FP7-ICT Strategic Targeted Research Project TrendMiner (No. 287863)

Large-scale, Cross-lingual Trend Mining and Summarisation of Real-time Media Streams

D5.5. Deployment of web services for new use cases

José L. Martínez Fernández (Editor, DAEDALUS), Paloma Martínez (UC3M), Adrián Luna (DAEDALUS), Julián Moreno (UC3M), Mateusz Kopeć (IPIPAN), Maciej Ogrodniczuk (IPIPAN), Márton Miháltz (RILMTA) and Thierry Declerck (DFKI)

Abstract

FP7-ICT Strategic Targeted Research Project TrendMiner (No. 287863)
Deliverable D5.5 Deployment of web services for new use cases (WP 5)

The last year of the TrendMiner project has been devoted to the inclusion of new languages and new domains. A prototype for social media monitoring in the health domain for Spanish has been developed. Extensions of the political use cases have been provided for the Hungarian and Polish languages and extensions of the financial use case have been provided for the Spanish language.
This deliverable describes the new services that have been developed and the extensions made to the original TrendMiner platform for supporting the new applications. The final status of the TrendMiner platform is described in details in D5.3.2 Real-time Stream Media Processing Platform and Cloud-based Deployment.

Keyword list: eHealth, finance, languages, monitoring, reputation, social media, real time

Nature: Prototype | Dissemination: PU
Contractual date of delivery: 31.10.2014 | Actual date of delivery: 31.10.2014
Reviewed By: ONTO, USFD
Web links: http://www.trendminer-project.eu/index.php/publications/85-public-deliverable

TrendMiner Consortium

This document is part of the TrendMiner research project (No. 287863), partially funded by the FP7-ICT Programme.

DFKI GmbH
Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrcken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

University of Southampton
Southampton SO17 1BJ
UK
Contact person: Mahensan Niranjan
E-mail: mn@ecs.soton.ac.uk

Internet Memory Research
45 ter rue de la Révolution
F-93100 Montreuil
France
Contact person: France Lafarges
E-mail: contact@internetmemory.org

Eurokleis S.R.L.
Via Giorgio Baglivi, 3
Roma RM
00161 Italia
Contact person: Francesco Bellini
E-mail: info@eurokleis.com

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1930
Fax: +44 114 222 1810
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

Ontotext AD
Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Sora Ogris and Hofinger GmbH
Bennogasse 8/2/16
1080 Wien Austria
Contact person: Christoph Hofinger
E-mail: ch@sora.at

Hardik Fintrade Pvt Ltd.
227, Shree Ram Cloth Market,
Opposite Manilal Mansion,
Revdi Bazar, Ahmedabad 380002
India
Contact person: Suresh Aswani
E-mail: m.aswani@hardikgroup.com

DAEDALUS S.A.
Avda. De la Albufera, 321, 1st floor
Madrid, E28031
Spain
Contact person: José Luis Martínez
E-mail: jmartinez@daedalus.es

Institute of Computer Science Polish Academy of Sciences 5 Jana Kazimierza Str.,Warsaw, Poland
Contact person: Maciej Ogrodniczuk
E-mail: Maciej.Ogrodniczuk@ipipan.waw.pl

Universidad Carlos III de Madrid
Advanced Databases Group
Department of Computer Science
Avda. de la Universidad, 30
Leganés
Spain
Tel: +34 91 624 95 00
Contact person: Paloma Martínez
E-mail: paloma.martinez@uc3m.es

Research Institute for Linguistics of the Hungarian Academy of Sciences
Benczúr u. 33., H-1068 Budapest, Hungary
Contact person: Tamás Váradi
Email: varadi.tamas@nytud.mta.hu

Executive Summary

This deliverable describes the deployment of web services related to the extension of the TrendMiner project to new domains (here: the health domain) and languages (here: Spanish, Polish, Hungarian).

Three new prototypes have been developed in the health and financial domains for the Spanish language. The first one shows how drugs, diseases and adverse effects of drugs are mentioned in social media, including blogs and social networks in Spanish. The second prototype takes benefit of the TrendMiner platform to show corporate reputation information in the financial domain in Spanish. The third one has been developed as a complement of the WP6 financial use case and aims to identify occupation relations between people and companies in news articles in Spanish.

Two extensions of the political use case have been provided. One extension is dealing with the Polish language, where tweets in the political domain have been collected and analysed by applying part of speech tagging and sentiment analysis. The second extension is dealing with the Hungarian language, for which Facebook data in the political domain has been linguistically and semantically analysed for detecting opinions and psychological and socio-psychological phenomena.

The final state of the TrendMiner platform, also considering the extensions described in this deliverable and in D10.1 “Newly generated domain-specific language data and tools”, is presented in details in D5.3.2 “Real-time Stream Media Processing Platform and Cloud-based Deployment v2”.

