods can be split into two main categories. The first category (1) uses a seed list of sentiment words, often from a pre-built sentiment dictionary. Later, for a specific domain corpus, the sentiment word list is used to learn other sentiment words. The second category (2) implements a method to obtain a sentiment word lexicon for a specific domain corpus. In (1) a straightforward approach is to extract sentiment words through the proximity to an initial seed of words (Liu, 2012). In an early approach, (Hatzivassiloglou & McKeown, 1997) used a seed of adjectives with a set of linguistic constraints to capture sentiment words. Later, (Turney & Littman, 2003; Turney & Littman, 2002) used (Hatzivassiloglou & McKeown, 1997) list of adjectives and the General Inquirer dictionary to perform sentiment classification analysis. In (Turney, 2002), sentiment phrases were captured by evaluating the proximity to an adjective or a verb. In this context, (Turney & Littman, 2002) reported a study where sentiment classification improved the classifier performance using only adjectives as sentiment words. Nonetheless, previous work (Esuli & Sebastiani, 2005; Heerschop et al., 2011; Takamura et al., 2005; Turney & Littman, 2003) has also shown that other word families, such as adverbs, nouns and verbs, are also qualified with sentiment intensity.

(Bethard et al., 2004) proposed a supervised statistical classification task to distinguish between opinionated (subjective) and factual (objective) documents. Under the purpose of obtaining a sentiment lexicon, subjective documents were used to compute words’ relative frequency. The authors used a pre-built lexicon – a seed list of 1,336 manually annotated adjectives (Hatzivassiloglou & McKeown, 1997) – and, later computed the sentiment lexicon with a modified log-likelihood ratio of the words’ relative frequency. Later, (Qiu et al., 2009) proposed to obtain a domain-specific sentiment lexicon by using an initial seed of sentiment words in a propagation method. Here, the authors used words from a sentiment lexicon as seed in a sentiment word detection process that iterates until no new sentiment words are added to the lexicon. The process detects sentiment words by observing its relation to the initial seed of sentiment words and later, the newly extracted sentiment words are used to detect more sentiment words. For this work the authors explored syntactic relations between sentiment words and features. This technique contrasts with (Hu & Liu, 2004b), in which the authors proposed to extract features with distance-based rules. However, as (Qiu

[5] http://mpqa.cs.pitt.edu/

et al., 2009) comment, (Hu & Liu, 2004b) proposed a method in order to detect product features and not to expand a sentiment word lexicon.

2.3 Subjectivity vs Objective Content

Subjectivity in natural language refers to certain combinations of the language used to express an opinion (Liu, 2010). Early work (Wiebe, 1994) defines subjectivity classification as an algorithm that evaluates, in a sentence or document, the linguistic elements that express a sentiment. In other words, objective and subjective sentences can be defined as follows: *"An objective sentence expresses some factual information about the world, while a subjective sentence expresses some personal feelings or beliefs."* (csz) That is, subjectivity in natural language refers to certain combinations of the language that are used to express an opinion (Tang et al., 2009; Wiebe, 1994). Early work (Wiebe, 1994) defines subjectivity classification as an algorithm that evaluates, in a sentence or document, the linguistic elements that express a sentiment – sentiment words or, as Wiebe denotes, *subjective elements*. For sentences labelled as subjective it is common to apply a sentiment classifier to evaluate the respective sentiment polarity and/or strength (Liu, 2010). Subjectivity classification has been extensively investigated in the literature (Hatzivassiloglou & McKeown, 1997; Hatzivassiloglou & Wiebe, 2000; Riloff et al., 2006; Riloff & Wiebe, 2003).

(Hatzivassiloglou & Wiebe, 2000) claim that adjectives are strong indicators of subjective sentences. Their method uses adjectives to detect potential subjective sentences. Previous work (Wiebe et al., 1999) had a similar method but, instead of adjectives, also used words from the family of nouns. More recently, (Wiebe & Riloff, 2005) introduced a bootstrapping method that learns subjective patterns from un-annotated documents. For this method, the authors needed to define an initial set of rules that were manually annotated and, to this end, required linguistic expertise (Scheible & Schütze, 2012). (Riloff et al., 2006) has also proposed a method that defines subsumed relationships between different elements (unigrams, n-grams and lexicon-syntactic patterns). The idea is that if an element is subsumed by another, the subsumed element is not needed, something that can remove redundant elements in the subjectivity classification (Liu, 2012).

2.4 Sentiment Classification

Broadly speaking, in data mining or machine learning a classification approach uses prior knowledge (e.g. documents or reviews) as training data to learn a model that can automatically classify new, unseen data. When labelled training data are available, the approach applied is supervised learning, otherwise in the case of unlabelled data an unsupervised learning approach is followed, or other hybrid or semi-supervised learning methods are considered. In this context, (Pang et al., 2002) argue that machine learning classification methods work well in sentiment analysis tasks. The authors argue that supervised learning fits a sentiment classification task as in a document-level classification. However, one should keep in mind that these models are highly dependent on the quality of the training data. Other researchers have also proposed sentiment classification algorithms for the subjective and sentiment classification problem (Hu & Liu, 2004b; Kim & Hovy, 2004; Turney, 2002; Yu & Hatzivassiloglou, 2003).

(Lourenco Jr. et al., 2014) propose an online sentiment classification method and argue that previous approaches lean towards offline classification. For the authors' work this is a critical point, as in their approach it's necessary to produce tweets sentiment judgements in real time. To this end, an alternative classification strategy was proposed: to insure fast learning times the training sets are kept as small as possible. To this aim, a set of association rules that are used for sentiment scoring is described. More formally, in this work a set of rules from the vocabulary training set $\mathcal{D}_n$ was used, in which a classifier $\mathcal{R}(t_n)$ at each time step n is defined. For a given message t_n a rule is valid if it's applicable to the respective message. Nevertheless, it should be kept in mind that opinions context tends to drift over time. Therefore, the quality of rules' coverage requires a reasonable amount of work and might also require maintenance rules. Besides, as (Wiebe & Riloff, 2005) mention, rule-based classifiers do not involve learning but merely classify sentences by observing state-of-the-art polarity characteristics that have been previously published. However, in (Lourenco Jr. et al., 2014) at each time step n the classifier updates its vocabulary and sentiment drifts (e.g. polarity changes for the same entity), something that slightly differs from traditional rule-based approaches.

2.5 Aspect-based Sentiment Analysis

The task of detecting overall sentiment, opinion holders and targets implies several steps (Liu, 2012). In a sentence-level sentiment analysis approach, (Meena & Prabhakar, 2007) showed that rules based on atomic sentiments of individual phrases can be helpful to decide on the overall sentiment of a sentence. However, in (Meena & Prabhakar, 2007), only adjectives and verbs were considered as features, which implies that only those can be related to the opinion target. Furthermore, as (Wilson et al., 2009) showed, other word families (e.g., nouns) may share dependency relations with opinion targets (also referred as aspects), which might be indicative of the sentiment expressed towards those terms. In another work by (Gildea & Jurafsky, 2002), the authors introduced a system based on statistical classifiers to identify semantic relationships. Their system analyses the prior probabilities of various combinations of semantic roles (predicate verb, noun, or adjective) to automatically label domain-specific semantic roles such as Agent, Patient, Speaker or Topic. Similarly to the semantic roles detection introduced by (Gildea & Jurafsky, 2002), we propose to analyse sentences lexical and syntactic relations to automatically label opinion targets.

3 SENTIMENT ANALYSIS IN MULTISENSOR

MULTISENSOR has established three pilot use cases: Journalism, commercial media monitoring, and small or medium-sized companies (SMEs) internationalisation. The first use case covers journalists interests that are challenged by the amount of information made available by various data streams, which are made available from known and unknown sources. The second use case addresses the commercial media monitoring client situation, while the third use case tackles the challenge of SMEs initiating or interested in the process of internationalisation with various products. Data streams reflect business decisions in which it is required to consider a variety of market information from multiple angles. Here, SMEs intend to expand to foreign markets that are challenged by not knowing relevant sources of information, existing trends and views, and public opinions on products (described in more detail in deliverable D8.2).

Bellow we will give more focus to the sentiment analysis module and describe its user requirements. The user requirements from the journalistic point of view and commercial media monitoring are guided by the description available in deliverable D8.2.

3.1 Motivation and user requirements

Use Case 1: Journalism

- **User search and visualization:** The software-based process of identifying the subjectivity and the polarity of a piece of text reflects the author’s opinion towards a specific subject. As described in section 2.2 sentiment analysis can be performed at a document-, paragraph- or sentence-level and can target specific entities that are mentioned in the text. To this end, the output will be a set of scores in a predetermined scale. While searching in social networks sentiment analysis may provide a very simple sentiment graphic over time of its findings (as, for example in regards of a brand). The algorithm will detect sentiment by considering the words surrounding the search term, as well as the sentiment semantic relations of the search term. This analysis might help in the detection of trends in news content with regards to politicians, celebrities, state representatives, among other entities or events. Hence, this type of analysis might prove to be valuable for journalist work therefore, not only for business reasons.
- **Sentiment extraction:** Extract words, sentences and whole news articles and categorize them into positive, neutral or negative groups. Also, highlight sentiment findings about entities (i.e. people or places).
- **Contributor analysis:** Creation of a sentiment lexicon for domain-specific sentiment words and the sentiment influence of popular entities.
- **Correlation:** Analyse news metadata and identify sentiment metadata relations between news (e.g. topics that relate to problematic events).

Use Case 2: Commercial media monitoring

- **User search and visualization:** The recent rise of social media and the exponential growth of the data produced necessitate the delivery of at least partly automated processes and tools that can help the modern analyst cover and assert multiple media sources (e.g., sentiment, topics, key influencers) in near real-time in order to succeed. However, to go beyond social networks (Use Case 1), the sentiment analysis module targets the Internet in general. Here, editors working at media monitoring companies need to filter large amounts of data. And, to this end, sentiment analysis module might provide sentiment filters that will help editors visualize the most relevant information and, consequently, reduce the number of decisions they have to make.

- **Correlation:** Analysts at media monitoring companies track opinions about the products a client is interest in. Here, the sentiment analysis module can help track sentiment-based topics and influencers present in the discussions about their clients. To produce a valuable insight for the client, our module can correlate the sentiment topics and express different sentiment associations that will help analysts' tasks.
- **Further analysis:** Understand the tone of the conversation (positive, negative or neutral) and unveil underlying topics of conversation. This analysis will allow the users to monitor content and identify negative associations, promote campaign ideas, etc.

