SENSE4US

Tools for evidence extraction from open data sources

Project acronym:	**SENSE4US**
Project full title:	**Data Insights for Policy Makers and Citizens**
Grant agreement no.:	**611242**
Responsible:	**Miriam Fernandez, Open University**
Contributors:	**Hassan Saif, Open University** **Leon Kastler, University of Koblenz**
Document Reference:	**D5.3**
Dissemination Level:	**PU**
Version:	**~~1.0~~FINAL**
Date:	**30/09/2015**

Project co-funded by the European Union under the Seventh Framework Programme

History

Version	*Date*	*Modification reason*	*Modified by*
0.1		Initial draft	Miriam Fernandez
0.2		Code Released	Hassan Saif
0.3		Functionality Description	Hassan Saif
0.4		Quality check	Aron Larsson
1.0		Final reviewed deliverable	Miriam Fernandez

Table of contents

Executive summary

D5.3 is a prototype deliverable. All the software and data produced as part of this deliverable is available under the Sense4us git repository (https://gitlab1.it-innovation.soton.ac.uk/).

In this document we describe the software produced as part of D5.3: (i) to represent and navigate the social media citizen's discussions and to extract their sentiment and (ii) to complement the data extracted from the social media citizen's discussions with Linked Open Data (LOD) information in order to provide more insightful information to the Policy Maker (PM).

The deliverable is therefore structured in three main parts:

1. The first part summarises SentiCircles, i.e., the Sense4us research and infrastructure used to analyse social media discussions. We describe the multiple improvements that have been made over this infrastructure in the last period and show how SentiCircles have been integrated into the overall Sense4us demo in collaboration with WP3.
2. The second part summarises the large-scale data-querying infrastructure developed by WP4. In particular we emphasise the different methods developed to extract connectivity structures from Open Data (OD) Sources.
3. The third part of this deliverable shows how WP5 and WP4 infrastructures have been integrated to provide more insightful information to Policy Makers by: (i) complementing the data extracted from social media conversations with specific information from open data sources and (ii) by using information extracted from open data sources to enhance sentiment analysis within the context of a Lexicon Adaptation task.

List of figures

List of tables

List of abbreviations

<Abbreviation>	<Explanation>
PM	Policy Maker
TF	Term Frequency
OD	Open Data

1 SentiCircles: Software Extensions

In this section we present a brief overview of SentiCircles, our developed lexicon-based model for sentiment analysis over microblogging data, and the different architectural modifications that we have implemented over it to accommodate visualisation and navigation requirements highlighted by the use-case partners of the project. Details about the research and implementation behind SentiCircles are specified in D5.1 and D5.2 respectively. For more details the reader is also referred to the following publications. [4][5][6][7]

SentiCircle aims to represent the sentiment orientation of words with respect to their contextual semantics. The main notion behind this is that the sentiment of a term is not static, as in traditional lexicon-based approaches, but rather depends on the context in which the term is used, i.e., it depends on its contextual semantics. For example, most existing sentiment analysis methods fail to detect the sentiment of the tweet "`#Syria. Execution continues with smile! :( #ISIS`", since they consider the existence of the word "`smile`" positive, even though it is used within the context of the negative word "`Execution`".

Thus, in order to understand the semantics and the sentiment of a target word like "ISIS"(Islamic State in Iraq and Syria), our method relies on the words that co-occur with the target word in a given collection of tweets. These co-occurrences are mathematically represented as a 2D geometric circle; where the target word ("ISIS") is at the centre of the circle and each point in the circle represents a context word that co-occurs with "ISIS" in the tweet collection (see Figure 1~~Figure 1~~).

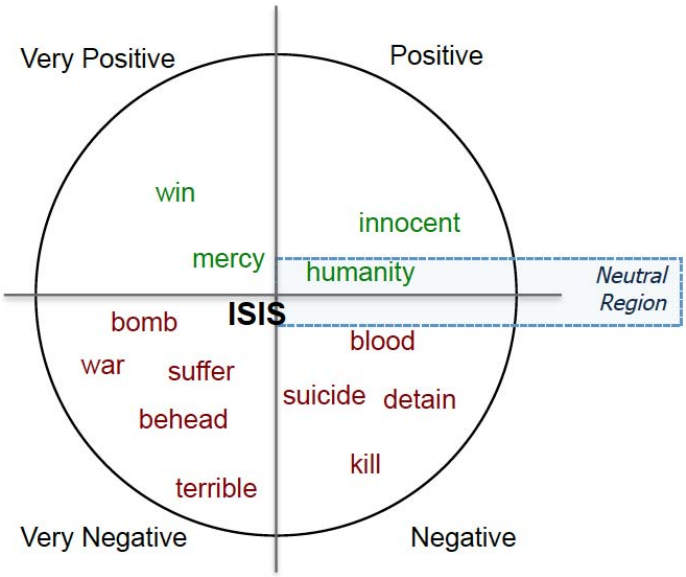


Figure 1: SentiCircle for the word "ISIS". Terms positioned in the upper half of the circle have positive sentiment while terms in lower half have negative sentiment

Following our 2-dimensional representation, see [5] for specific details on how this model is generated, the two upper quadrants of the SentiCircle contain words a positive sentiment, with upper left quadrant representing stronger positive sentiment. Similarly, terms in the two lower quadrants have negative sentiment values. The "*Neutral Region*" is located very close to X-axis in the "*Positive*" and the "*Negative*" quadrants only. Terms lie in this region have very weak sentiment.

The SentiCircles representation is used for detecting the positive and negative sentiment expressed on social media. SentiCircles can be used in the following two sentiment analysis tasks:

- **Entity-level Sentiment Analysis**: which aim to detect the sentiment of a given named entity (e.g., "Obama", "David Cameron", "ISIS")
- **Post-level Sentiment Analysis**: which aim to detect the overall sentiment of a given post (e.g., "*#Syria #ISIS. Execution continues with smile! :(*")

In the last period several additional functionalities have been added to SentiCircles. These extensions include:

1. The creation of an entity identification web-based platform, so that entities can be automatically identified from the collected social media conversations.
2. The provision of filtering mechanisms, design to allow the PM selecting specific terms while navigating the social media conversations.
3. The extraction of corpus statistics associated to the results of the sentiment analysis. These statistics provide PMs with provenance information about the obtained results.
4. The generation of the corpus-level SentiCircle, which provides a high level overview of the sentiment emerging from the topic under analysis (i.e., the topic of interest selected by the PM).

Specific details of these extensions and their implementation are reported in the following sections.

1.1 Entity Identification

Performing entity extraction from tweet data allows us to identify the different entities (Persons, Locations, Organisations) mentioned within the tweets, and how sentiment is expressed towards them. To extract these entities our developed tools make use of AchemyAPI.[1] We selected AlchemyAPI due to high performance in comparison with other entity extraction tools [2].

To use this tool in the context of our research we have developed a web-based platform on top of AlchemyAPI for the extraction and visualisation of entities. As we can see in Figure 2~~Figure 2~~ the input for our platform is a collection of tweet messages. The output is a list of entities and their associated concepts extracted from the tweet collection. Information of these entities is then added to the SentiCircle generation process.