Contents

TrendMiner Consortium **2**
Executive Summary **3**
Contents **4**
1. A real time prototype for health monitoring in social networks **5**
1.1 Architecture 5
1.1.1 Data warehouse 6
1.1.2 Gatherer processes and data sources 6
1.1.3 Inquirer processes 8
1.1.4 Visualization module and prototype features 10
2. A new financial use case for the TrendMiner user interface **15**
2.1. Measuring companies reputation in Spanish 16
2.1.1. Use case specification 16
2.1.2. TrendMiner platform integration 17
2.1.3. Reputation analysis through the TrendMiner platform 18
2.1.4. Conclusions 20
2.2. Complementing the Spanish financial use case: detecting people positions in companies 20
2.2.1. Use case description 20
2.2.2. Conclusions 21
3. Political use case for the Polish language **22**
3.1. Analyzing political tweets in Polish 22
3.2. Polish political Twitter account list 22
Initial version of the list 22
Extending the list with automatically retrieved accounts 23
3.3. Tweet collection tools 23
3.3.1. Collecting scheme 23
3.3.2. Current setting 26
3.4. Visualisation of Polish tweets 27
3.4.1. Architecture 27
3.4.3. Tweet uploading tool 31
4. Hungarian Political Use Case **34**
5. Conclusions **35**
6. Bibliography and references **36**

1. A real time prototype for health monitoring in social networks

This section describes the architecture and capabilities of the prototype developed in the health domain to monitor mentions of drugs, diseases, adverse effects, indications and other health related issues. The resources developed to carry out this monitoring process are described in deliverable D10.1 [8]. Integration of this prototype in TrendMiner was not foreseen at the level of components, but is pursued at the level of the generated semantic annotations. This decision was motivated by the time at our disposal, as the new partners involved in the development of the prototype joined TrendMiner in the last year of the project. Therefore the description, below, of the implemented system is not to be confused with the TrendMiner platform, although it was in its design inspired by the TrendMiner architecture.

1.1 Architecture

The system is composed by five main components, shown in Figure 1. The central component is the ***data warehouse***, which acts as a core information repository. A ***set of gatherer processes*** feed the system with texts related to the health domain; while another ***set of concurrent inquirer processes*** process the collected texts using a ***GATE Annotation Pipeline*** (see [6]). Finally, the ***visualization module*** provides an interface to analyse the data and thus, help the user to discover data insights.