Use Case 3: SME internationalisation

- **User search and visualisation:** SME internationalisation deals with the process of expanding from regional or a national market to a new foreign market. SMEs aim to increase their sales and diversify the risk, hence, maximize their productivity and expand their market broad. To this end, an effective and efficient sentiment analysis can be an invaluable tool in identifying potential pitfalls or opportunities for launching new products, as well as assess the public presence of SMEs in online news and social media.
- **Correlation:** Explore how sentiment about the SMEs varies according to the country (using news content geographic metadata) over time (temporal dimension). Geographic analysis (spatial dimension) allows the detection of tendencies about SMEs in different locations. However, the sentiment might also be influenced by age groups. Furthermore, it supports decision-making on opening new markets by offering an additional layer of information (sentiment analysis) on existing views and trends with regard to specific products and services.

3.2 Sentiment Analysis module

Before describing the sentiment analysis steps, we provide an overview of the sentiment analysis pipeline. Figure 2, taken from deliverable D7.2, presents all the main components of MULTISENSOR including the sentiment analysis module. The module is described in deliverable D7.2 as:

> **Sentiment Analysis Service [WP3]**
>
> Detects sentiments and their strengths in English text and annotates the text with this polarity and sentiment information. Analysis will include detection of sentiment towards entities. Generated facts will be expressed as RDF.

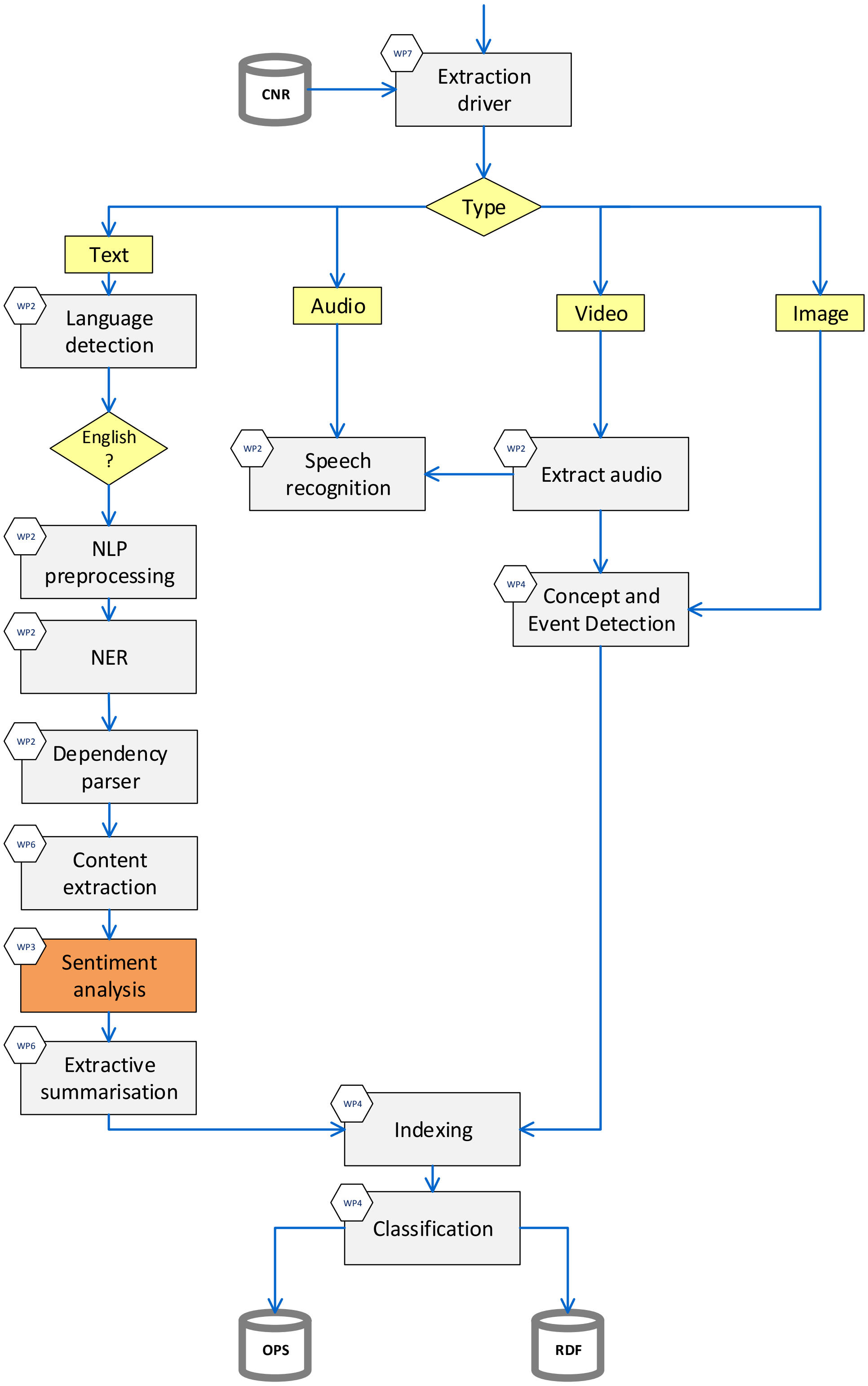


Figure 2: MULTISENSOR architectural pipeline.

In Figure 3, the architecture and interactions among the three main modules of WP3 are highlighted. As depicted in Figure 2, the textual content and social network analysis data (e.g., tweets) are forwarded to the sentiment extraction module. Sentiment annotated content is sent to the Information propagation and social interaction analysis module, while sentiment enriched object is sent to the context extraction module. Finally, it's shown that at the end of the WP3 pipeline the generated data is stored in repositories for later use by the client applications. Here, all information regarding the sentiment analysis module will be represented and updated in RDF and stored in the RDF repository.

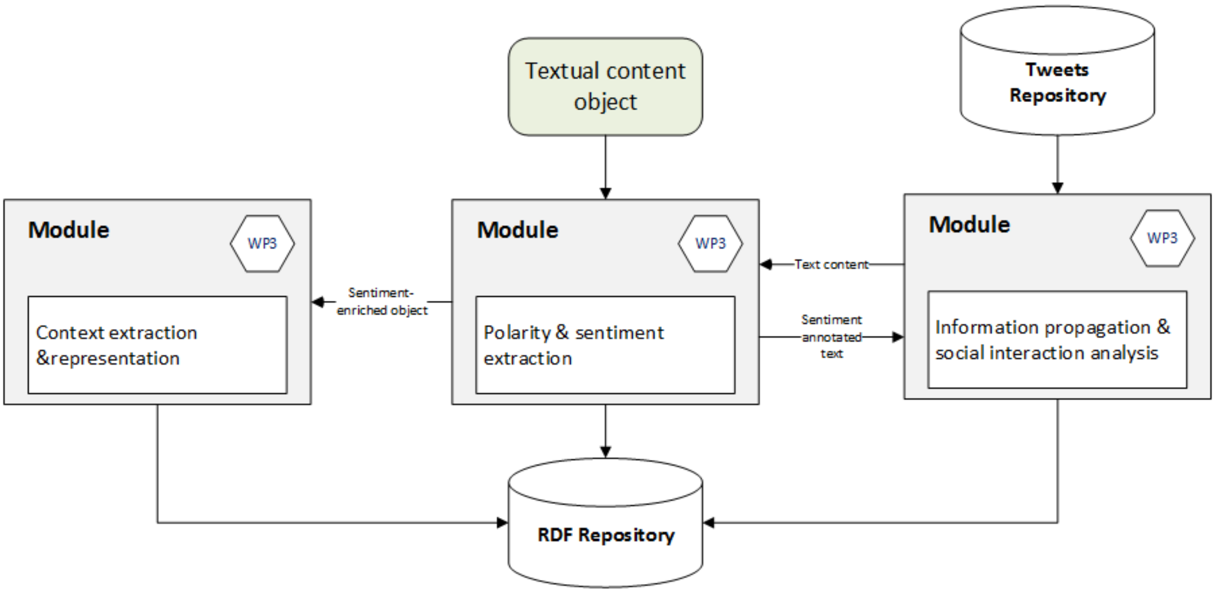

Textual content object
Tweets Repository
Module
WP3
Context extraction &representation
Sentiment-enriched object
Module
WP3
Polarity & sentiment extraction
Text content
Sentiment annotated text
Module
WP3
Information propagation & social interaction analysis
RDF Repository

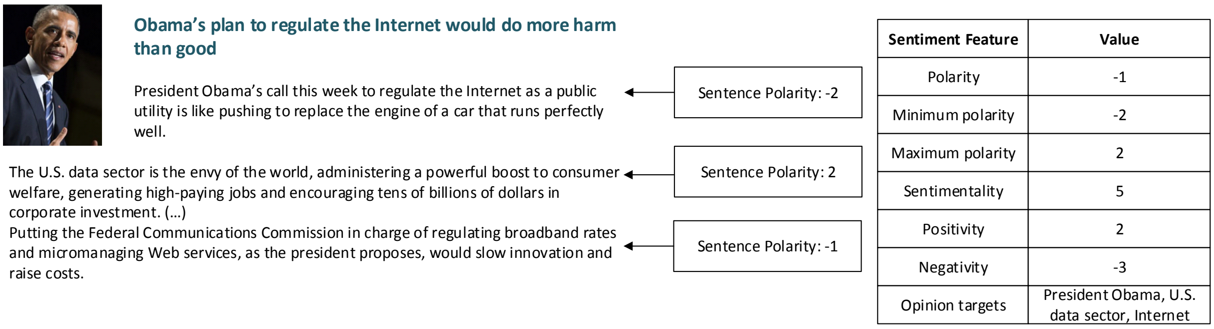
Figure 3: WP3 architecture.

3.3 List of Sentiment Features

Guided by the high-level scientific objectives and the specified user requirements described in Section 3.1 we propose a list of sentiment features that characterize the sentiment within news content that is consumed by MULTISENSOR. Given a news article we will provide the following sentiment features:

- **Polarity:** is computed as the sum of the positive and negative scores associated with a sentence or a document (full body news content).
- **Minimum polarity:** document sentence with the polarity lowest score.
- **Maximum polarity:** document sentence with the polarity highest score.
- **Sentimentality:** is computed as the sum of the absolute values of the positive and negative scores associated with a sentence or a document (full news content).
- **Positivity:** is computed as the sum of the positive scores associated with a sentence or a document.
- **Negativity:** is computed as the sum of the negative scores associated with a sentence or a document.
- **Opinion targets:** the existing opinion targets at sentence- and document- level are extracted.

In Figure 4 an example for a news article is shown that illustrates how the abovementioned sentiment features will be extracted.



Obama's plan to regulate the Internet would do more harm than good

President Obama's call this week to regulate the Internet as a public utility is like pushing to replace the engine of a car that runs perfectly well.