Note that, as explained in D5.2, a SentiCircle is generated for every word in the corpus. Integrating entity information in the SentiCircle generation process allows us to determine which words represent entities. This is relevant for visualisation and integration purposes. In terms of visualisation, knowing which entities emerge from the data helps the PM to filter conversations around specific Persons, Organisations, etc. Moreover, additional information from LOD sources can be extracted for those entities, providing further insights to the PM. Specific details about the use of entities is provided in sections 1.2 and 3.

[1] http://www.alchemyapi.com/

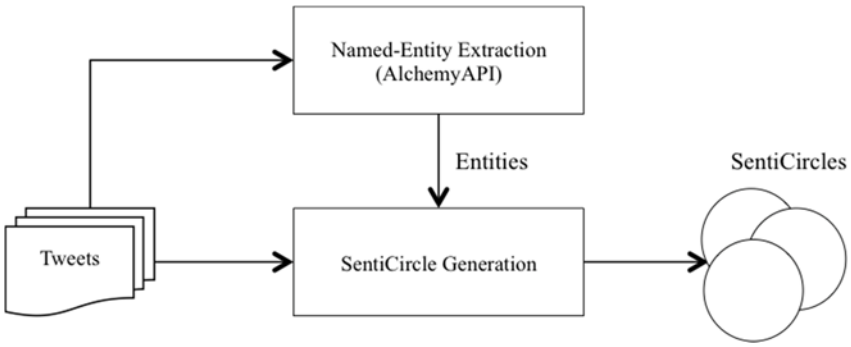


Figure 2: Integration of the entity identification tool during the Generation of SentiCircles

Figure 3~~Figure 3~~ and Figure 4~~Figure 4~~ show screenshots of the web-based platform. Figure 3~~Figure 3~~ shows processing information, indicating the number of records (tweets) from which extraction has been completed. Figure 4~~Figure 4~~ shows the results of the entity extraction task including: the identified entities, the tweets in which the entities have been identified, and a confidence score that provides an indication of the accuracy of the entity identification process.

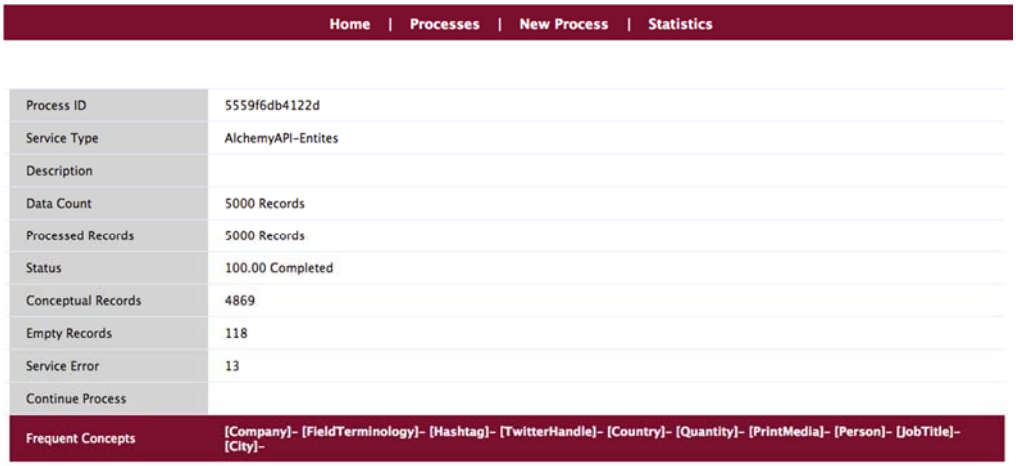


Figure 3: Web platform developed on top of AlchemyAPI

Entity	Relevance	Tweet
Apple	0.33	Apple is developing an electric car, report says http://t.co/62Fpl2DFTv
Tesla	0.736387	Is Apple going to challenge Tesla with an electric car? http://t.co/dlOCe6wJqY
Apple	0.60712	Is Apple going to challenge Tesla with an electric car? http://t.co/dlOCe6wJqY
Apple	0.33	Apple working on self-driving electric car: Report http://t.co/SRmT61l0Sd
Apple	0.33	WSJ: Tim Cook approved Apple electric car project a year ago, hundreds of employees working on it: Following a report today... #AAPL_Company
Apple	0.908333	Compelling reasons why @Apple are not making an electric... http://t.co/tNn84uyoEU
Apple	0.33	Apple is working on an electric car, Wall Street Journal reports (http://t.co/A4kgMgM9hb) http://t.co/18mn73vshx
YouTube	0.33	I liked a @YouTube video http://t.co/ymYNKksp5a MacBook Pro Upgrades and Apple's Secret Electric Car
Apple	0.938758	Apple sued for poaching engineers with deep expertise in electric car systems. Sounds like sour grapes but Apple????? http://t.co/x0d2byGdQl
Apple	0.924808	Hitting The Brakes On Apple's Electric Car: The rumor mill has shifted into overdrive about Apple developing a... http://t.co/AW7lU2P3w3

Figure 4: Entity extraction output

1.2 Filtering

The number of context terms that occur in the SentiCircle of a given target term (e.g., ISIS) depends on the size of the studied corpus. SentiCircles extracted from a large corpus usually contain a large number context terms. This makes the SentiCircle difficult to visualize.

To facilitate the navigation of SentiCircles, and to provide a more insightful visualisation to PMs, we have incorporated multiple filtering mechanisms to the SentiCircle sentiment analysis platform. Specifically, given a term (either a contextual or a target term within the SentiCircle), the filtering mechanisms are able to indicate:

- If the term is an entity: This allows the PM to concentrate on specific persons, organisations, etc.
- If the term is a stopword: This allows the PM to discard terms that may be irrelevant for their search.
- The Part of Speech (POS) of the term, i.e., if the term is a Noun, Adjective, preposition, etc. This allows PM to concentrate on specific objects (nouns) or on qualifiers of those objects (adjectives, adverbs). For example, if they want to know how citizens qualify electric cars ("expensive", "clean", etc.)
- If the term is a username, e.g., @obama. This allows the PMs to concentrate on the specific users that are mentioned within the tweets that they aim to analyse.
- If the term is a hashtag. Hastags, e.g., #greenenergy, are topics manually specified by the authors in their tweets. Focusing on the hashtags that emerge around a specific topic of interest (e.g., electric cars) can help PMs to identify what are the relevant subtopics.

To provide this set of filtering mechanisms we implemented several functionalities, as described in the table below. These functionalities are applied on the raw tweets in a given Twitter collection prior to generating the SentiCircle. The filtering mechanisms enable the end user to choose a specific category to visualize, providing a more meaningful navigation of the information.