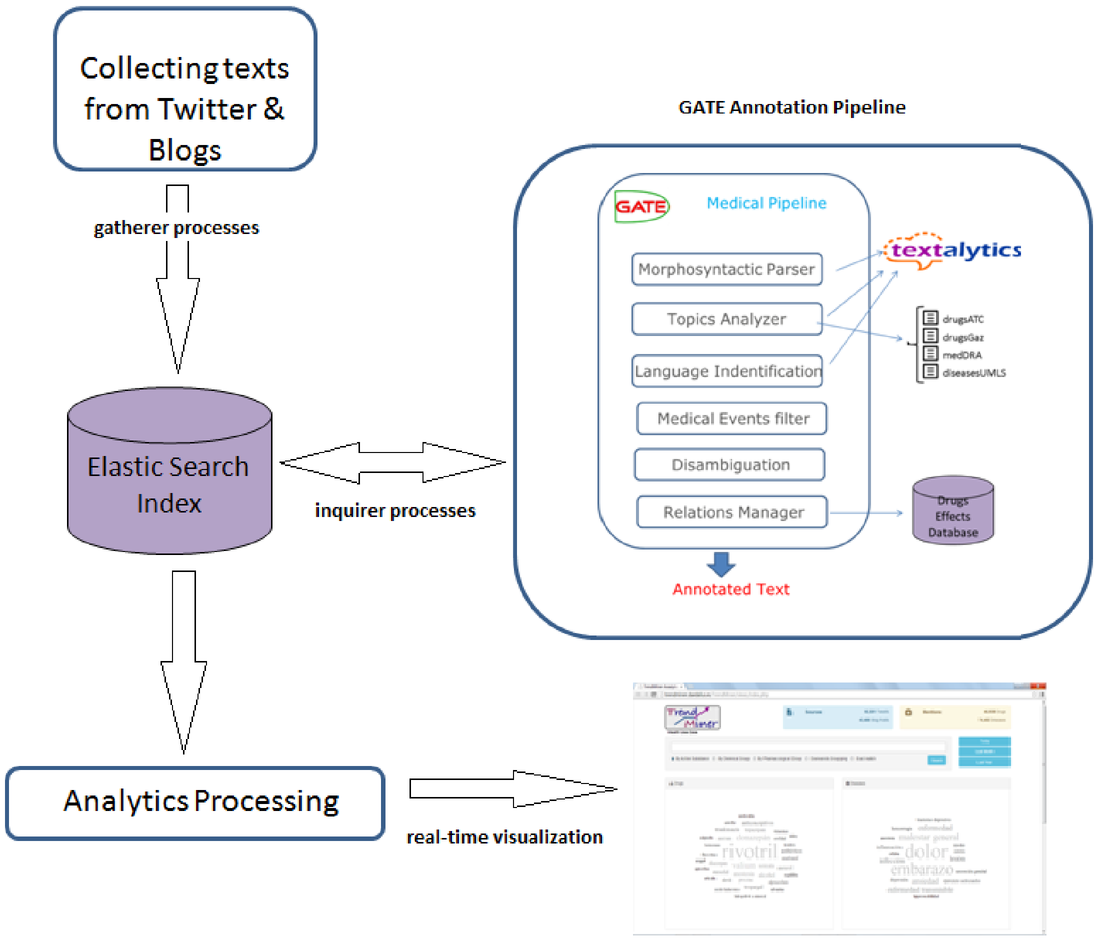


Figure 1. Health Use Case Prototype Architecture

1.1.1 Data warehouse

This module is responsible for storing efficiently the high volume of real-time data from social networks that the system manages, as well as for providing advanced search functionalities that allow the visualization module to generate complex analytics.

The data warehouse is based on Elasticsearch [1], which is a flexible, powerful, open source, distributed and real-time search and analytics engine. Some of the key factors that made us take the decision of choosing this architecture are its distributed capabilities and the fact that it can easily and horizontally scale when the system growth starts affecting performance. Furthermore, Elasticsearch runs on top of Apache Lucene[1], so that it offers quite complex search capabilities, high-performance and is trustworthy, due to its well-known reliability.

An example of an indexed document is shown in Figure 2. It contains information about a tweet and the user that created it, and, under the object “tags”, the different annotations created by the system. Each of the annotations is an array of objects, each containing the form, the identifier of the term, and the offset in which the mention appears in the text.

The field “relations” distinguishes between the element “drug”/”drug_id” and “disease”/”disease_id” and also contains another field “isDocAE”, which is a boolean field. If true, the relation is already present in our database and thus, is a known relationship.

1.1.2 Gatherer processes and data sources

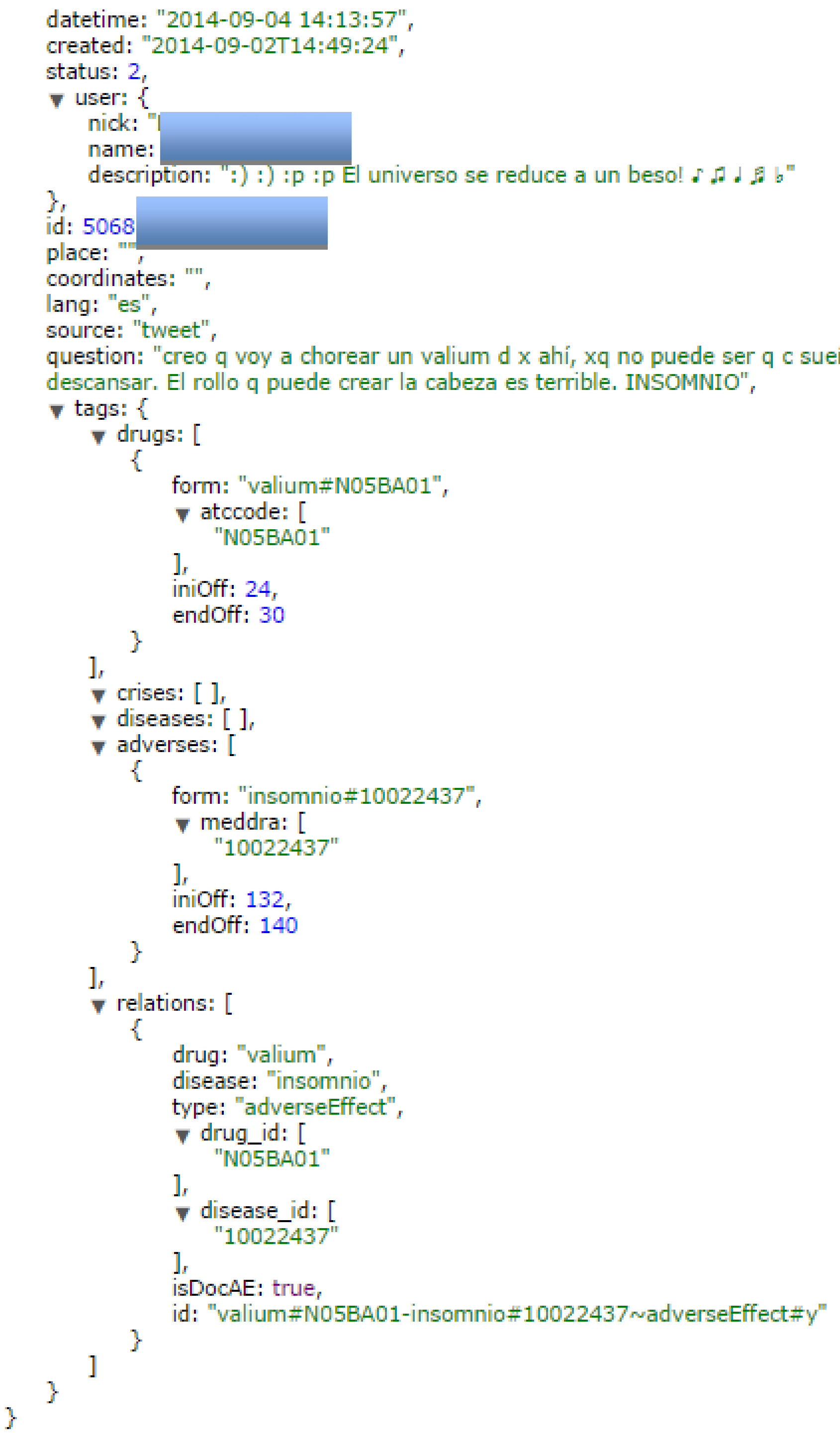
This module may be considered as the feeder of the system. We have deployed two different subtypes of those gatherer processes regarding the type of data source they are querying.