The U.S. data sector is the envy of the world, administering a powerful boost to consumer welfare, generating high-paying jobs and encouraging tens of billions of dollars in corporate investment. (...)
Putting the Federal Communications Commission in charge of regulating broadband rates and micromanaging Web services, as the president proposes, would slow innovation and raise costs.

Sentiment Feature	Value
Polarity	-1
Minimum polarity	-2
Maximum polarity	2
Sentimentality	5
Positivity	2
Negativity	-3
Opinion targets	President Obama, U.S. data sector, Internet

Figure 4: Polarity and sentiment extraction in a news article about Internet regulation.

In addition to the features extracted by the sentiment extraction module we might provide the following additional features:

- **Entities reputation**: Popular entities have a tendency to be mentioned as esteemed or disdained entities. These entities have an impact on the sentiment expressed in the news content and influence other entities.
- **Entities sentiment relations**: Entities are semantically related to sentiment words, places, entities and these relations can prove to be valuable in the analysis of the sentiment expressed in a news article. This analysis will leverage on characterizing an entity by its sentiment context.

4 SENTIMENT ANALYSIS EXTRACTION

Sentiment features express the sentiments in English text (see D7.1, section 3.2.2). Therefore, this analysis includes the detection of sentiments towards entities and generated facts which are later annotated in a RDF file. As in the following example,

Sentence:

```
public static final URI NEGATIVE_SCORE_VALUE;
public static final URI POSITIVE_SCORE_VALUE;
public static final URI SENTIMENTALITY_VALUE;
public static final URI POLARITY_VALUE;
```

News:

```
public static final URI OPINION;
public static final URI MIN_POLARITY_VALUE;
public static final URI MAX_POLARITY_VALUE;
```

In this section, we will detail how we aim to extract and weight the set of the proposed sentiment features (section 3.3). For each feature different methods will be explored.

4.1 Sentiment Words and Sentiment-Phrases Extraction

To evaluate the sentiment in media content that is consumed by MULTISENSOR such as news articles or blogs two main tasks are important: the identification of relevant sentiment words and its sentiment weight.

Corpus-based approach

The most elementary representation of a sentiment word is the single word (unigram). (Pang & Lee, 2004) notice that unigrams represent fairly good results in relation to bigrams or adjectives-only sentiment words representation. However, unigrams might fail to capture numerous sentiment words such as *basket case* (Liu, 2010). For that reason, in order to represent sentiment words we propose to use unigrams and bigrams (adjective-word pair). The bigram representation is computed as follows:

1. Adjectives influence the following word(s) by influencing on its positive or negative sentiment weight.
2. In a sentence, a word will pair with an adjective if it occurs after the adjective within a distance of 3 words.

It's expected that the full set of possible adjective-word pairs (bigrams) becomes too large and, as a consequence, not useful in capturing sentiment-based pairs. A possible method to overcome this problem is to apply the mutual information criterion to quantify the sentiment influence of each bigram,

$$MI(adjective-word) = \frac{freq(adjective-word)}{freq(adjective)\cdot freq(word)}, \tag{4.1}$$

where $freq(\cdot)$ represents the occurrence frequency.

The semantic orientation of an sentiment word details whether it's positive or negative and, a state-of-the-art technique is to apply the point-wise mutual information (PMI) to estimate the degree of statistical dependence between two words (Turney, 2002; Turney, 2001). PMI evaluates the probability of two words co-occurring together or individually. This measure is known for its ability to measure the strength of semantic associations (Turney & Littman, 2003). In this context, the semantic orientation of a sentiment word can be computed by observing its co-occurrence with a positive and negative word reference.

$$SO(word) = \log_2\left(\frac{hits(word,\text{"}excellent\text{"}) \cdot hits(\text{"}poor\text{"})}{hits(word,\text{"}poor\text{"}) \cdot hits(\text{"}excellent\text{"})}\right), \tag{4.2}$$

where $hits(word)$ and $hits(word,\text{"}excellent\text{"})$ are given by the number of hits a search engine returns given these keywords as search queries.

Dictionary approach

SentiWordNet is a popular sentiment lexicon which was introduced by (Esuli & Sebastiani, 2006). SentiWordNet is a lexicon created semi-automatically by means of linguistic classifiers and human annotation. In SentiWordNet[6] each synset is annotated with its degree of positive, negative and neutral relevance. Another popular sentiment lexicon is SentiStrength[7], which learns the sentiment word scores based on a function that describes words and phrases sentiment.

Rule-based opinion targets

(Moghaddam & Ester, 2012) proposed a set of rules to extract semantic relationships between words. These have proven to be quite successful in asserting semantic relations between opinion phrases. Table 1 shows the applied rules. For example, rules number 1 and 5 are able to extract the opinion phrases (*works*, *amazing*) and (*small*, *blurry*) from sentences *"The auto-mode works amazing."* and *"The LCD is small and blurry"* respectively. The proximity between an opinion target and a single sentiment word is key to building the opinion target semantic roles.

Table 1: Patterns to capture opinion-phrases (N is a noun, A is an adjective, V is a verb, h is a head term, m is a modifier, and $\langle h,m\rangle$ is an opinion phrase).

Opinion-phrase pattern
1. $amod(N,A) \longrightarrow \langle N,A\rangle$
2. $acomp(V,A) + nsubj(V,N) \longrightarrow \langle N,A\rangle$
3. $cop(A,V) + nsubj(A,N) \longrightarrow \langle N,V\rangle$
4. $dobj(V,N) + nsubj(V,N0) \longrightarrow \langle N,V\rangle$
5. $\langle h1,m\rangle + conj\ and(h1,h2) \longrightarrow \langle h2,m\rangle$
6. $\langle h,m1\rangle + conj\ and(h1,h2) \longrightarrow \langle h,m2\rangle$
7. $\langle h,m\rangle + neg(m,not) \longrightarrow \langle h,not + m\rangle$
8. $\langle h,m\rangle + nn(h,N) \longrightarrow \langle N + n,m\rangle$
9. $\langle h,m\rangle + nn(N,h) \longrightarrow \langle n + N,m\rangle$

An important step to extract opinion targets in news articles is to understand how a sentiment word is semantically related to an opinion target. To this end, we propose to

[6] http://sentiwordnet.isti.cnr.it/

[7] http://sentistrength.wlv.ac.uk/

implement a sentence-level method that will identify the sentiment words and sentiment phrases. Figure 5 provides an example of how we aim to extract opinion targets from a sentence. To resolve this problem we suggest dealing with the task of identifying opinion targets as a sequence labelling problem. The problem of opinion target extraction as a sequence labelling task can be resolved using the conditional random fields (CRF) method. Hence, given a sequence of tokens, $x = x_1 x_2 \dots x_n$ it's required to generate a sequence of labels $y = y_1 y_2 \dots y_n$. To build the model, a set of labels are defined as sentiment words and opinion targets. Similarly to (Choi et al., 2005) opinion holders detection model, we aim to create a linear-chain CRF based on a undirected graph $G = (V, E)$, where for each n tokens of a sentence V is the set of random variables $Y = \{Y_i | 1 < i \leq n\}$ is the set of $n - 1$ edges forming a linear chain. To this aim, this model will capture opinion targets from news content.

Sentence: The U.S. data sector is the envy of the world, administering a powerful boost to consumer welfare, generating high-paying jobs and (…).

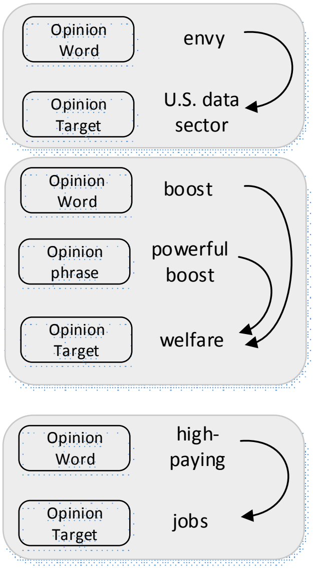


Figure 5: An overview of the opinion target extractor.

4.2 Sentiment Classification of Online News Articles

Given the plethora and diversity of classifiers available and the numerous examples of their successful application in the domain of NLP (Natural Language Processing), we intend to evaluate several representative algorithms (according to the task) and select the best performing, according to standard IR performance metrics. Let's consider a set of D news articles $D = \{d_1, \dots, d_i\}$ containing news articles about a particular event. Each piece of news is represented by a set of words or bigrams (pair adjective-word). To this end, the sentiment classifier aims to learn a model that is able to predict the sentiment value for the different news content granularity: full-news, sentence and opinion targets. In Table 2 we present three classifiers that are used for sentiment classification.

Table 2: Classification algorithms.

Support Vector Machines (SVM)	The SVM algorithm aims at linearly separating the features with a hyperplane. The features projected

	near the hyperplane limits will be selected. Support vectors define the optimal division between the categories (Joachims, 1998).
RIPPER	This algorithm identifies the class (or category) by building a set of decision rules (Cohen & Singer, 1999). RIPPER uses the technique of *direct representation* where each document is represented by a list of features without selecting a subset of the most relevant features. This algorithm contemplates the absence and presence of a feature.
Linear Regression (Vowpal Wabbit[8])	This classifier uses a linear classifier to assign a prediction. Vowpal Wabbit (VW) implements a stochastic gradient-descent classifier for optimizing the square loss of a linear model. In addition VW operates in an entirely online fashion overcoming practical limitations such as efficiency and scalability (Yuan et al., 2011).

4.2.1 Evaluation metrics

For evaluation we will apply the metrics precision, recall and F-measure. Bellow, the metrics definitions are provided, which are also included in deliverable D8.1. (Section 3.1).

Precision

In the field of information retrieval (Baeza-Yates & Ribeiro-Neto, 2011), precision is the fraction of retrieved documents that are relevant to the query. Given a document collection *R* and a query statement *Q,* we identify the subset of documents Rel that are relevant to it and the subset of documents $Retr$ that have been retrieved based on it. The retrieved documents that are relevant capture the number of documents in the intersection of $Retr$ and Rel, i.e., $Retr$ ⍰ Rel. In this case, precision is the proportion of retrieved documents that are actually relevant:

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \qquad (4.3)$$

The value of precision ranges between 0 and 1. A precision equal to 0 corresponds to the case where no relevant document has been retrieved while a precision equal to 1 corresponds to the case where all retrieved documents are also relevant.

Recall

[8] https://github.com/JohnLangford/vowpal_wabbit/

Recall in information retrieval captures the fraction of the documents that are relevant to the query and are successfully retrieved. Given a document collection *R* and a query statement *Q* we identify the subset of retrieved relevant documents $Retr$ ⍰ Rel and the subset of relevant documents *Rel* in *R*. In this case, recall is the proportion of retrieved relevant documents over all relevant documents:

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \qquad (4.4)$$

Similar to precision, the value of recall also ranges from 0 to 1, where 0 refers to the case where no relevant document is retrieved and 1 refers to the case where all relevant documents are retrieved.