Table 1: filtering functionalities

Functionality	*Description*	*Input*	*Output*
ParseTweets()	This function applies tokenization and part-of-speech tagging (i.e., assigning a word with its part-of-speech tag) on a collection of tweets messages. It uses TweetNLP, a POS-tagging library specifically built to work on Twitter data.	A collection of tweet messages	List of tagged tweets. Each tweet is broken down into a set of individual words. Each word is marked with its part-of-speech tag (e.g., "happy" is "adjective", @obama is "username", and "#healthcare" is "hashtag"

detectStopwords()	This function detects the stopwords in a given tweet message.	A collection of tokenized and *tagged* tweet messages. A pre-complied list of stopwords, the Van stoplist, which consists of 250 stopwords (e.g., “is”, “otherwise”, “whatever”)	A collection of tweets whose words are marked with a stopword flag .

1.3 Corpus Statistics

Several relevant statistics can be extracted from the SentiCircle of a given term/entity. These statistics include:

1. The total number of tweets that the target term occurs within
2. The total number hashtags and usernames within the SentiCircle

The aim of extracting these statistics is to provide PMs with information about the provenance of the results presented to them. The table below contains the functionalities that have been implemented to extract the above statistics for each SentiCircle.

Table 2: Functionalities to extract general statistics

Functionality	*Description*	*Input*	*Output*
extractTweetsCount()	Given a SentiCircle, this function extracts the number of tweets in which the SentiCircle’s target term occurs in.	The target term of a given SentiCircle (e.g., “Obama”)	The number of tweets that the target term (e.g., “Obama”) occurs in.
extractUsernamesCount()	This function extracts the number of usernames (e.g., “@obama”) in a given SentiCircle. Remember that the each term in the SentiCircle is marked with its Part-of-Speech and social tags due to applying the filtering	A SentiCircle	The number of usernames found in the SentiCircle.

	mechanisms at the time of building the SentiCircle (See Section 1.2).		
extractHashtagsCount()	This function extracts the number of hashtags (e.g., "#healthcare") in a given SentiCircle.	A SentiCircle	The number of hashtags found in the SentiCircle.

1.4 Corpus SentiCircle

As described earlier, each word in an analysed corpus is represented by means of its SentiCircle, which denotes the word's contextual semantic and sentiment orientations.

The corpus SentiCircle aggregates information of all the extracted SentiCircles for an analysed corpus. The goal is to obtain the collective sentiment about the topic discussed in the corpus. The corpus SentiCircle is computed as follows:

- Calculate the geometric median for each SentiCircle, where each SentiCircle is built around a different word (term) of the corpus.
- Construct the SentiCircle of the corpus by placing all terms' geometric medians within it.
- Calculate the overall sentiment of the corpus by extracting the geometric median of the Corpus' SentiCircle.

Functionality	*Description*	*Input*	*Output*
generateCorpusCircle()	This function generate the corpus circle for a given twitter collection (Twitter corpus)	A list of the geometric medians of all the SentiCircles in the corpus. This list is automatically constructed at the time of generating the SentiCircle model for the given tweet corpus (See Section 1.5.1)	The corpus SentiCircle (The output format of this function is described in Section 1.5.1)

1.5 Main Functions and Usage

The SentiCircle sentiment analysis platform is implemented using Java 6 and provides multiple functionalities for the semantic representation and sentiment extraction of words. The new functionalities added to the SentiCircle platform, as described in the previous sections, required several changes in the usage, and the input and the output of the model. In

this section we report the changes made on the main functionalities of the SentiCircle sentiment analysis platform.[2]

```
public void generateModel(java.lang.String dataFile) throws
java.io.IOException
```

The above function generates the SentiCircle model for a given tweet corpus. The generated model is stored under the "data/model" directory. Note that, for every word in the tweet collection, a SentiCircle is generated.

Changes: This function has been re-implemented to include:

1. Tagging the context terms in the generated SentiCircles with their POS-tags and social tags required for the filtering mechanism (Section 1.2)
2. Generating statistics about the terms' SentiCircles (Section 1.3)
3. Generating the Corpus SentiCircle (Section 1.4)

As such, the new pipeline for generating the SentiCircle model for a given Twitter corpus is depicted in the figure below.

[2] For detailed description of the implementation and usage of the SentiCircle code, please refer to D5.2, Section 1.3

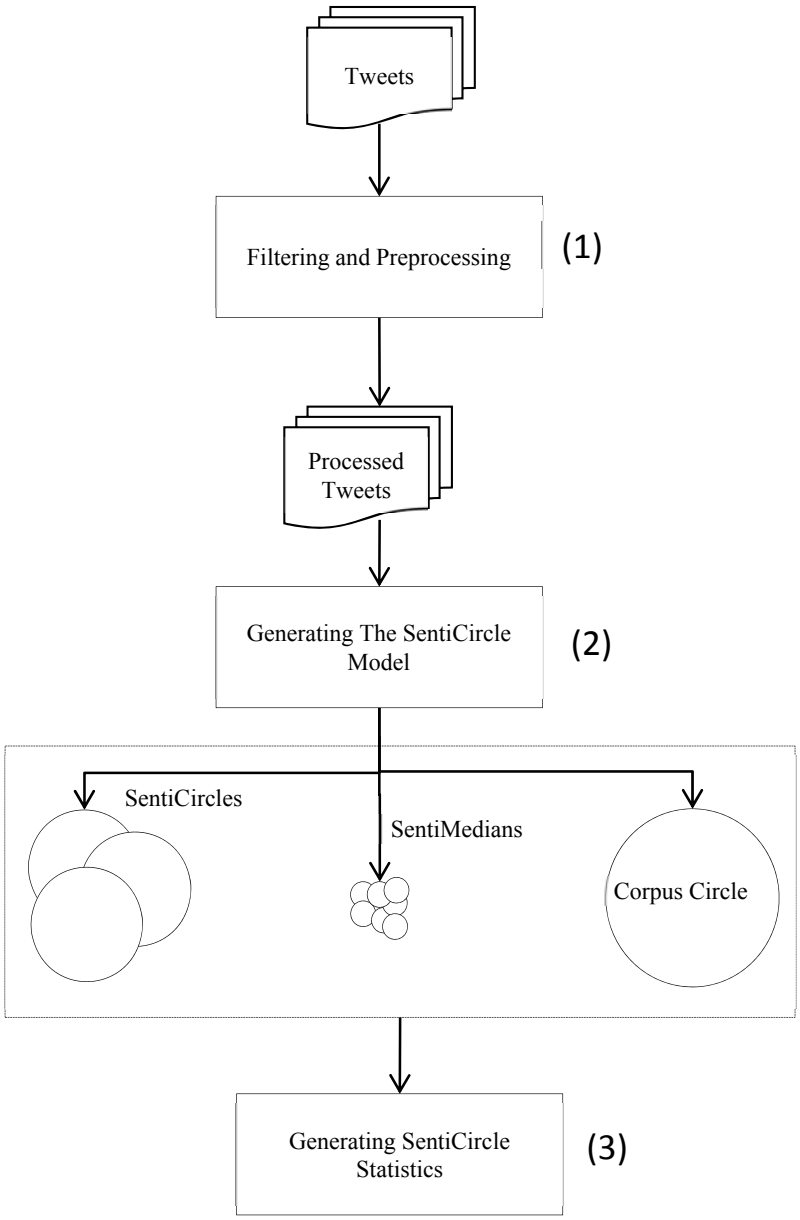


Figure 5: Modifications to the SentiCircle sentiment analysis platform

Parameters: `dataFile -` the file path of the tweet collection (see Input Format)

Returns: Directory of the generated model in JSON format (see Output Format)

Throws: `java.io.IOException` - input file not found

Input Format `dataFile:` a semicolon-separated CSV file. Each line in the file represents a tweet message, as follows: `"tweet_id";"tweet_message"`, where *tweet_id* is a number representing the id of a tweet in the file, and *tweet_message* is the text of the tweet.