First, a process to query the Twitter APIs has been developed in order to collect tweets compliant to certain filters, in our case, tweets that contain specific keywords like drug or disease names and which are written in Spanish. This was necessary, as the gatherers used so far in TrendMiner were primarily focusing on the timeline of certain users, but not on keywords and language. This Twitter-gatherer processes work in real time, feeding the system with new tweets. In order to do so, the Twitter Streaming API [9] is used. In such mode, first a streaming connection request is issued and after that, Twitter streams tweets to the process as they arrive. An example of some of the keywords used to filter the Twitter stream is shown on Figure 3.

On the other hand, a crawling process that collects posts from Saluspot [5] has been developed. Saluspot is a Spanish website that aims to bring closer in an online platform common people and medicine experts. This website allows its users to ask free of charge and anonymously about issues related to health, lifestyle and drugs. Once a question is posted any of the registered, accredited doctors can answer and even multiple answers are possible. The system continuously evolves, but in principle, each question contains information about the user’s gender and age, the date of posting and one or more answers together with the identity of the doctor who

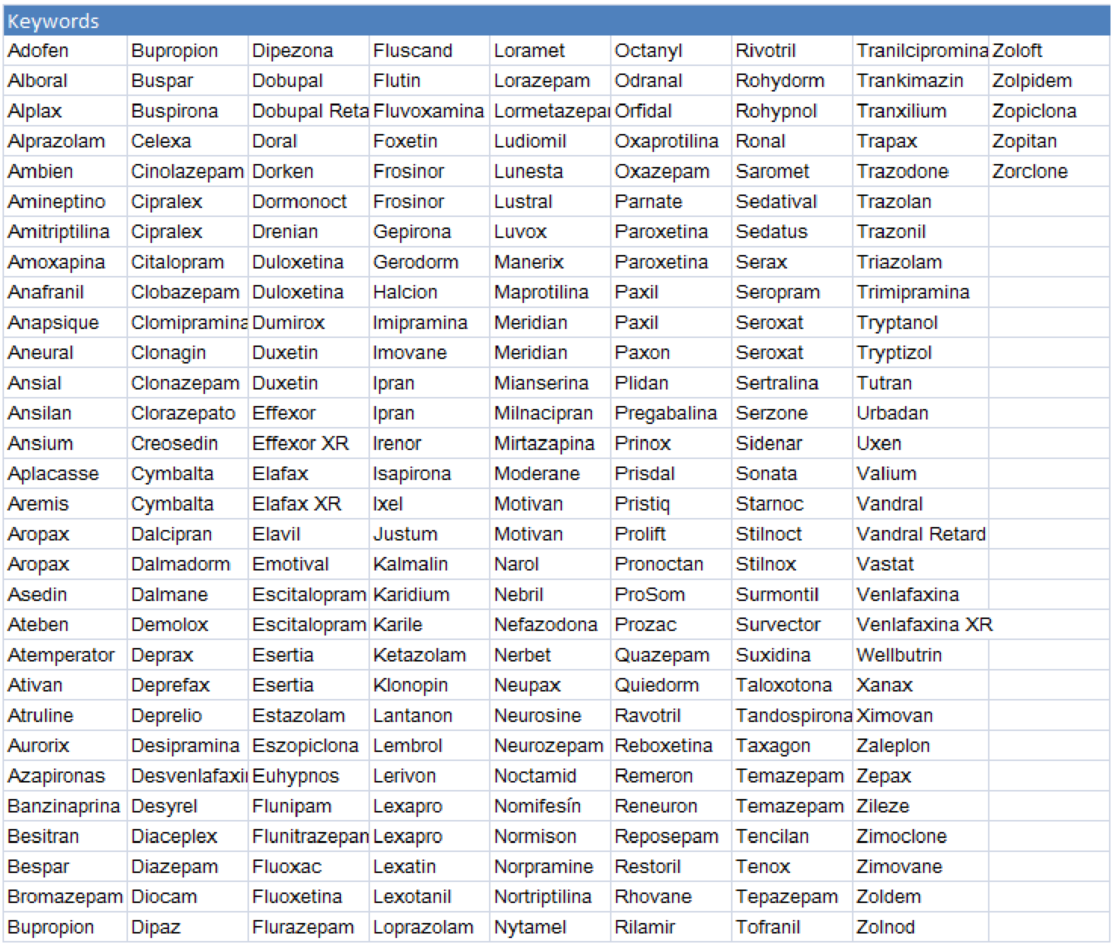
[1] See http://lucene.apache.org/core/

answered and a reliability measure based on the number of doctors who accepted to tackle this particular question.



```
    datetime: "2014-09-04 14:13:57",
    created: "2014-09-02T14:49:24",
    status: 2,
  ▼ user: {
        nick: "[redacted]
        name:
        description: ":) :) :p :p El universo se reduce a un beso! ♪ ♫ ♩ ♬ ♭"
    },
    id: 5068[redacted]
    place: "",
    coordinates: "",
    lang: "es",
    source: "tweet",
    question: "creo q voy a chorear un valium d x ahí, xq no puede ser q c sueño no pueda
    descansar. El rollo q puede crear la cabeza es terrible. INSOMNIO",
  ▼ tags: {
      ▼ drugs: [
            {
                form: "valium#N05BA01",
              ▼ atccode: [
                    "N05BA01"
                ],
                iniOff: 24,
                endOff: 30
            }
        ],
      ▼ crises: [ ],
      ▼ diseases: [ ],
      ▼ adverses: [
            {
                form: "insomnio#10022437",
              ▼ meddra: [
                    "10022437"
                ],
                iniOff: 132,
                endOff: 140
            }
        ],
      ▼ relations: [
            {
                drug: "valium",