F-score

The F-score is often used in the field of information retrieval for measuring search, document classification, and query classification performance. The F-score has also been widely used in natural language processing research, such as the evaluation of named entity recognition and word segmentation. In binary classification, the F-score (also known as F-measure) is a measure of a test's accuracy. It considers both precision and recall of the test to compute the score. The F-score can be interpreted as a weighted average of the precision and recall, where an F-score reaches its best value at 1 and worst score at 0.

The traditional F-measure or balanced F-score is the harmonic mean of precision and recall:

$$F - score = \frac{2TP}{2TP\ + FP + FN} \qquad (4.5)$$

4.3 Initial Results

The goals of the experiments are firstly, to see how accurately we can perform a sentiment classification task, and secondly, to examine the correlation between opinion phrases and opinion targets. For this analysis, one challenge to overcome is the lack of labelled data. To this end, we have selected a labelled dataset from SemEval-2014 challenge. This dataset contains opinionated sentences from the restaurants domain and it is part of Task 4: *Aspect Based Sentiment Analysis* of the abovementioned challenge[9]. In addition, the dataset contains a total of 1601 annotated sentences, out of which 1198 and 403 are positive and negative respectively. Furthermore, the dataset presents a mean of 66 characters and 12 words per sentence.

Sentiment classification

For our sentiment classification task, the sentences are classified according to a deterministic binary classification in which sentences are classified as either positive or negative. To classify the sentences we applied a 10-fold cross validation using the Weka (Hall et al., 2009) implementation of SVM. Table 3 shows the initial sentiment classification results.

[9] http://alt.qcri.org/semeval2014/task4/

Table 3: Sentiment classification of comments from restaurant reviews.

Polarity	Precision	Recall	F-score
positive	0.792	0.915	0.849
negative	0.583	0.331	0.422

5 SOCIAL NETWORK ANALYSIS

In this section we review social interaction analysis and information propagation analysis techniques. The aim is to understand the patterns of activity of users in social networks, mine their profiles and interests, and discover the interconnections and interaction patterns among users. We apply community detection, link analysis, and information propagation techniques in order to gain insights into groups of users that are interested in the same topic (or media type/genre), and into the role of individual users within their online community. Specific metrics such as betweenness and modularity will be analyzed in the content of community detection algorithms. In addition, we develop methods to assess the authority and influence of a given user within a topic-based network of contributors.

5.1 Community Detection

The detection of online dynamic communities will assist the work of a journalist. Twitter analysis, for example, can be used for detecting main events and mainstream trends, such as political party meetings, by extracting all users that are interconnected with respect to their common interest to a topic. Moreover, the detection of influential communities shows the influential users, who are able to create a mainstream trend. Apart from the detection of events and influential users, a new topic of discussion may also be detected, through the analysis of social networks, offering to a journalist quick access to a novel news item. Since most community detection algorithms provide hierarchical structure of the detected communities, it is possible to identify sub-communities on a local level, such as the speech of a municipality representative. Therefore, the online community detection techniques offer multiple benefits to a journalist.

In this chapter we will provide a review of the existing community detection algorithms. Then we discuss the proposed framework for the detection of communities in online social networks. This framework is based on the SocialSensor project (FP7-287975), where a community detection algorithm is applied on each graph snapshot, defined by the user network of mentions (@user). For the implementation of our framework, we combine existing implementations from SocialSensor and igraph. The open source collection of functions igraph employs several community detection algorithms for Network Analysis in R, Python and C. The latest igraph version 1.0.1 (June 26, 2015) runs in R through the igraph R-package.

We present selected methods in community detection and its corresponding implementations in user-friendly environments. The following algorithms provide community structure in static graphs, which is a necessary tool for the extraction of dynamic communities in dynamic graphs, obtained usually from Social Media. A detailed survey in community detection in Social Media may be found in (Papadopoulos et al., 2012).

5.1.1 Girvan–Newman algorithm

The Girvan–Newman community detection algorithm (Girvan and Newman, 2002; Newman and Girvan, 2004) is a divisive hierarchical process, based on the edge betweenness centrality measure (Freeman, 1977), which may be quickly calculated following Brandes (Brandes, 2001). The edge betweenness is measured by the number of shortest paths that pass through a given edge and determines the edges which are more likely to connect different communities. The edge with the highest edge betweenness is removed and the remaining edges are re-assigned new edge betweenness scores. The process generates a dendrogram with the whole graph as root node and the graph vertices as leaves. In order to extract the detected communities, the modularity score is computed at each dendrogram cut, so as to be maximized. The modularity has been defined as follows (Newman and Girvan, 2004):

$$Q = \sum_i (e_{ii} - a_i^2), \qquad a_i = \sum_j e_{ij} \tag{5.1}$$

where e_{ij} are the elements of a $k \times k$ summetric matrix and k is the number of communities at which the graph is partitioned. The elements e_{ij} are defined as the fraction of all edges in the network that link vertices in community i to vertices in community j.

The Girvan–Newman community detection algorithm may be implemented by the igraph function: `cluster_edge_betweenness`

5.1.2 Modularity Maximization

The Girvan–Newman algorithm requires the maximization of a modularity function, as a stopping criterion, for the optimal extraction of communities. However, Clauset et al., have presented an alternative hierarchical approach for community detection, using the modularity function (5.1) as an objective function to optimize. Initially, all vertices are separate communities and any two communities are merged if the modularity increases. The algorithm stops when the modularity is not increasing anymore. The modularity function is defined as (Clauset et al., 2004):

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{5.2}$$

where A_{ij} is the (i, j) element of the adjacency matrix, m is the number of edges in the graph, k_i is the degree of node i and $\delta(i, j)$ is 1 if $i = j$ and 0 otherwise. The modularity function (5.2) coincides with (5.1) but may be generalized to any null model with expected number of edges between vertices i and j (Fortunato, 2010):

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - P_{ij}] \delta(c_i, c_j) \tag{5.3}$$

The modularity maximization algorithm of (Clauset et al., 2004) may be implemented by the igraph function: `cluster_fast_greedy`

5.1.3 Louvain method

The Louvain method (Blondel et al., 2008) is based on the maximization of the modularity presented in Eq. (5.2) and involves two phases that are repeated iteratively. In the first phase, each vertex forms a community and for each vertex *i* the gain of modularity is calculated for removing vertex *i* from its own community and placing it into the community of each neighbour *j* of *i*. The vertex *i* is moved to the community for which the gain in modularity becomes maximal. In case the modularity decreases or remains the same, vertex *i* does not change community. The first phase is completed when the modularity cannot be further increased. In the second phase, the detected communities formulate a new network with weights of the links between the new nodes being the sum of weights of the links between nodes in the corresponding two communities. In this new network, self-loops are allowed, representing links between vertices of the same community. At the end of the second phase, the first phase is re-applied to the new network, until no more communities are merged and the modularity attains its maximum.

The original implementation of the Louvain method appears also as the igraph function: `cluster_louvain`.

5.1.4 **Bridge Bounding**

Bridge Bounding (Papadopoulos et al., 2009) is able to identify a community locally, surrounding a seed node i. Starting from the seed node i, each neighbor j is assigned to the community of node i if the following two conditions are satisfied:

1. Node j is not a member of any other community.
2. The edge connecting node j to node i is not a bridge, i.e. a community boundary.

The criterion, whether an edge $(i \leftrightarrow j)$ between node i and node j is a bridge, is the "local bridging" of an edge:

$$b_L(i \leftrightarrow j) = 1 - \frac{z_{ij}^{(3)}}{\min\{k_i - 1, k_j - 1\}} \tag{5.4}$$

where $z_{ij}^{(3)}$ is the number of triangles containing the edge $(i \leftrightarrow j)$. The algorithm stops when it is not possible to add nodes in the community i.

Bridge Bounding was developed for the needs of the WeKnowIt project (FP7-215453) and its implementation may be found as an open-source project[10], where other local community detection algorithms are also implemented (Clauset, 2005; Luo et al., 2006; Bagrow, 2008).

5.1.5 **Infomap method**

The Infomap method (Rosvall and Bergstrom, 2008; Rosvall et al., 2010) is an information-theoretic approach for community detection. The Infomap method is based on the previous work of (Rosvall and Bergstrom, 2007), in which the authors showed that the problem of finding a community structure in networks is equivalent to solving a coding problem. In general, the goal of a coding problem is to minimize the information required for the transmission of a message. Initially, Infomap employs the Huffman code (Huffman, 1952) in order to give a unique name (codeword) in every node in the network. In contrast to the Louvain method, which maximizes the modularity of Eq. (5.2), Infomap minimizes the Shannon information (Cover and Thomas, 2012) required to describe the trajectory of a random walk on the network. A global information minimum (in bits) for the description of the trajectory of a random walk X on the network, with n states and corresponding probabilities p_i, is given by Shannon's source coding theorem (Cover and Thomas, 2012):

[10] https://github.com/kleinmind/bridge-bounding

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{5.5}$$

which is the Shannon information of the random walk X.

The objective function, which minimizes the description length of a random walk on the network (described by the corresponding sequence of codewords on each visited node), is called the "map equation" (Rosvall and Bergstrom, 2008; Rosvall et al., 2010), and is minimized over all possible network partitions $\mathbf{M}$:

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{L}) + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(\mathcal{P}^{i}) \tag{5.6}$$

The first term of Eq. (5.6) describes the entropy of the random walk movements between communities and the second part is the entropy of movements within communities (exiting the community i is considered a movement of the second term). The fraction of transitions within the i-th community is denoted by $p_{\circlearrowright}^{i}$ and $H(\mathcal{P}^{i})$ is the entropy of community $\mathcal{P}^{i}$. The probability $q_{\curvearrowright}$ that the random walk switches communities on any given step is:

$$q_{\curvearrowright} = 1 - \sum_{i=1}^{m} p_{\circlearrowright}^{i} \tag{5.7}$$

The computational procedure followed for the minimization of Eq. (5.6) is presented in the supporting Appendix[11] of (Rosvall and Bergstrom, 2008).

Originally, the Infomap method was implemented in C++[12] and later it appeared as the igraph function: `cluster_infomap`.

5.2 Centrality and Social Influence Analysis

The information shared on a social platform (be it likes, engagement or declaration of interest on products/brands) can be exploited to build more refined user profiles. For example, if a user follows/comments on sport news on the social platform, we may infer that he is interested in sports and exploit this information for better user profiling and monitoring. However, the mere knowledge of the set of users interested in sports is barely enough: it is intuitively clear that certain users may be much more ``important'' or ``central'' than others. They may stand out because they are prolific content creators, or attract a legion of followers, or frequently set new trends, or a combination thereof.