Output Format: JSON format as follows:

```
{
```

```
    "modelName": "the name of the generated SentiCircle Model",
    "modelDir": "the relative path of the model's directory"
}
java -jar senticircle.jar -inf -model sts_gold_binary.csv
```

Output Files

The directory under which the model is generated contains three main files as follows:

1. SentiCircle (JSON): This file contains a list of JSON objects representing the SentiCircles. The format of each JSON object is as follows:

```
{
   "label": "the name label of the target term"
   "median": { // The SentiMeidan of the SentiCircle
      "x": ”The X-coordinate of the SentiMedian Point”
      "y": “The Y-coordinate of the SentiMedian Point”
      "r": “The radius from the origin of the SentiMedian point”
      "theta": “The angle between origin and the SentiMedian point”
      "region": “The sentiment region to which the SentiMedian belong”
      }
   },
   "numTweets": “The total number of tweets for the SentiCircle”
   "tweetIds": “Ids of the tweets the target term occurs in”
   "points": [ “List of Context Points within the SentiCircle”
         {
         "label": "The label of the context point"
         "x": “The X-coordinate of the context point within the SentiCircle”
         "y": “The Y-coordinate of the context point within the SentiCircle”
         "r": “The radius from the origin of the context point”
         "theta": “The angle between origin and the SentiMedian point”
         "tag": "The POS-tag or social tag of the context point"
         "region": “The sentiment region to which the context point belong”
         "correlation": { // the sentiment correlation between the target
point and the context point
            "sentiment": “the sentiment label”,
            "correlation": “the value of the sentiment correlation”,
            "relative_correlation": “the relative correlation value”
         }
      },
         … rest of all context points
   ]
}
```

As can be noted, the field “tag” in JSON object represents the part-of-speech tag of the context term in the SentiCircle. The figure below shows the list of values that this filed could possibly have:

Tag	Description	Examples
Nominal, Nominal + Verbal		
N	common noun (NN, NNS)	books someone
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee
S	nominal + possessive	books' someone's
^	proper noun (NNP, NNPS)	lebron usa iPad
Z	proper noun + possessive	America's
L	nominal + verbal	he's book'll iono (= *I don't know*)
M	proper noun + verbal	Mark'll
Other open-class words		
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't is eats
A	adjective (J*)	good fav lil
R	adverb (R*, WRB)	2 (i.e., *too*)
!	interjection (UH)	lol haha FTW yea right
Other closed-class words		
D	determiner (WDT, DT, WP$, PRP$)	the teh its it's
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., *to*) 4 (i.e., *for*)
&	coordinating conjunction (CC)	and n & + BUT
T	verb particle (RP)	out off Up UP
X	existential *there*, predeterminers (EX, PDT)	both
Y	**X** + verbal	there's all's
Twitter/online-specific		
#	hashtag (indicates topic/category for tweet)	#acl
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello
U	URL or email address	http://bit.ly/xyz
E	emoticon	:-) :b (: <3 o_O
Miscellaneous		
$	numeral (CD)	2010 four 9:30
,	punctuation (#, $, '', (,), ,, ., :, ``)	!!! .... ?!?
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)	ily (*I love you*) wby (*what about you*) 's ♫ --> awesome...I'm

Figure 6: Part of Speech Annotations

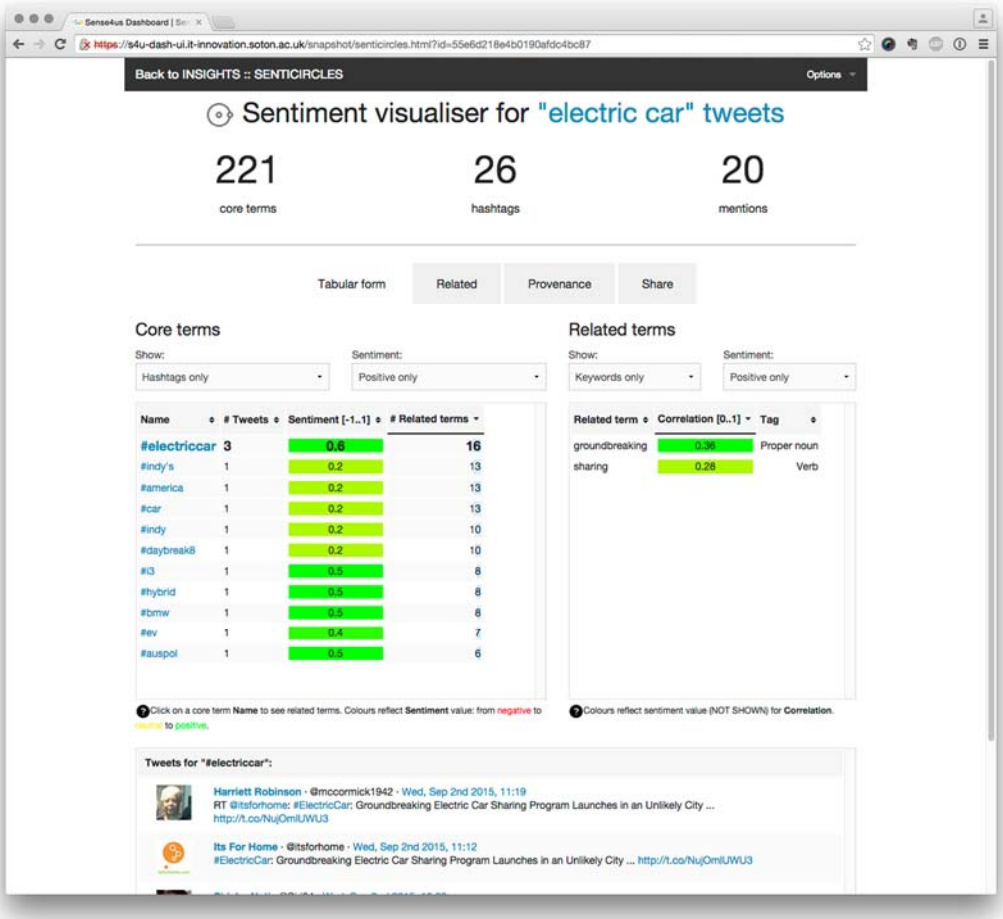
2. Corpus SentiCircle (JSON): This file contains the SentiCircle of the analysed Twitter corpus. The format of this file is similar the above SentiCircle output format.
3. SentiMedian (CSV) This file contains list of the SentiMedians of each target term in the Twitter corpus with the following format:

```
Term              X-coordinate      Y-coordinate
"electric_car";   "0.00439";        "-0.00111"
"sound";          "0.00608";        "-0.01871"
"green";          "0.11324";        "0.15587"
```

2 Platform Integration

The following figure reflects how the SentiCircles sentiment analysis platform has been integrated into the overall Sense4us platform. Figure 7~~Figure 7~~ displays the results of the sentiment analysis performed over a Twitter search for “electric cars”. The general statistics are displayed at the top of the page. The analysis provided 221 core terms (i.e., 221 SentiCircles) to describe the analysed corpus. This corpus contains a total of 26 related hashtags and 20 mentions to relevant users.