                disease: "insomnio",
                type: "adverseEffect",
              ▼ drug_id: [
                    "N05BA01"
                ],
              ▼ disease_id: [
                    "10022437"
                ],
                isDocAE: true,
                id: "valium#N05BA01-insomnio#10022437~adverseEffect#y"
            }
        ]
    }
}
```

Figure 2. Example of an analysed document in the Elasticsearch index



Keywords								
Adofen	Bupropion	Dipezona	Fluscand	Loramet	Octanyl	Rivotril	Tranilcipromina	Zoloft
Alboral	Buspar	Dobupal	Flutin	Lorazepam	Odranal	Rohydorm	Trankimazin	Zolpidem
Alplax	Buspirona	Dobupal Reta	Fluvoxamina	Lormetazepa	Orfidal	Rohypnol	Tranxilium	Zopiclona
Alprazolam	Celexa	Doral	Foxetin	Ludiomil	Oxaprotilina	Ronal	Trapax	Zopitan
Ambien	Cinolazepam	Dorken	Frosinor	Lunesta	Oxazepam	Saromet	Trazodone	Zorclone
Amineptino	Cipralex	Dormonoct	Frosinor	Lustral	Parnate	Sedatival	Trazolan	
Amitriptilina	Cipralex	Drenian	Gepirona	Luvox	Paroxetina	Sedatus	Trazonil	
Amoxapina	Citalopram	Duloxetina	Gerodorm	Manerix	Paroxetina	Serax	Triazolam	
Anafranil	Clobazepam	Duloxetina	Halcion	Maprotilina	Paxil	Seropram	Trimipramina	
Anapsique	Clomipramina	Dumirox	Imipramina	Meridian	Paxil	Seroxat	Tryptanol	
Aneural	Clonagin	Duxetin	Imovane	Meridian	Paxon	Seroxat	Tryptizol	
Ansial	Clonazepam	Duxetin	Ipran	Mianserina	Plidan	Sertralina	Tutran	
Ansilan	Clorazepato	Effexor	Ipran	Milnacipran	Pregabalina	Serzone	Urbadan	
Ansium	Creosedin	Effexor XR	Irenor	Mirtazapina	Prinox	Sidenar	Uxen	
Aplacasse	Cymbalta	Elafax	Isapirona	Moderane	Prisdal	Sonata	Valium	
Aremis	Cymbalta	Elafax XR	Ixel	Motivan	Pristiq	Starnoc	Vandral	
Aropax	Dalcipran	Elavil	Justum	Motivan	Prolift	Stilnoct	Vandral Retard	
Aropax	Dalmadorm	Emotival	Kalmalin	Narol	Pronoctan	Stilnox	Vastat	
Asedin	Dalmane	Escitalopram	Karidium	Nebril	ProSom	Surmontil	Venlafaxina	
Ateben	Demolox	Escitalopram	Karile	Nefazodona	Prozac	Survector	Venlafaxina XR	
Atemperator	Deprax	Esertia	Ketazolam	Nerbet	Quazepam	Suxidina	Wellbutrin	
Ativan	Deprefax	Esertia	Klonopin	Neupax	Quiedorm	Taloxotona	Xanax	
Atruline	Deprelio	Estazolam	Lantanon	Neurosine	Ravotril	Tandospirona	Ximovan	
Aurorix	Desipramina	Eszopiclona	Lembrol	Neurozepam	Reboxetina	Taxagon	Zaleplon	
Azapironas	Desvenlafaxi	Euhypnos	Lerivon	Noctamid	Remeron	Temazepam	Zepax	
Banzinaprina	Desyrel	Flunipam	Lexapro	Nomifesín	Reneuron	Temazepam	Zileze	
Besitran	Diaceplex	Flunitrazepan	Lexapro	Normison	Reposepam	Tencilan	Zimoclone	
Bespar	Diazepam	Fluoxac	Lexatin	Norpramine	Restoril	Tenox	Zimovane	
Bromazepam	Diocam	Fluoxetina	Lexotanil	Nortriptilina	Rhovane	Tepazepam	Zoldem	
Bupropion	Dipaz	Flurazepam	Loprazolam	Nytamel	Rilamir	Tofranil	Zolnod	