This subsection is devoted to the question of how to select the most important nodes in a social network (Sun and Tang, 2011). There is a large body of research in the literature about this question, which can be divided into centrality measures (which are used to rank users based only on their structural role in the network's underlying graph), and social influence measures (which incorporate additional data from users' activity logs and the propagation of information). In turn, both centrality and influence measures can be classified into local or global: local measures associate a score to each user that is typically based on their set of

[11] http://www.pnas.org/content/suppl/2008/01/10/0706851105.DC1/06851SuppAppendix.pdf

[12] http://www.mapequation.org/code.html

immediate neighbours, whereas global measures take the full network structure into account.

Below we summarize the main approaches.

5.2.1 Centrality Scores

Let $G = (V, E)$ denote a directed social network, where $(u, v) \in E$ represents the fact that the information can flow from u to v; for example, in Twitter it may mean that v is a follower of u.
Let $E^{+}(u) = \{v|(u, v) \in E\}$ denote the set of followers of u and, $E^{-}(u) = \{v|(u, v) \in E\}$ denote the set of user who can influence u.
Below we define some of the most commonly used centrality scores (Chakrabarti and Faloutsos, 2012).

- **Outdegree (number of followers)**: the number $|\, E^{+}(u)|$ of outgoing arcs. It is the most basic measure of ''prestige'' of a node.
- **Degree log-ratio:** essentially, the ratio between in-degree and out-degree. The score is higher if the node is a hub and can potentially influence a large number of other users, while other nodes do not easily influence the hub node. It is conveniently expressed in a logarithmic scale: $\gamma(u) = \log(\frac{|\, E^{+}(u)|+1}{|\, E^{-}(u)|+1})$,

  where the +1 terms are included to ensure the domain of definition includes all users.
- **Harmonic centrality (Boldi and Vigna, 2013):** $\gamma(u) = \sum_{v \neq u} \frac{1}{d(u,v)}$, where $d(u, v)$ denotes the shortest-path distance between u and v in the graph.
- **Betweenness centrality (Freeman, 1997):** is based on the total number of shortest paths between all pairs of vertices that pass through a node. Let's denote the number of shortest paths between vertices j and k by $\eta(j, k)$ and let's denote the number of such paths passing through u by $\eta_u(j, k)$. The betweenness centrality for node u is defined as $\sum_{j,k \neq u} \eta_u(j, k)/\eta(j, k)$.
- **Pagerank (Brin and Page, 1998).** This is one of the most widely known global centrality measures. In its basic form, it measures how often a node will be visited by a user performing a random walk in the graph (i.e., following outgoing links at random). More formally, the Pagerank score of all nodes in an n-vertex graph can be viewed as a vector $p \in \mathbb{R}^n$which by definition is the unique solution to

$$\mathrm{p} = \mathrm{\alpha pA} + (\mathrm{a} - \mathrm{\alpha})\mathrm{q} \qquad (5.8)$$

where A is the adjacency matrix of the graph (normalized so that all non-empty rows have unit $l_1$norm), and q is a preference vector (which must be a distribution).

5.2.2 Information Propagation

Interactions in social networks result in varying levels of influence. For instance, when someone tweets a message on Twitter, his followers are exposed to it. They may in turn retweet, resulting in the spread of information and opinions throughout the network. This phenomenon is studied via diffusion models. The problem of understanding the dynamics characterizing social influence and the influence-driven diffusion of information, also known as information cascades, in social networks is motivated by the following idea (Gladwell, 2000):

Ideas and products and messages and behaviours spread like viruses.

This motivates a wide range of applications for social influence analysis: word-of-mouth (viral) marketing, identifying influencers and promote their deeper engagement with the considered system, generating personalized recommendations based on those influencers, feed ranking, just to mention a few.

Propagation models describe how the diffusion of information happens (Easly and Kleinberg, 2010). The social network is modelled as a graph where each vertex represents a user. At a given time, each node is either active (a user which already adopted the item), or inactive and an active node never becomes inactive again. Time advances in discrete steps and each node’s tendency to become active increases monotonically as more of its neighbours become active.

Under the *Independent Cascade* (IC) model (Kempe et al, 2003), each new active node v at time t is considered contagious and it has one chance of influencing each inactive neighbour u, independently of the history thus far. The activation of u by v succeeds with probability p_{uv}; the activation trial is unique, as v cannot make further attempts to activate v in subsequent rounds.

By contrast, in the *Linear Threshold* (LT) model (Kempe et al, 2003), we are given a weight function $w_{u,v}$that vanishes on non-neighbouring pairs and satisfies $\sum_v w_{u,v} \leq 1$ for each node u. Then, every vertex independently selects a uniform random threshold $\lambda_u \in [0,1]$. Then, at each step, an inactive vertex u becomes active if the sum of the weights of its edges incoming from activated neighbours reaches the threshold λ_u.

In both models, the process runs until no more activations are possible. These propagation models naturally give rise to a measure of influence $\gamma(u)$ based on the *expected spread* of a node, i.e., $\gamma(u)$ is the expectation of the size of active of a random cascade from u under the model. Note that the input required is not only the unweighted graph of connections among nodes, but also the set of influence probabilities (IC model) or edge weights (LT model). A discussion of methods to estimate these may be found in (Goyal et al, 2011).

By way of example, consider Twitter as an application scenario, and assume the IC model. We assume that each user shares content and we rank them according to the expected number of reshares that they get. Users with higher scores are those able to trigger largest number of retweets.

Regarding efficiency considerations, while exact computation of the expected size of the spread for a single user in the IC model is *#P*-hard (Kempe et al, 2003) and hence infeasible, it can be approximated arbitrarily well by means of Monte Carlo simulation.

6 SOCIAL NETWORK ANALYSIS IN MULTISENSOR

6.1 Community Detection Module

The analysis of online social networks has attracted much attention over the last years (Marcus et al., 2011; McKelvey et al., 2012; Konstantinidis et al., 2013), but the state-of-the-art Infomap method (Section 5.1.5) is not involved at any stage.

TwitInfo (Marcus et al., 2011) is a platform that is used for the detection and visualization of topics-events on Twitter, which are defined as the tweets that contain the desired keyword. The detection of topics-events is done by identifying temporal peaks in the frequency of tweets. A peak is detected when the frequency of all topic-related tweets becomes greater than a given threshold that depends on the local mean and standard deviation of the frequency of tweets.

The Truthy platform (McKelvey et al., 2012) is a real-time social network platform for the analysis of Twitter posts that clusters tweets into “memes”, i.e. sets of tweets containing a common hashtag, mentioned username, hyperlink or phrase. Truthy provides interactive visualization of the meme diffusion network and user-oriented statistics. The detection of dynamic communities and topic-based influencers is done by filtering the top-twenty most retweeted users and their neighbours.

The SocialSensor framework (Konstantinidis et al., 2013) for the detection of communities in dynamic graphs utilizes the Louvain method (Blondel et al., 2008), which allows for detecting communities in static graphs. In the special case of Twitter, any two users are mutually connected if they share the same mentions. A user is a part of a community only for a certain period of time, subject to his/her increasing or decreasing interest for a topic. The Louvain method is applied to a sequence of graph snapshots $G_1, G_2, \dots, G_n, \dots$ and identifies the communities $C = \{C_{1n}, C_{2n}, \dots, C_{kn}\}$ that are present across n timeslots.

In contrast, the MULTISENSOR framework utilizes the state-of-the-art Infomap method (Rosvall and Bergstrom, 2008; Rosvall et al., 2010), described in Section 5.1.5, which is better than the Louvain method in terms of accuracy. The superiority of the Infomap method compared to the Louvain method is reported in (Lancichinetti et al., 2009; Fortunato, 2010).

6.1.1 Proposed framework

Starting from the graph snapshot G_1, the first set of communities $\{C_{11}, C_{21}, \dots, C_{k1}\}$ is extracted by applying the Infomap community detection algorithm (Rosvall and Bergstrom, 2008; Rosvall et al., 2010). Next, the second set of communities $\{C_{12}, C_{22}, \dots, C_{l2}\}$ is extracted from the graph G_2 and a matching process is performed between all the $k \times l$ community pairs, in order to test if a community appears in both graph snapshots G_1 and G_2. The measure used for testing the matching between communities is the Jaccard coefficient (Jaccard, 1912). If the Jaccard coefficient

$$J(C_{i2}, C_{i1}) = \frac{|C_{i1} \cap C_{i2}|}{|C_{i1} \cup C_{i2}|} \tag{5.9}$$

exceeds a matching threshold φ, the pair is matched and C_{i2} is added to the timeline of the dynamic community T_1. Following previous analysis (Greene et al., 2010), we set the threshold at $\varphi = 0.3$, and the process continues for all timeslots considered and the corresponding graph snapshots $G_3, G_4, \ldots$. Our framework, presented in Figure 6, slightly advances the SocialSensor framework (Konstantinidis et al., 2013) for community detection, due to the introduction of the Infomap method, applied on each graph snapshot $G_1, G_2, \ldots$ for community detection.

The most frequent states that communities might have are illustrated in Figure 7. The time-evolving communities are denoted by T_i and they are represented by a timeline of the communities it comprises. The most frequent community conditions are birth, death, merging, splitting, growth, decay and irregular occurrence. In Figure 7, for example, the dynamic community T_1, after timeslot $n-1$ splits into two dynamic communities, namely T_1 (the largest part of T_1 at timeslot $n-1$) and T_7. Despite that T_1 appeared also in previous timeslots (e.g. $n-2, n-3, \ldots$), the dynamic community T_2 is detected for the first time at timeslot $n-1$ (birth) and is merged with the dynamic community T_3, formulating a larger community, which continues on to timeslot $n+2$. Dynamic community T_7 decays and disappears at timeslot $n+2$, demonstrating the user's lack of interest after a certain period of time. Dynamic communities T_4 and T_6 are both created at timeslot $n-1$ and they both disappear at timeslot n. However, the community T_4 reappears at timeslot $n+1$ (irregular occurrence) while T_6 does not.