SentiCircles are visualised in a table shape. The left hand side table contains the core terms (each core term is a SentiCircle, where the core term is at the centre of the circle). The list of core terms can be filtered by displaying only a subset of core terms based on the filtering mechanisms described in section 1.2. It can also be filtered by the sentiment of those core terms (i.e., display the most positive, the most negative or the neutral ones). For each core term the table visualises the number of tweets in which the core term appears, its sentiment within the corpus, and a number of related terms (i.e., contextual terms that appear in the SentiCircle of the core term). Contextual terms can also be filtered by their part of speech and by their sentiment with respect to the core term. At the bottom of the page, the interface also displays examples of tweets containing the core term under the specific filtering conditions selected by the user. This helps the PM to acquire a better understanding of why certain core terms display a particular sentiment, or how two terms (e.g., “electric car” and “pollution”) are related in the social media conversations.



‘

Figure 7: Integration of SentiCircles into the Sense4us platform

SentiCircles provides a novel way to summarise and navigate social media conversations, focusing on the core terms emerging from the conversations, their sentiment and the context that surround those core terms. This novel navigation of social media conversations can provide users with relevant insights that, to the best of our knowledge, no other existing social media sentiment analysis tool is currently providing.

3 Open Data Querying Infrastructure

In this section, we briefly describe the functionalities of the Open Data Querying Infrastructure. The main goal of this infrastructure is to identify and retrieve relevant information, like major stakeholders or related concepts, between different entities within the Linked Open Data (LOD) cloud. A detailed description of this infrastructure will be given in Deliverable 4.3.2 but all functionality is currently accessible under the Sense4us git repository (https://gitlab1.it-innovation.soton.ac.uk/).

Linked Open Data (LOD)[3] is a growing initiative that publishes information not only in the form of pure textual or hypertext documents but as structured knowledge representation where links between different entities are not only syntactical given, like on websites, but have defined semantics, what a connection means. Entities in LOD, e.g. organisations, persons, or even concepts, are identified by unique resource identifiers (URIs). Knowledge about entities is formulated as a triple of three URIs. The first URI identifies the subject of the described knowledge, the last URI identifies the object, and the middle URI defines the type of relation, also called predicate. The benefit of LOD to many other knowledge representation approaches is that it is designed to be distributed over many different data sources that can be interlinked with each other to form a large, connected data graph. Our system uses this global and openly accessible collection of information to support policy makers in identifying stakeholders and relevant concepts. This approach widens the understanding of a policy topic and helps policy makers by gaining insights.

3.1.1 Concept

The concept behind the developed Open Data Querying Infrastructure is, that within the LOD cloud, related information is connected through a path of triples. To find these paths, the following assumptions are given:

1. Related entities are typically closer to each other (short path length)
2. Related entities are typically connected via multiple paths (high connectivity)
3. Related entities are typically connected by paths that contain important predicates (high path relevance). The importance of a predicate in a path is given by the combination of how often it occurs in the data set and how often it is used in combination of its connected subject or object.

3.1.2 Architecture

Our architecture consists of two components, query execution and connectivity ranking. The first retrieves paths between two given entities on one or two given SPARQL endpoints. SPARQL Endpoints are interfaces that allow users to query specific datasets.

If only one endpoint is given, the system tries to find paths between both entities within the given endpoint.

If two endpoints are given, the system tries to find both entities in both endpoints and tries to find a connection between the entities. If both entities are in one endpoint, the previous mentioned procedure is executed; otherwise the system tries to identify a connection between the surrounding entities of both entities that reaches from one dataset to the other.

[3] http://linkeddata.org/

This approach allows executing the identification of connectivity structures (collection of paths that connect two entities with each other) between multiple entities in multiple datasets by applying both procedures multiple times. To speed up the process, a maximum path length can be specified to reduce the query execution time.

Given a connectivity structure, found within the query execution process, the connectivity ranking gives every path a value depending on the previously mentioned metrics: path length, connectivity value, and path relevance. The first two metrics can directly be derived by the given data. For the path relevance, the system has to query the endpoint with a set of lightweight queries. After that, the system returns the results.

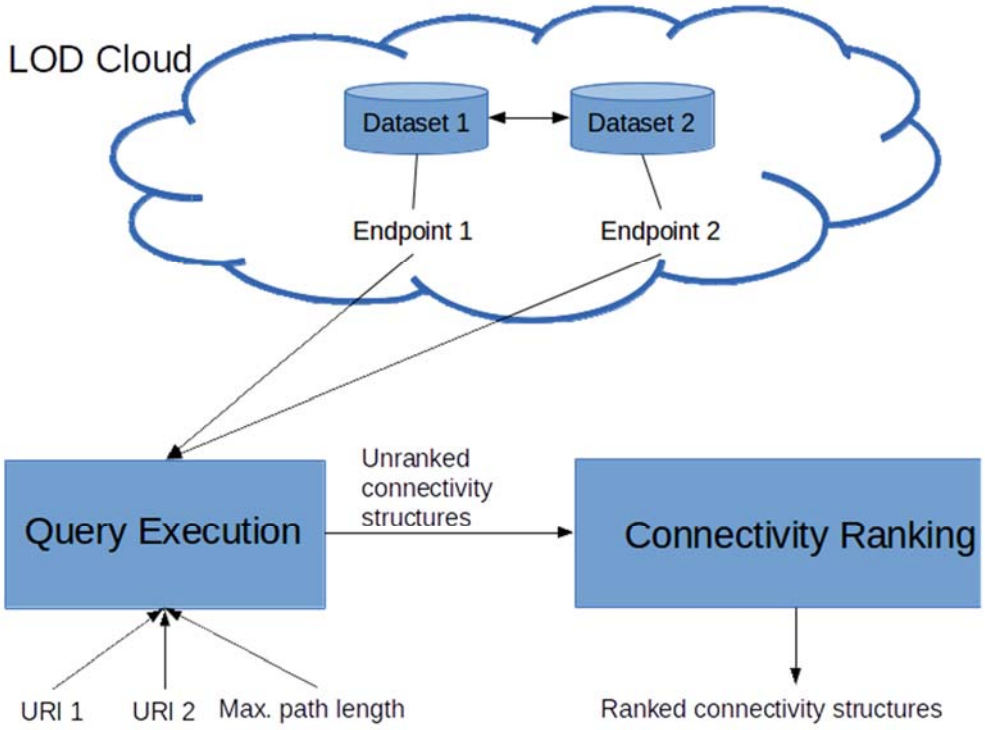


Figure 8: Architecture of Connectivity Structure Discovery Framework

4 Complementing Social Media Discussions With Information From Open Data Sources

In this section we describe how the SentiCircles sentiment analysis platform has been integrated with the Open Data Querying Infrastructure to enrich the information extracted from the social media discussions and to provide further insights to the PM. Note that this integration has been done at the backend and further development is needed to bring the results of this integration to the frontend, i.e., to the interface of the Sense4us platform. We illustrate this integration with an example around the policy use case on electric cars presented in D2.1.