Figure 3. Example of keywords used to query the Twitter Streaming API in order to get medical related texts

However, due to the intrinsic complexity of the crawling process and the lack of an API that would have been able to alert our system, notifying that new posts (or new answers to the already collected posts) are available, this system does not work in real-time, but collects posts as a snapshot of the currently available information in this website. Also, in this first version, only the most reliable answer was collected.

We have also considered other sources of information like forumclinic.org [3] and enfemenino.com [2]. However, *forumclinic* is now (since March 2014) closed due to improvement works, and *enfemenino* is a general purpose forum with some sections related to our topic, but where much informal language is used and the topics are not curated as they seem to be in *Saluspot*. Hence, we decided to consider only Twitter and *Saluspot* at this stage of the project and leave those two other information sources for the future as possible improvements of the system.

It is important to remark that this set of processes just collect the source texts from the different data sources and store them in our data warehouse. The only filtering carried out is the use of keywords (see the examples in Figure 3) when interacting with the Twitter Streaming API.

1.1.3 Inquirer processes

The module dedicated to this specific task of annotation and post-filtering is the ***set of concurrent inquirer processes***, which make use of the ***GATE Annotation Pipeline.*** For the sake of simplicity, we have decided to explain those two processes as two different modules, to clearly mark the distinction between the two important tasks that they carry out: the semantic annotation of text and the control of concurrency that the system needs to carry out, in order to comply with some of the well-known V's of the Big Data paradigm. The system requires access to a big **V**olume of data, at high **V**elocity and guarantying the **V**eracity of the data.

Basically, the inquirer processes take care of the concurrent access to the data warehouse. There is the possibility to run several annotation processes; the inquirer provides the exclusion mechanisms among the different annotation processes assuring avoiding data corruption and hence, assuring veracity. Each of those processes seek for the latest created and unlocked text, reserve it by locking it, and then execute an instance of the GATE Annotation Pipeline, which semantically annotates the text that is being considered. Finally, it stores the response given by the pipeline back to the Elasticsearch data warehouse.

The GATE Annotation Pipeline is described in more detail in deliverable 10.1 [8]. Basically it is a pipeline composed of 6 different stages:

- Language Identification,
- Morphosyntactic Parsing,
- Topics Extraction,
- Medical Events Filtering,
- Disambiguation Module and
- Relations Manager.

The Language Identification stage filters out every text that is not written in Spanish. The gatherer already requires from the Twitter API for texts only in Spanish, but since this may sometimes fail, another filtering step is performed while analysing the document. The identification is made by the Textalytics Language Identification API [7], which uses statistic techniques based on n-grams.

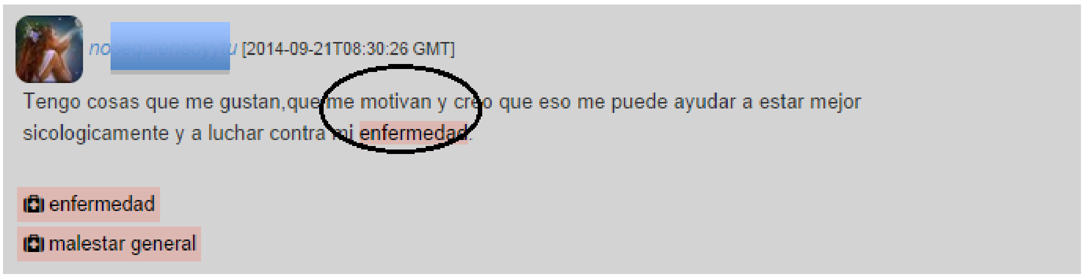
The Morphosyntactic Parsing is performed by the Textalytics Lemmatization, PoS Tagging and Parsing APIs [7]. This step is of great importance in order to perform the disambiguation step that comes later, due to the high ambiguity that exists in medical texts. In our approach, the morphosyntactic information from the parser, which is based on linguistic features, is used in order to avoid annotations of drugs and diseases whose lexical representations have been tagged as verbs, articles, adjectives or any other class different to noun.