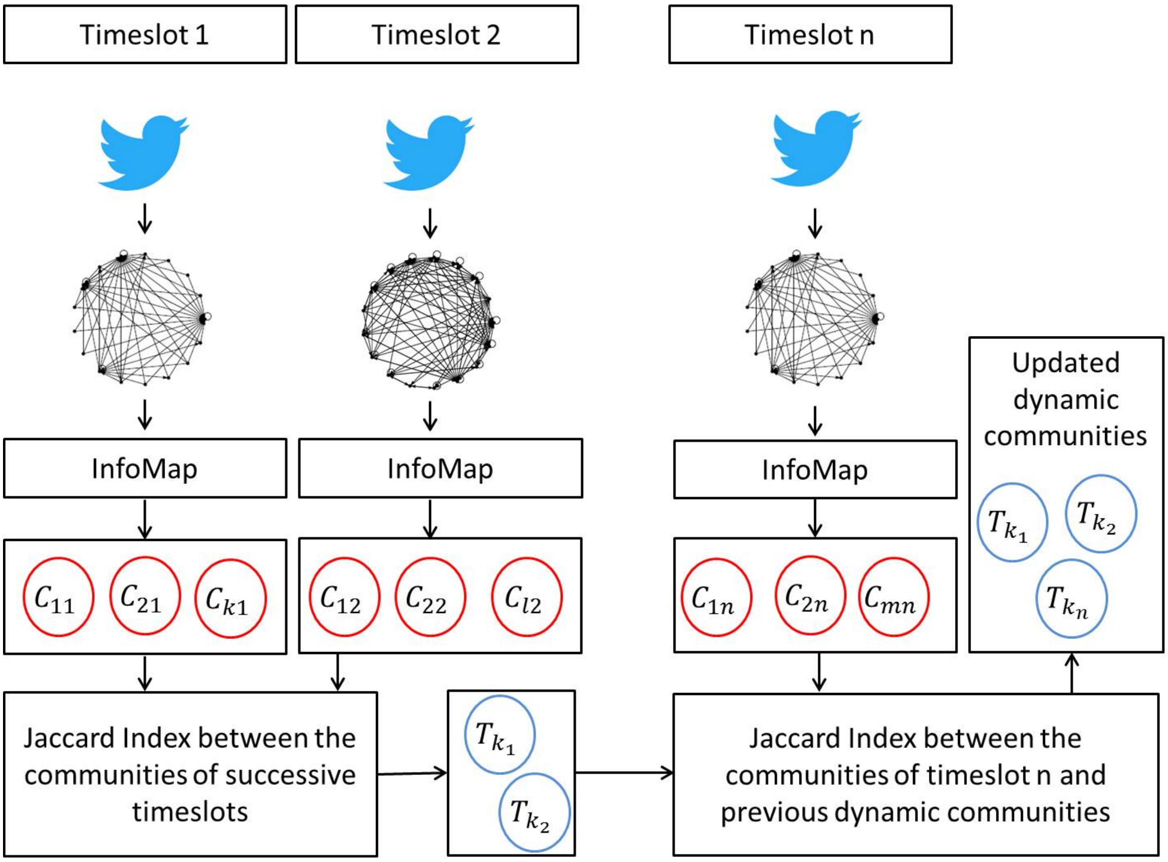

Timeslot 1
Timeslot 2
Timeslot n
InfoMap
InfoMap
InfoMap
C_{11}
C_{21}
C_{k1}
C_{12}
C_{22}
C_{l2}
C_{1n}
C_{2n}
C_{mn}
Updated dynamic communities
T_{k_1}
T_{k_2}
T_{k_n}
Jaccard Index between the communities of successive timeslots
T_{k_1}
T_{k_2}
Jaccard Index between the communities of timeslot n and previous dynamic communities

Figure 6: MULTISENSOR framework for community detection in online social media.

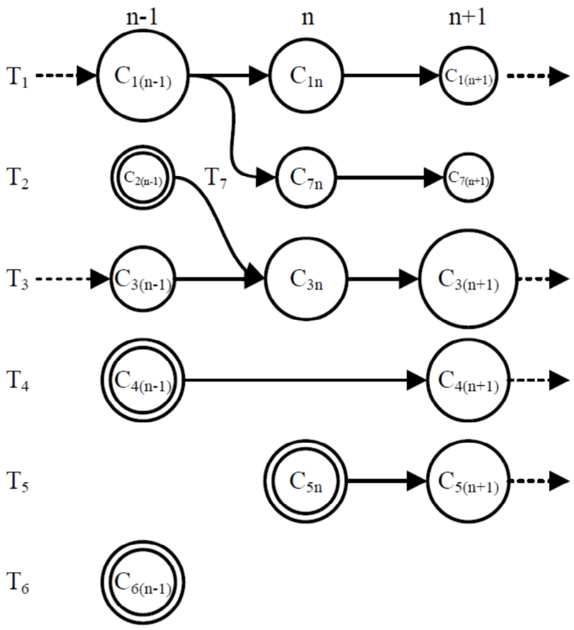


Figure 7: Time-evolving communities (Konstantinidis et al., 2013).

6.1.2 Community detection experiments

We collected Twitter posts from 28th July 2015 to 11th August 2015, with respect to the following hashtags:

- #energy_policy
- #energy_crisis
- #renewable
- #dishwasher
- #civilengineering
- #foodmanufacturing
- #homeappliances

We define timeslots as 24h-periods corresponding to daily snapshots of the users' network. The nodes of the graph are Twitter users and the links represent each mention of one to another. The active users in these 15 days, with respect to the above hashtags, are 9,563 and 23,485 posts were extracted. The posts that mention a user are 14,754 and therefore 14,754 users mention other users. The posts of users that mention themselves aiming to increase their impact on the network, are removed.

Links between users are extracted day by day. In Figure 8, we illustrate the sequence of graph snapshots $G_1, G_2, \dots, G_{15}$, where G_1 is the graph of the 28th of July 2015 and G_{15} is the graph of the 11th of August 2015. From Figure 8, it is obvious that the network structure changes over time, i.e. some communities appear and then disappear.

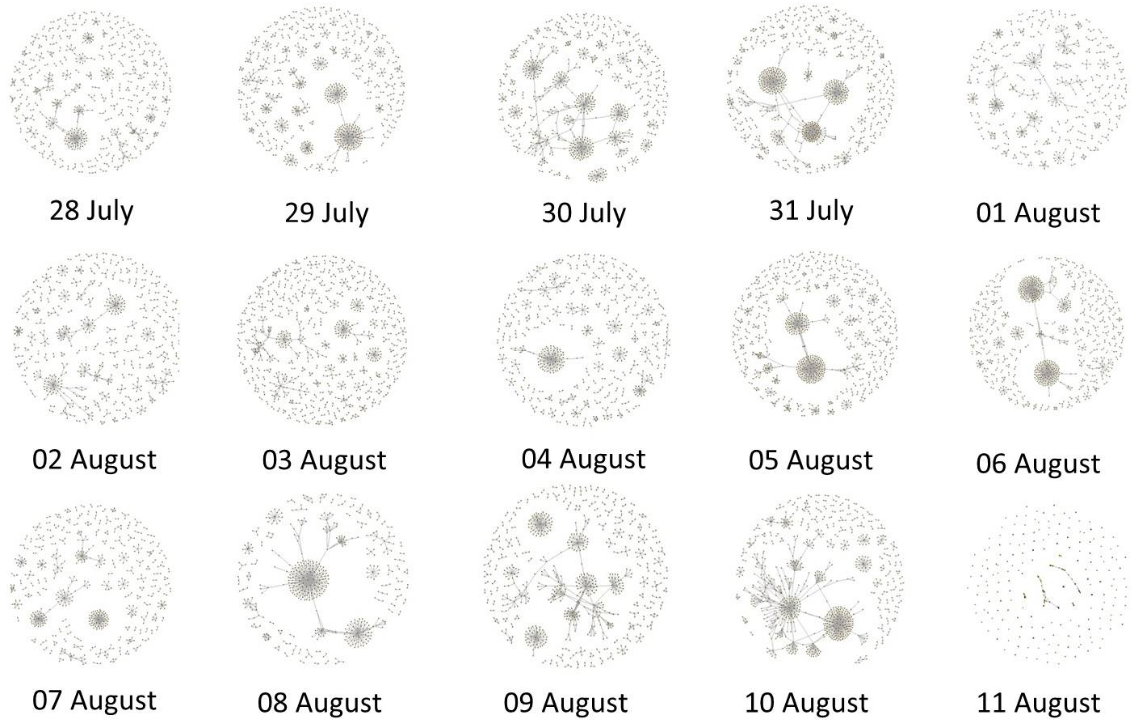


Figure 8: Graph snapshots from 28th July 2015 to 11th August 2015.

In order to detect communities in each one of the graph snapshots $G_1, G_2, \dots, G_{15}$, we employ the Infomap method, which was described earlier in Section 5.1.5. Trivial communities having less than 3 users are not considered. The framework we follow for community detection is independent from the language at which the Twitter post is written. The multilingual character of Twitter posts is presented in Table 4, where the 10 most popular languages are presented.

Table 4: The number of Twitter posts for each language.

English	22566	**Portuguese**	35
Spanish	106	**Danish**	33
Indian	80	**German**	23
Italian	43	**Dutch**	21
French	37	**Turkish**	19

The size and number of communities are critical parameters for the description of the network structure. We demonstrate the 3rd largest community of 28th July in Figure 9. In this community, we find Twitter users that are individual persons or companies and in their description we observe 3 languages (English, Spanish and French).

User	Description
MSSM	The @EarthInstitute's M.S. in Sustainability Management
S&C Electric Company	Welcome to the US Twitter feed for S&C Electric Company, a global provider for electric power systems. http://www.sandc.com/blogs/
IMMdesignlab	IMM is an international multidisciplinary design laboratory.
Primate Seven	I am an Earthling, like yourself, looking for fresh fruit and veggies, nuts and berries, clean air, water and land, peace and quiet, healthy trees, and friends.
WRI Climate	Updates from World Resources Institute's Climate Program. Tweets by Max Frankel & Rhys Gerholdt
UGT-FORESTALES	Sindicato Profesional de Agentes Medioambientales/Forestales de UGT. Defendemos y protegemos el Patrimonio Natural. #AAFF #AAMM #StopLeyMontesPP
The Honey Dive	Chelle Willes- English speaking/ activist/ inspirationalist/ retweetivist/ aspiring to do something with my life!
K. S. H.	ALL animals deserve happiness. Opinionated. TCK. I love felines! Earth's preservation ~~ or else. Insomniac.
Jeren	Beautiful, strong, independent and hard working.... C'est moi! Je suis désolé d'être grossier et parfois de ne pas répondre à temps\nParis Sorbonne UAD @psuad
John Lundin	Environmental activist, author of JOURNEY TO THE HEART OF THE WORLD and THE NEW MANDALA, editor of TheDailyPlanet and @TheEarthNetwork
twitfer	MEXICAN w/ a green mindset + developed taste for what is intellectually amusing, pleasant for the senses & describing of human nature. Huge fan of FC Barcelona.
Paulo Lopes Ferreira	Environmental policy and sustainability consultant

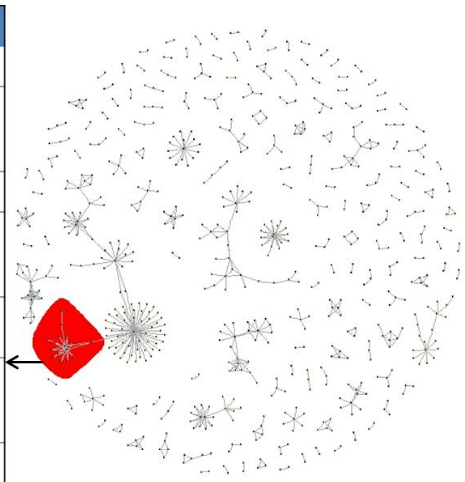


Figure 9: The 3rd largest community of 28th July.