4.1 Integration Infrastructure

Figure 9~~Figure 9~~ represents the information flow in which information extracted from social media conversations is enriched with information extracted from LOD. The goal of this integration is to provide the PM with further information that can help her to support or refute arguments or to obtain additional knowledge around specific issues of interest.

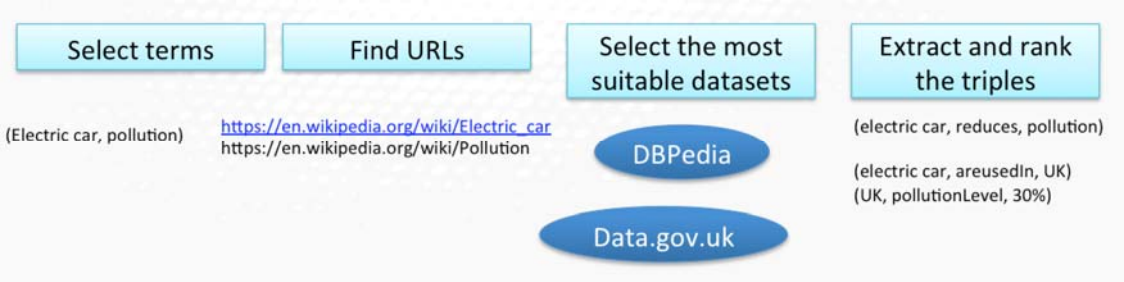


Figure 9: Integration Flow

In this particular example we show how to use SentiCircles in combination with the Open Data Querying Infrastructure to request additional information about the relation between electric cars and pollution. From the results obtained by SentiCircles the PM can observe which terms are assigned more positive and negative sentiment, which terms are related to each other, influence each other, and how they appear together in the tweets (see Section 2 for details).

Following our example, the PM observes that the term pollution is related with electric cars with a very negative sentiment. To explore this relation she observes some of the tweets in which those terms appear together, for example: "*Electric cars increase pollution #no2electriccars*". In this particular case, the PM may want to find out more information about the relation between electric cars and pollution, and if there are any facts in LOD supporting this claim.

The selected pair of terms is then provided as an input to the Open Data Querying infrastructure. The first step that this infrastructure does (if terms haven't already been categorised as entities by the SentiCircle sentiment analysis platform) is to try to map those terms in LOD and obtain the URLs that identify them.

Once the URLs for the terms have been identified (e.g., Electric car -> https://en.wikipedia.org/wiki/Electric_car), the Open Data Querying infrastructure obtains the connectivity structures linking those two terms. Connectivity structures are then ranked based on the previously mentioned criteria: sort path, high connectivity, and high path relevance.

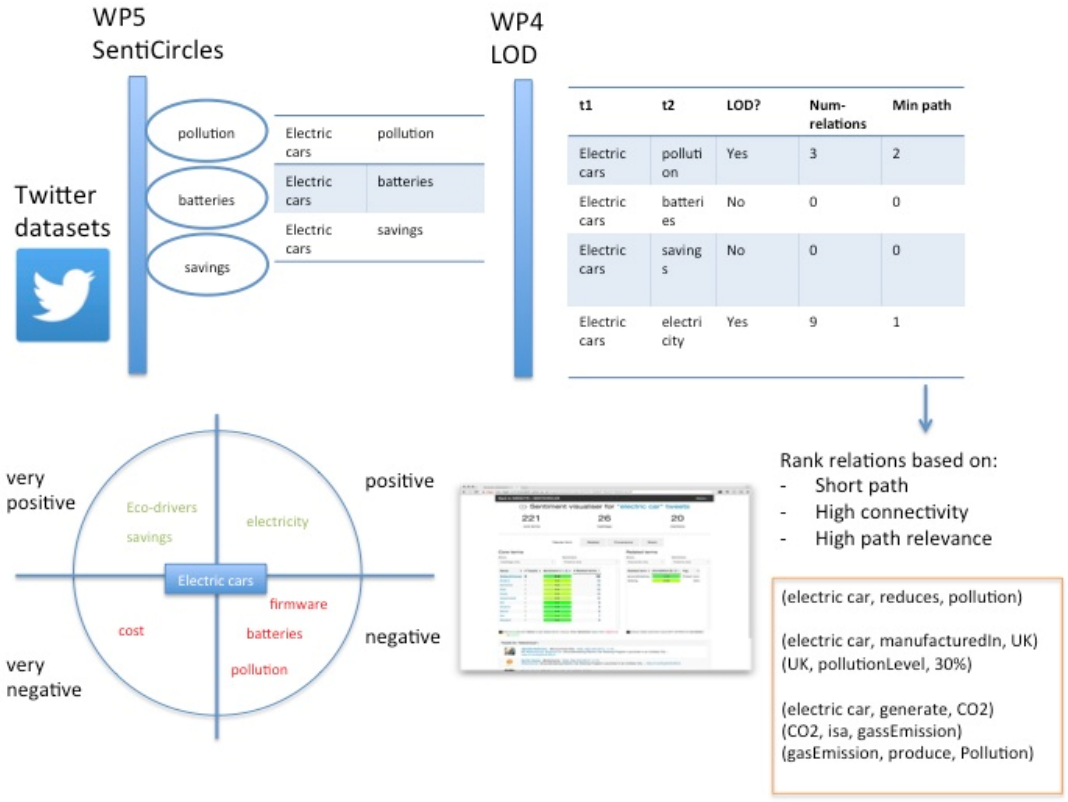


Figure 10: Integration Infrastructure

Figure 10~~Figure 10~~ shows in more detail how the integration infrastructure works. When collecting social media information around a certain topic (in this case "electric cars") the SentiCircles sentiment analysis platform will provide the terms that emerge from the social media conversations and the sentiment around those terms (see Section 2).

Selected pairs of terms are then sent to the LOD query infrastructure to obtain additional information about them. In our example these pairs include (electric cars and pollution) (electric cars and batteries) (electric cars and savings). Following the flow presented in Figure 9~~Figure 9~~, connectivity structures are searched for the selected pairs of terms and ranked based on the path length, the level of connectivity and the path relevance.

4.2 Integration Experiment

To test our integration infrastructure we collected 74,127 Twitter posts around "*electric cars*" between 03/02/2015 and 20/02/2015. These posts were input to the SentiCircles sentiment analysis platform, which extracted 3,011 core terms, with their corresponding SentiCircles. These core terms were filtered by using the entity-filtering mechanisms reported in section 1.2. After this filtering 5,000 pairs of terms were produced and processed by WP4's querying infrastructure.