The Topics Extraction module also benefits from the Textalytics Topics Extraction API [7]. In fact, it uses an advanced functionality of it, since several users defined dictionaries were created for this application. The user defined dictionaries allow users of the Textalytics APIs to add new terms to the ontology, providing also valuable information like semantic linked data, morphosyntactic tag, synonyms, canonical form, etc. (see D10.1 for more information on those terminological resources).

In order to ease the use of Textalytics APIs inside of the well known and extensively used GATE Platform, a plug-in was created and made public through a plugin repository which is available from the Textalytics Website[2].

In order to increase the precision of the system, several disambiguation and filtering steps are steps are performed after the default annotation performed by Textalytics. This post-filtering filtering uses linguistic features like the morphosyntactic information provided by the parser, parser, together with co-occurrence information between drugs and diseases to filter out terms out terms not likely to be real mentions of medical events. If we look at the example shown in shown in

Figure 4, "*motivan*" is a term in our dictionary, because it is an antidepressive whose main active substance is "*paroxetina*". However, in this case the morphosyntactic parser detects that it is acting as a verb ("motivar" is the Spanish form of "to motivate") in the sentence and doesn't tag it as a drug.


[redacted] [2014-09-21T08:30:26 GMT]

Tengo cosas que me gustan,que me motivan y creo que eso me puede ayudar a estar mejor sicologicamente y a luchar contra mi enfermedad.

enfermedad

malestar general

Figure 4. Example of tweet annotated after the post-filtering stage

Finally, another plug-in aims to discover possible relations between terms, actually between drugs and diseases or adverse effects, classifying them into adverse effects, indications or pairs that hold a possible relationship. The two first possible classifications are relations that were extracted from several sources and stored in our database of relations between drugs and effects. In contrast, the latter group has been created to point out possible un-catalogued or unknown relations that may be discovered due to situations of high recurrence pointed out by the system.

It is important to remark, that the semantic annotation task is the most time consuming task and constitutes the bottleneck of the system. As we said, the inquirer processes select the texts to process in descending order of creation, so that the most recent information is available first to the user. We consider that it may be more valuable to have the latter information first even when due to hardware requirements, it is not possible to analyse every incoming text in real time. Therefore, the system works in such a way that if the input rate of incoming texts is higher than the multi-threaded annotation rate, such texts will not be processed automatically in real time, but once this peak situation is reversed and the system manages to annotate at a higher rate than the indexing of new documents, it will start annotating the unprocessed documents.

1.1.4 Visualization module and prototype features

Finally, the visualization component performs the analytics in order to communicate to the final user of the system meaningful relations, patterns of co-occurrence, and other data insights.

[2] See https://textalytics.com/GATE-plugin-text-analytics

The interface provides several search modes based on the ATC code[3] of the drugs. This classification system and the usage we make of it in our annotation pipeline are described in deliverable D10.1 [8]:

- *By Active Substance* (level 5 ATC): The 5th level in the ATC code specification distinguishes drugs by two digits regarding to their "active substance main group". When this mode is selected, if the ATC code of the substance that is being searched is available, every substance that has the exact same ATC code (that means they share active substance main group) is presented in the interface.
- *By Chemical Main Group* (level 4 ATC): This search mode groups together every mention of substances that share a chemical group in the ATC scheme- *By Pharmacological Main Group* (level 3 ATC): Exactly the same as the previous two groups but grouping up to the 3rd level.
- *Downwards Grouping*: This search mode is a little different than the three explained before, since it focuses on the element searched to decide up to what level it should group. It basically gets the element that defines the search, which actually is a node (or several nodes) in the ATC structure tree, and groups together every element below this node or nodes.
- *Exact Match*: It looks for specific mentions of the exact terms, allowing a bit of fuzziness, regardless of the ATC code of the mentions.

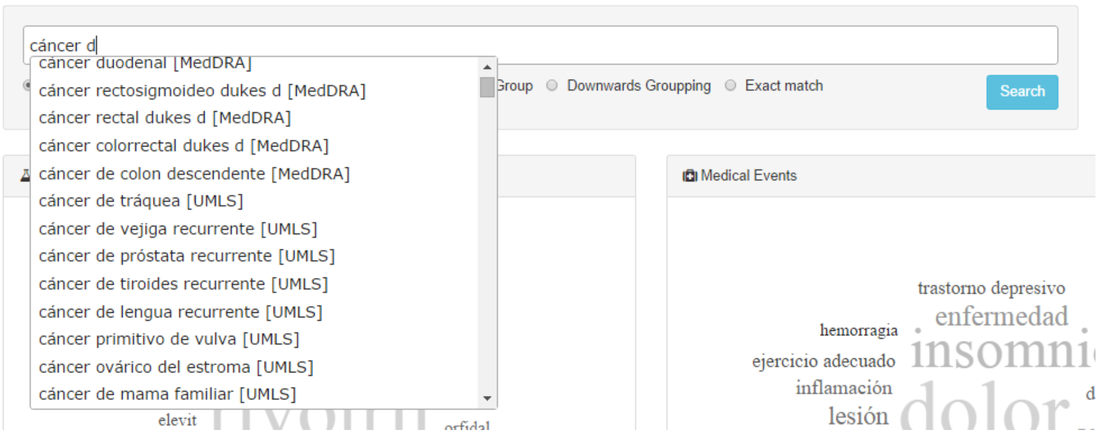
An example showing the differences between the different search modes for the query "fluoxetina" is shown in Figure 10 to Figure 13. It is clear that the number of mentions and the relations displayed change between the different search modes.