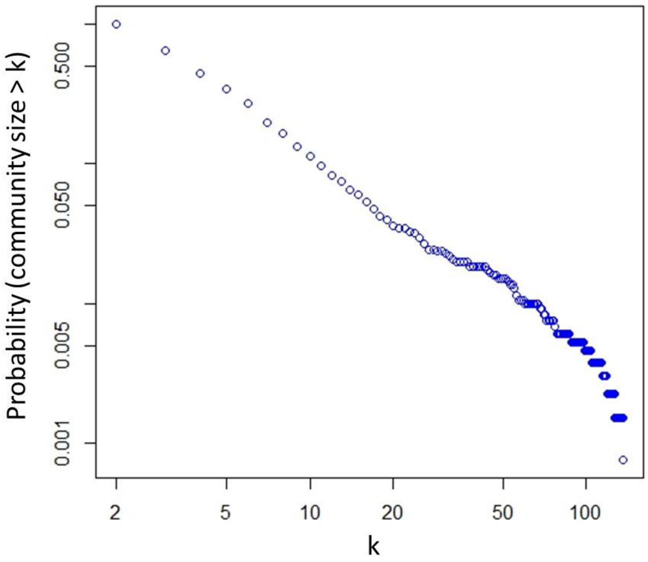


Figure 10: The distribution of the community size.

In Figure 10, we present the distribution of the community size in logarithmic scale. We see that the formulation of large communities is very rare and most of the communities are very small, so it is highly probable to observe communities with a small number of users.

The main novelty of our framework is the use of the Infomap method for community detection. The goal of Infomap is to minimize the Shannon Information needed to describe the trajectory of a random surfer, namely "codelength". The evolution of codelength in time (Figure 11) shows that the 1st and 2nd of August have the best partition into communities. In these initial experiments we present our framework and the performance of Infomap in time, with respect to the codelength. In the following deliverables, we shall compare our framework with existing works and more indices will be included in the analysis.

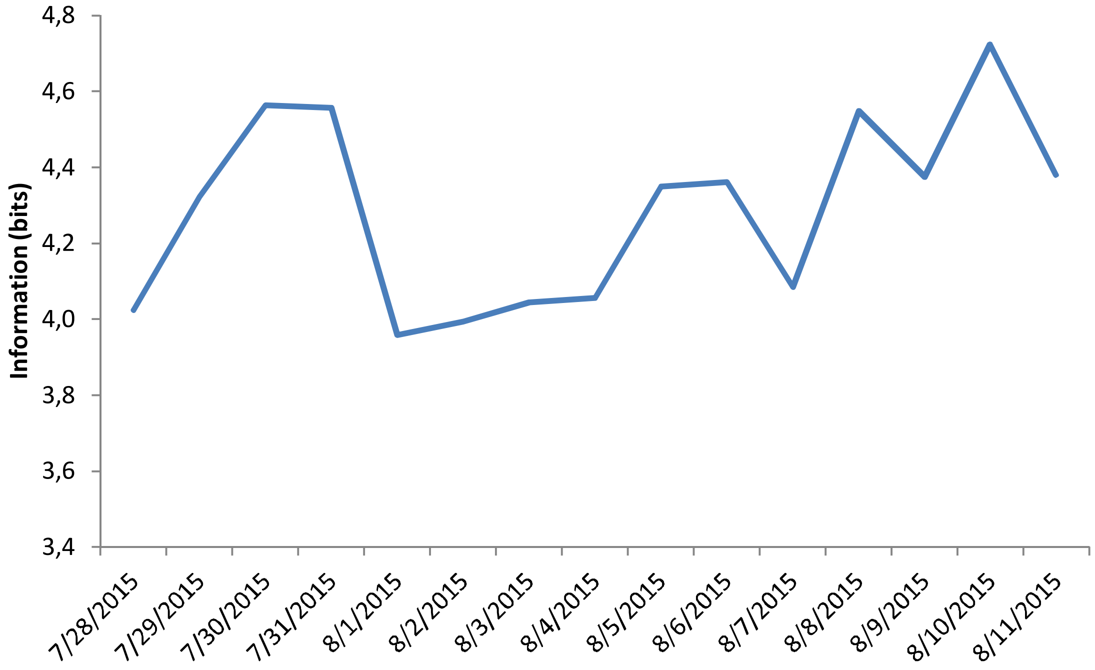


Figure 11: Codelength for each day.

In total, we detected 1,323 communities in the 15-day period from 28th July 2015 to 11th August 2015. However, most of them are static and their lifetime is restricted to one day. Using the framework of Section 6.1.1, we detected 60 dynamic communities, i.e. groups of users that are members of the same community for more than one day. The first 30 detected dynamic communities are reported in Table 5. Dynamic communities appear from the first two days (28th-29th July). The community $C_{29,2}$ of the 29th July is added to the timeline of the community $C_{12,1}$ of the 28th July. The dynamic community T_2 is the most persistent, since it lasts 8 days. The dynamic community T_{27} is the second most persistent and lasts 5 days. In general, most of the dynamic communities last 2 or 3 days.

Table 5: The first 30 detected dynamic communities.

Day:	1	2	3	4	5	6	7	8	9	10	11	12
T_1	$C_{12,1}$	$C_{29,2}$										
T_2	$C_{15,1}$	$C_{18,2}$	$C_{14,3}$	$C_{8,4}$	$C_{11,5}$	$C_{8,6}$	$C_{37,7}$	$C_{8,8}$				
T_3	$C_{53,1}$	$C_{71,2}$										
T_4	$C_{70,1}$	$C_{58,2}$										
T_5		$C_{38,2}$	$C_{39,3}$	$C_{45,4}$								
T_6		$C_{56,2}$	$C_{22,3}$									
T_7			$C_{40,3}$	$C_{19,4}$								
T_8			$C_{76,3}$	$C_{30,4}$								
T_9				$C_{10,4}$	$C_{16,5}$	$C_{12,6}$	$C_{8,7}$					
T_{10}				$C_{16,4}$	$C_{52,5}$							
T_{11}				$C_{22,4}$	$C_{50,5}$							
T_{12}				$C_{36,4}$	$C_{19,5}$	$C_{19,6}$						
T_{13}				$C_{72,4}$	$C_{25,5}$							
T_{14}				$C_{78,4}$	$C_{62,5}$							
T_{15}					$C_{55,5}$	$C_{65,6}$						
T_{16}						$C_{21,6}$	$C_{30,7}$					
T_{17}						$C_{31,6}$	$C_{82,7}$					
T_{18}						$C_{47,6}$	$C_{77,7}$					
T_{19}						$C_{48,6}$	$C_{40,7}$					
T_{20}						$C_{57,6}$	$C_{84,7}$					
T_{21}						$C_{61,6}$	$C_{91,7}$					
T_{22}						$C_{78,6}$	$C_{49,7}$					
T_{23}							$C_{69,7}$	$C_{48,8}$				
T_{24}							$C_{80,7}$	$C_{31,8}$				
T_{25}							$C_{108,7}$	$C_{61,8}$				
T_{26}							$C_{123,7}$	$C_{42,8}$				
T_{27}								$C_{5,8}$	$C_{13,9}$	$C_{18,10}$	$C_{19,11}$	$C_{14,12}$
T_{28}								$C_{7,8}$	$C_{17,9}$	$C_{62,10}$	$C_{8,11}$	
T_{29}								$C_{49,8}$	$C_{44,9}$			
T_{30}								$C_{57,8}$	$C_{47,9}$			

6.2 The Contributor Analysis Module

The contributor analysis performs the tasks related to information propagation and social influence and interaction analysis. It comprises two distinct submodules: the first one is used to compute local authority scores on the fly given a twitter handle by querying the Twitter API for information about the user and his immediate connections, whereas the second one performs computation of global authority scores relying on Twitter data that has been previously crawled.

6.2.1 Local Authority Scores

This submodule is comprised of several services. The Java code is available on the MULTISENSOR subversion repository[13].

The first service, TwitterSearchNamesForMultisensor, is used to discover users relevant to a given query.

The second service, TwitterCrawlerForMultisensorSingleKey, takes as input a Twitter screenname (e.g., "@barackobama"). Then it proceeds to extract the user's profile information via the Twitter API and outputs local measures of the user's authority based on 3 criteria: reach (number of followers and size of the ego network), relevance to a given set of keywords and resonance (the likelihood that the content shared by the considered user goes viral). More specifically, the output of the submodule is:

- General information about the user account (name, location, language, user description, number of tweets, number of friends);
- Outdegree (number of followers);
- Degree log-ratio (as in Section 5), which is referred to as network influence score;
- Retweet influence score: average fraction of followers that retweet a random post by the user. Here we exclude posts that the user himself is retweeting from someone else, as we wish to reward active involvement;
- A combined local influence score, defined as the sum of the network influence score and the retweet influence score;
- A set of scores indicating the user's main areas of interest, e.g., {news = 10.0 music = 20.0 government = 40.0 food-drink = 10.0 family=10.0}. For this, the module relies on a predefined list of interest categories (file interests_users) with associated keywords.

For demonstration purposes, we use the service Twitter Crawler. For Multisensor MultiKeys.java, which is a variant of the second one that uses several Twitter keys to perform the crawling. This has the advantage of being faster, as it softens the impact of Twitter rate limitations. However, the use of this variant may be limited to testing until we ensure that this practice does not contravene Twitter's terms of use.

6.2.2 Global Authority Scores

For global authority computations (Pagerank and influence), we use indexed Twitter data to retrieve twitters matching a certain set of tokens. This data is collected by CERTH using the

[13] http://wp3/ms-svc-contributorAnalysis/src/main/java/eu/multisensor/services/

Twitter data crawler from the SocialSensor project. This crawler makes use of Twitter's streaming API to produce JSON-encoded data containing the set of posts relevant to the given set of tokens, together with information about the profiles of the posters and the associations among them.