A manual inspection of these triples was performed by two different researchers. Among the retrieved triples multiple ones were found useful, especially the ones directing to specific documentation of PDFs. For example, the following connectivity structure indicates a link to

(http://theicct.org/sites/default/files/publications/ICCT_EV-fiscal-incentives_20140506.pdf). This document provides a global comparison of fiscal incentive policy for electric vehicles.

<http://dbpedia.org/resource/Electric_car>
<http://www.w3.org/2000/01/rdf-schema#seeAlso>
<http://dbpedia.org/resource/United_Kingdom> ,
<http://dbpedia.org/resource/France> ,
<http://dbpedia.org/resource/Electric_car> ,
<http://dbpedia.org/resource/Plug-in_electric_vehicle> ,
<http://dbpedia.org/resource/Electric_car_use_by_country> ,
<http://dbpedia.org/resource/Government_incentives_for_plug-in_electric_vehicles> ;
<http://dbpedia.org/ontology/wikiPageExternalLink>
<*http://theicct.org/sites/default/files/publications/ICCT_EV-fiscal-incentives_20140506.pdf*> ;
<http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Sustainable_technologies> .

During the performed manual inspection we also observed that, while connectivity structures can provide very useful information, they are very difficult to understand by non-technical audiences. We are working now, in collaboration with WP4 and WP3 on different ways in which these connectivity structures can be visualised and on ways in which the different ranking strategies can be evaluated.

5 Exploring the use of Semantic Information for Sentiment-Lexicon Adaptation

As part of the integration with WP4 we have also studied how the extracted semantic information can be used to enhance sentiment analysis, in particular within a Sentiment-Lexicon Adaptation task. General-purpose sentiment lexicons are simple and effective for calculating the overall sentiment of texts using a general collection of words, with predetermined sentiment orientation and strength. However, words' sentiment often vary with the contexts in which they appear, and new words might be encountered that are not covered by the lexicon. Although much research has been done on creating domain-specific sentiment lexicons, very little attention has been giving to the problem of lexicon adaptation in social media, and to the use of semantic information as a resource to perform such adaptations. Within Sense4us we have developed a lexicon adaptation approach, oriented towards microblogging data, which uses contextual (SentiCircles) as well as semantic (LOD) information to update the words' weighted sentiment orientations and to add new words to the lexicon. We have evaluated our approach on three different Twitter datasets, and show that enriching the lexicon with contextual and semantic information improves sentiment computation.

5.1 Semantic Enrichment for Context-based Lexicon Adaptation

In [6] we presented our approach for context-based Lexicon Adaptation based on SentiCircles. However, relying on the context only for detecting terms' sentiment might be insufficient. This is because the sentiment of a term may be conveyed via its conceptual semantics rather than by its context [8]. For example, the context of the word "*Ebola* " in "*Ebola continues spreading in Africa!*" does not indicate a clear sentiment for the word. However, "*Ebola*" is associated with the semantic type (concept) "*Virus/Disease*", which suggests that the sentiment of "Ebola " is likely to be negative.

To address this issue we propose enriching our lexicon adaptation model based on SentiCircles [6] with semantic information. For this purpose we perform the following steps:

- Conceptual semantic extraction: This step extracts the entities that appear in a tweet collection (e.g., "Obama ", "Illinois ", "NBC ") along with their associated semantic types ("Person ", "City ", "Company ") and their semantic subtypes (e.g., "Politician ", "US County ", "TV Network "). To extract these entities and its associated semantic types we use the developed web entity identification platform (see Section 1.1) and the open-data querying infrastructure developed by. (See Open Data Querying Infrastructure)
- Conceptual semantic enrichment: This step incorporates the conceptual semantics extracted from the previous step as additional terms within the generation of SentiCircles. The approach for building the SentiCircle model does not change [5]. Note that we currently rely only on the entities' semantic subtypes for the semantic enrichment phase, excluding the semantic types. Unlike semantic types, semantic subtypes capture more fine-grained knowledge about the entity (e.g., "Obama " > "Politician ").

5.2 Experiment Setup

In this section we present the experimental set up used to assess our proposed lexicon adaptation model. This setup requires the selection of: (i) the sentiment lexicon to be adapted, (ii) the context (Twitter datasets) for which the lexicon will be adapted, (iii) the

different configurations for adapting the lexicon, and (iv) the semantic information used for the semantic adaptation model. All these elements will be explained in the following subsections. We evaluate the effectiveness of our method by using the adapted lexicons to perform tweet-level sentiment detection, i.e., detect the overall sentiment polarity (positive, negative) of tweets messages. Our baseline for comparison is the original sentiment lexicon without any adaptation.

5.2.1 Sentiment Lexicon

For the evaluation we choose to adapt the state-of-the-art sentiment lexicon for microblogs; Thelwall-Lexicon [9][10]. Thelwall-Lexicon is a general purpose sentiment lexicon specifically designed to function on microblogging data. It consists of 2546 terms coupled with values between -5 (very negative) and +5 (very positive), defining their sentiment orientation and strength. Terms in the lexicon are grouped into three subsets of 1919 negative terms ($prior_t$ 2 [-2,-5]), 398 positive terms ($prior_t$ 2 [2,5]) and 229 neutral terms ($prior_t$ 2{ -1,1}). Thelwall-lexicon was selected for this evaluation because, to the best of our knowledge, it is currently the best performing lexicon for computing sentiment in microblogging data.

5.2.2 Evaluation Datasets

To assess the performance of or lexicon adaptation method we require the use of datasets annotated with sentiment labels. For this work we selected three evaluation datasets often used in the literature of sentiment analysis [3] (SemEval, WAB and GASP). These datasets differ in their sizes and topical focus. Numbers of positive and negative tweets within these datasets are summarised in Table 3.

Table 3: Twitter datasets used for evaluation. Details on how these datasets were constructed and annotated are provided in [3]

Dataset	Tweets	Negative	Positive	Unigrams
SemEval Dataset	7520	2178	5342	22340
WAB Dataset	5482	2577	2905	10917
GASP Dataset	6032	4999	1033	12868

5.2.3 Conceptual Semantics

We use AlchemyAPI to extract the conceptual semantics of named entities from the three evaluation datasets (Section 3.2). Table 4 lists the total number of entities extracted and the number of semantic types and subtypes mapped against them for each dataset.

Table 5 shows the top 10 frequent semantic subtypes under each dataset. As mentioned in Section 3.2, we only use the entities' semantic subtypes for our semantic enrichment, mainly due to their stronger representation and distinguishing power than general higher level types (e.g., "Person").