Given this data, our service builds a topic-dependent network of contributors based on the mentions in the set of monitored tweets. It also computes retweet probabilities between users in this network, and finally outputs two ranked lists of users, one by decreasing order of Pagerank and another one by decreasing order of influence in the Independent Cascade model (Section 5.2). The code for these computations is written in Python using the SciPy library.

7 CONCLUSIONS

This document provides a detailed view of the technical vision of the WP3 Sentiment Analysis and Social Network analysis. It consolidates the work proposed by the MULTISENSOR architecture specifications.

We described a number of sentiment features inspired by the requirements of the use cases and discussed the proposed algorithms that will be used to extract and evaluate the sentiment within news textual content. We conclude the sentiment chapter by discussing preliminary results.

We also developed tools to leverage social media in order to build a topic-dependent network of contributors and analyze it via the discovery of communities and the evaluation of the importance of users within the network, taking into account both the associations among users and the patterns of information propagation within the network.

Ongoing and future work will focus on extending the baseline algorithms and improve the accuracy and applicability of these methods. These advanced methods will be reported in D3.4. D3.4 will also include comparison with the baseline techniques according to the metrics reported in the self assessment deliverables (D1.1&D1.2).

8 REFERENCES

Aue, A. and Gamon, M. 2005. Customizing sentiment classifiers to new domains: a case study. Proceedings of Recent Advances in Natural Language Processing (RANLP) 1, 207–218.

Baccianella, S., Esuli, A., and Sebastiani, F. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC) 25, 2200–2204.

Baeza-Yates, R. and Ribeiro-Neto, B. 2011. Modern Information Retrieval (2nd ed.). Addison-Wesley Publishing Company, USA.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., and Jurafsky, D. 2004. Automatic Extraction of Opinion Propositions and their Holders. Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium on Exploring Attitude and Affect in Text (AAAI), 22–24.

Boldi, P., and Vigna, S. 2014. Axioms for centrality. Internet Mathematics 10 (3-4): 222-262.

Brin, S., and Page, L. 1998.. The anatomy of a large-scale hypertextual Web search engine. Social Network Data Analytics, Computer Networks and ISDN Systems, 30(1): 107–117.

Bross, J. and Ehrig, H. 2013. Automatic Construction of Domain and Aspect Specific Sentiment Lexicons for Customer Review Mining. Proceedings of the 22Nd ACM International Conference on Conference on Information Knowledge Management (CIKM), 1077–1086.

Cambria, E., Schuller, B., Xia, Y., and Havasi, C. 2013. New Avenues in Opinion Mining and Sentiment Analysis. IEEE Intelligent Systems 28, 2, 15–21.

Chakrabarty, D., and Faloutsos, C. 2012. Graph mining: laws, tools, and case studies. Synthesis Lectures on Data Mining and Knowledge Discovery, 7(1): 1-207. Morgan & Claypool Publishers.

Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. 2005. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP), 355–362.

Cohen, W.W. and Singer, Y. 1999. Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems (TOIS), 141–173.

Ding, X., Liu, B., and Yu, P.S. 2008. A holistic lexicon-based approach to opinion mining. Proceedings of the International Conference on Web Search and Web Data Mining (WSDM), 231–240.

Easley D., and Kleinberg, 2010. J. Networks, crowds, and markets: reasoning about a highly connected world. Social Network Data Analytics, Computer Networks and ISDN Systems, 30(1): 107–117.

Esuli, A. and Sebastiani, F. 2005. Determining the semantic orientation of terms through gloss classification. Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM), 617–624.

Esuli, A. and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. Proceedings of the 5th Conference on Language Resources and Evaluation (LREC) 6, 417–422.

Gildea, D. and Jurafsky, D. 2002. Automatic Labeling of Semantic Roles. Computational Linguistics, MIT Press Linguistics 28, 3, 245–288.

Gladwell, M. 2000. The tipping point: how little things can make a big difference. Little Brown, 2000.

Goyal, A., Bonchi, F., and Lakshmanan, V. S. 2011. Learning influence probabilities in social networks. Proceedings of the ACM International Conference on Web search and Data Mining (WSDM), 241-250.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. 2009. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11, 1, 10–18.

Hatzivassiloglou, V. and Mckeown, K.R. 1997. Predicting the semantic orientation of adjectives. Proceedings of the 35th Annual Meeting on Association for Computational Linguistics (ACL), 174–181.

Hatzivassiloglou, V. and Wiebe, J.M. 2000. Effects of adjective orientation and gradability on sentence subjectivity. Proceedings of the 18th Conference on Computational Linguistics (COLING) 1, 299–305.

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., and De Jong, F. 2011. Polarity analysis of texts using discourse structure. Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM), 1061–1070.

Hu, M. and Liu, B. 2004a. Mining and summarizing customer reviews. Proceedings of the tenth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 168–177.

Hu, M. and Liu, B. 2004b. Mining opinion features in customer reviews. Proceedings of the Association for the Advancement of Artificial Intelligence 19th International Conference on Artifical Intelligence (AAAI), 755–760.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. Machine Learning: ECML-98, 137–142.

Kamps, J., Marx, M., Mokken, R.J., and Rijke, M. DE. 2004. Using wordnet to measure semantic orientation of adjectives. Proceedings of 4th International Conference on Language Resources and Evaluation (LREC), 1115–1118.

Kempe, D., Kleinberg, J. and Tardos, E. 2003. Maximizing the spread of influence through a social network.. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 137–146.

Kim, S.-M. and Hovy, E. 2004. Determining the sentiment of opinions. Proceedings of the 20th International Conference on Computational Linguistics (COLING) 4, 1367–1373.

Lim, K.W. and Buntine, W. 2014. Twitter Opinion Topic Model: Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM), 1319–1328.

Liu, B. 2010. Sentiment analysis and subjectivity. Handbook of Natural Language Processing, CRC Press, Taylor and Francis Group.

Liu, B. 2012. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, Morgan and Claypool Publishers, 1–167.

Lourenco JR., R., Veloso, A., Pereira, A., Meira JR., W., Ferreira, R., and Parthasarathy, S. 2014. Economically-efficient Sentiment Stream Analysis. Proceedings of the 37th International ACM Conference on Research Development in Information Retrieval (SIGIR), 637–646.

Meena, A. and Prabhakar, T. V. 2007. Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis. Proceedings of the 29th European Conference on Advances in Information Retrieval (ECIR) 4425, 573–580.

Moghaddam, S. and Ester, M. 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. Proceedings of the 34th international ACM conference on Research and development in Information Retrieval (SIGIR), 665–674.

Moghaddam, S. and Ester, M. 2012. On the Design of LDA Models for Aspect-based Opinion Mining. Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM), 803–812.

Pang, B. and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the Association of Computational Linguistics (ACL), 271–278.

Pang, B. and Lee, L. 2008. Opinion mining and sentiment analysis. Journal of Foundations and Trends in Information Retrieval, Now Publishers Inc. 2, 1-2, 1–135.

Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 10, 79–86.

Popescu, A.-M. and Etzioni, O. 2005. Extracting product features and opinions from reviews. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP), 339–346.

Qiu, G., Liu, B., Bu, J., and Chen, C. 2009. Expanding domain sentiment lexicon through double propagation. Proceedings of the 21st international jont conference on Artifical intelligence, 1199–1204.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. 1985. A Comprehensive Grammar of the English Language. General Grammar Series, Longman.

Ramage, D., Hall, D., Nallapati, R., and Manning, C.D. 2009. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 1, 248–256.

Rao, D. and Ravichandran, D. 2009. Semi-supervised Polarity Lexicon Induction. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (COLING), 675–682.

Riloff, E., Patwardhan, S., and Wiebe, J. 2006. Feature subsumption for opinion analysis. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 440–448.

Riloff, E. and Wiebe, J. 2003. Learning extraction patterns for subjective expressions. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 105–112.

Scheible, C. and Schutze, H. 2012. Bootstrapping Sentiment Labels For Unannotated Documents With Polarity PageRank. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), 1230–1234.

Sun, J., and Tang, J. 2011. A survey of models and algorithms for social influence analysis. Social Network Data Analytics, Springer, 177–214.

Takamura, H., Inui, T., and Okumura, M. 2005. Extracting Semantic Orientations of Words using Spin Model. Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics (ACL) 47, 133–140.

Tang, H., Tan, S., and Cheng, X. 2009. A survey on sentiment detection of reviews. Journal of Expert Systems with Applications, Pergamon Press 36, 7, 10760–10773.

Titov, I. and Mcdonald, R. 2008. Modeling online reviews with multi-grain topic models. Proceedings of the 17th international conference on World Wide Web (WWW), 111–120.

Turney, P. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL), 417–424.

Turney, P. and Littman, M. 2002. Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Tecnical report ERC-1094, National Research Council of Canada ERB-1094, 11.

Turney, P.D. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. Proceedings of the 12th European Conference on Machine Learning (EMCL) 2167, 491–502.

Turney, P.D. and Littman, M.L. 2003. Measuring praise and criticism: Inference of semantic orientation from association. Journal of ACM Transactions on Information Systems (TOIS), ACM New York 21, 4, 37.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., and Mcdonald, R. 2010. The Viability of Web-derived Polarity Lexicons. Proceddings of the Annual Conference of the North American Chapter of the Association of Computational Linguistics (NAACL/HLT), 777–785.

Wiebe, J., Bruce, R., and O'Hara, T. 1999. Development and use of a gold-standard data set for subjectivity classifications. Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL), 246–253.

Wiebe, J. and Riloff, E. 2005. Creating subjective and objective sentence classifiers from unannotated texts. Proceedimgs International Conference Computational Linguistics and Intelligent Text Processing (CICLing), 486–497.

Wiebe, J.M. 1990. Identifying subjective characters in narrative. Proceedings of the 13th Conference on Computational Linguistics (COLING) 2, 401–406.

Wiebe, J.M. 1994. Tracking point of view in narrative. Journal of Computational Linguistics, MIT Press Cambridge 20, 2, 233–287.

Wilson, T. and Wiebe, J. 2005. Annotating Attributions and Private States. Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky (CorpusAnno), 53–60.

Wilson, T., Wiebe, J., and Hoffmann, P. 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-level Sentiment Analysis. Journal Computational Linguistics 35, 3, 399–433.

Wilson, T., Wiebe, J., and Hwa, R. 2004. Just How Mad Are You? Finding Strong and Weak Opinion Clauses. Proceedings of the 19th Association for the Advancement of Artificial Intelligence Conference on Artifical Intelligence (AAAI), 761–767.

Yu, H. and Hatzivassiloglou, V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 10, 3, 129–136.

Yuan, G., Ho, C., and Lin, C. 2011. Recent Advances of Large-scale Linear Classification. Computer 3, 1–15.