Table 4: Unique Entity/Types/Subtypes for SemEval, WAB and GASP using AlchemyAPI

	SemEval	WAB	GASP
Number of Entities	2824	685	750
Number of Semantic Types (Concepts)	31	25	23

Number of Semantic Subtypes	230	93	109

Table 5: Top 10 frequent semantic subtypes of entities extracted from the three datasets

SemEval		WAB		GASP	
Subtype	Frequency	Subtype	Frequency	Subtype	Frequency
TVActor	505	AdministrativeDivision	93	AwardWinner	350
AwardWinner	351	GovernmentalJurisdiction	91	Politician	328
MusicalArtist	344	Location	66	Celebrity	321
Filmactor	324	Placewithneighborhoods	49	Location	104
Athlete	316	PoliticalDistrict	45	AdministrativeDivision	103
Location	263	Sportsteam	12	GovernmentalJurisdiction	102
GovernmentalJurisdiction	263	FieldofStudy	11	PlaceWithNeighborhoods	15
Footballplayer	238	Invention	10	Musicalartist	14
Celebrity	230	MusicalArtist	10	AutomobileCompany	13
AwardNominee	225	VentureFundedCompany	9	BroadcastArtist	11

5.2.4 Configuration of the Lexicon Adaptation Model

We tested our context-based adaptation model and the semantic adaptation model under three different configurations:

- *Semantic Lexicon Update (SLU)*: The lexicon is adapted only by updating the prior sentiment of existing terms. For example, the prior sentiment of the pre-existing word "Great" in Thelwall-Lexicon (i.e., $prior_{great}$=+3) will be updated based on the semantically enriched SentiCircle (for example, the value of $prior_{great}$ is updated to +1). The rules used to update prior sentiment values based on SentiCircles are described in [6]
- *Semantic Lexicon Expand (SLE)*: The lexicon is adapted only by adding new opinionated terms. Here new words, such as "Tragedy" and "Ebola", along with their computed sentiment based on the semantically enriched SentiCircle will be added to the lexicon.
- *Semantic Lexicon Update and Expand (SLUE)*: The lexicon is adapted by adding new opinionated terms ("Tragedy" and "Ebola") and by updating the prior sentiment of existing terms ("Great").

5.3 Results of Lexicon Adaptation with Semantic Enrichment

In this section we present the results of assessing the effectiveness of our semantically enriched context-based model for lexicon adaption. Table 6 shows the average results across the three datasets considering the three different settings of lexicon adaptation: semantic update, semantic expand, and semantic update and expand.

Table 6: Average results across the three datasets of Thelwall-Lexicon adapted by the semantic model. Italic=significance at 0.05, None-Italic=significance < 0.001

Model	Lexicon	Accuracy	Negative Sentiment			Positive Sentiment			Average		
			P	R	F1	P	R	F1	P	R	F1
	Original	73.49	71.90	79.09	73.88	66.30	64.23	63.77	69.10	71.66	68.83
Semantic Model	SLU	**76.47**	75.54	76.31	75.29	65.93	69.20	66.87	**70.74**	**72.75**	**71.08**
	SLE	73.78	72.40	77.02	73.58	65.44	65.66	*64.31*	68.92	71.34	68.94
	SLUE	76.42	75.47	76.31	75.24	65.90	*69.10*	66.81	*70.69*	72.71	71.03

As we can see in the above table, our results show that the adapted sentiment lexicons outperformed the original lexicon in accuracy by nearly 3% and F1 measure by 2.25% when enriching the lexicon adaptation model with semantics.

6 Conclusions

D5.3 is a prototype deliverable, which summarises the code released as part of WP5 and as part of WP4-WP5 integration. All the code and data described in this deliverable is available in the Sense4us code repository (https://gitlab1.it-innovation.soton.ac.uk/). The current deliverable aims therefore to be a manual for other researchers and developers who may want to make use of or extend our research.

The deliverable reports the extensions performed to the SentiCircles sentiment analysis infrastructure previously described in D5.1 and D5.2. Specifically, this deliverable reports the developed entity extraction mechanisms, the data filtering mechanisms and the statistics extraction mechanisms to provide the PM with further information about the sentiment and opinions emerging from social media conversations.

This deliverable also summarises how all these functionalities have been integrated into the general Sense4us demo in collaboration with WP3 and how they provide PMs with novel ways to navigate the sea of information available as social media conversations.

To enhance the sentiment analysis provided by the SentiCircles sentiment analysis platform we have complemented the information extracted by it with the information extracted by the Open Data Querying Infrastructure developed by WP4. We report here how the output of WP5 can be enhanced by finding specific connectivity structures among pairs of terms of particular interest to the PM. Connectivity structures are ranked based on three criteria: path length, high connectivity and path relevance.

An experiment to show how connectivity structures extracted from OD can enrich the analysis of social media conversations have been conducted using 74K tweets around the “electric cars” policy example presented in D2.1. Our analysis shows that ranking strategies can be enhanced by considering the existence of specific documents (such as PDFs) within the triples retrieved as part of the connectivity structures. Further work in this direction includes the exploration and combination of ranking strategies to provide further filtering to the retrieved connectivity structures and the implementation of visualisation mechanisms to present this information in an understandable way for non-technical audiences.

In addition, we have also explored how the information obtained from Open Data can help to enhance sentiment analysis, in particular within a Sentiment-Lexicon Adaptation task. In [6] we proposed a method to adapt sentiment lexicons based on contextual information, where the domain or context of adaptation is defined by a collection of posts. In this deliverable we have presented a semantic enrichment of this method where conceptual semantics are used to better capture the context for which the lexicon is being adapted. Our results show that the adapted sentiment lexicons outperformed the original lexicon in accuracy by nearly 3% and F1 measure by 2.25% when enriching the lexicon adaptation model with semantics.

7 References

[1] Saif, H., He, Y., Alani, H.: Alleviating data sparsity for twitter sentiment analysis. In:Proc. Workshop on Making Sense of Microposts (#MSM2012) in WWW 2012. Lyon, France (2012)

[2] Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Proc. 11th Int. Semantic Web Conf. (ISWC). Boston, MA (2012)

[3] H. Saif, M. Fernandez, Y. He, and H. Alani. Evaluation datasets for twitter sentiment analysis a survey and a new dataset, the sts-gold. In Proceedings, 1st Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM) in conjunction with AI*IA Conference, Turin, Italy, 2013.

[4] H. Saif, M. Fernandez, Y. He, and H. Alani. 2014a. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In Proc. 9th Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland, 2014.

[5] H. Saif, M. Fernandez, Y. He, and H. Alani. 2014b. SentiCircles for Contextual and Conceptual Semantic Sentiment analysis of Twitter. Extended Semantic Web Conference (ESWC), Crete, 2014.

[6] H. Saif, M. Fernandez, Y. He, and H. Alani. 2014c. Adapting Sentiment Lexicons using Contextual Semantics for Twitter Sentiment Analysis. In Proceeding of the first semantic sentiment analysis workshop: conjunction with the eleventh Extended Semantic Web conference (ESWC). Crete, Greece.

[7] H. Saif, Y. He, M. Fernandez and H. Alani. 2014d Semantic Patterns for Sentiment Analysis of Twitter, The 13th International Semantic Web Conference (ISWC), Riva del Garda - Trentino Italy

[8] Cambria, E.: An introduction to concept-level sentiment analysis. In: Advances in Soft Computing and Its Applications, pp. 478–483. Springer (2013)

[9] Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. J. American Society for Information Science and Technology 63(1), 163–173 (2012)

[10] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. J. American Society for Info. Science and Technology 61(12) (2010)