The prototype allows viewing the annotated source texts that match a specific search, focusing on their drug and disease mentions and showing the discovered relations, like shown in the example of Figure 6.

Also, in order to facilitate the use of the system, the search box is designed to display every every possible term in our vocabulary. The resources used to build the different dictionaries are dictionaries are also compiled and indexed into another ElasticSearch index, using an n-gram n-gram analyser at index time (using from 2 to 20 grams for each word indexed). In contrast, at contrast, at search time, the standard analyser is used (standard *tokenizer*, lower case token filter token filter and stopwords filter). By doing this, the system quickly responds to the user of the user of the system with the hundred most likely terms that match the input provided by the user. by the user. Another advantage of this system is that it allows looking for both the canonical form canonical form and any of the synonyms or alias that the term has in our database. See

Figure 5 for more details.

[3] ATC stands for „Anatomical Therapeutic Chemical" and the ATC Classification system is "is used for the classification of active ingredients of drugs. See http://www.wikiwand.com/en/Anatomical_Therapeutic_Chemical_Classification_System for more details.



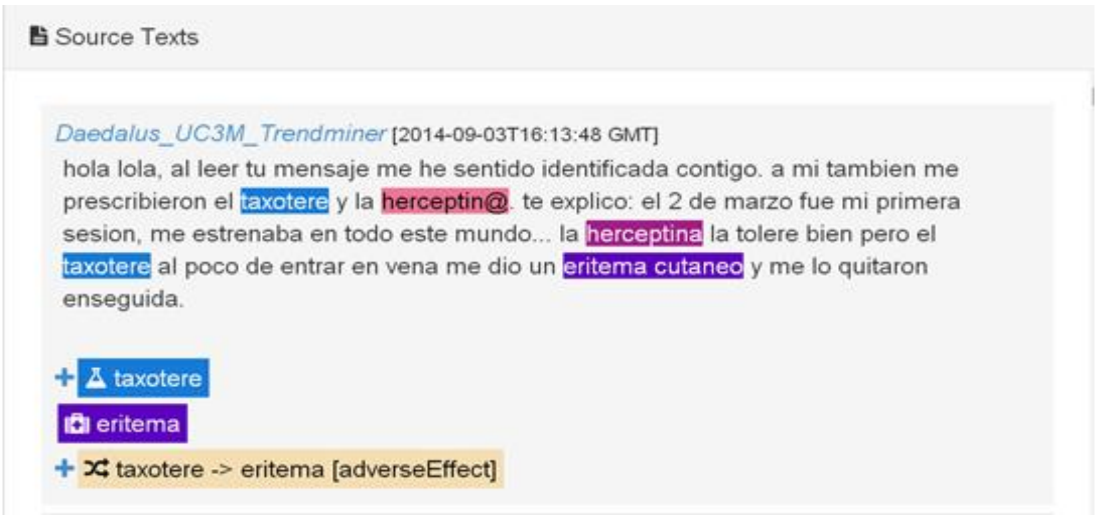
cáncer d

cáncer duodenal [MedDRA]
cáncer rectosigmoideo dukes d [MedDRA]
cáncer rectal dukes d [MedDRA]
cáncer colorrectal dukes d [MedDRA]
cáncer de colon descendente [MedDRA]
cáncer de tráquea [UMLS]
cáncer de vejiga recurrente [UMLS]
cáncer de próstata recurrente [UMLS]
cáncer de tiroides recurrente [UMLS]
cáncer de lengua recurrente [UMLS]
cáncer primitivo de vulva [UMLS]
cáncer ovárico del estroma [UMLS]
cáncer de mama familiar [UMLS]

Figure 5. Example showing some of the options provided by the search box against an example query

Also, individual bar graphs that aggregate the number of mentions of discovered relations of the different kinds for the texts that match the query are presented, as well as co-occurrences between drugs (drug-drug) and between drugs and diseases (drug-disease). Examples are given in Figure 7 and Figure 8.

Besides, we provide linked data for both drugs and diseases or adverse effects, linking each term with information from CIMA[4], for drugs mentions and from UMLS[5] and MedDRA[6], for diseases and adverse effects. These resources are described in [8].



Source Texts

Daedalus_UC3M_Trendminer [2014-09-03T16:13:48 GMT]

hola lola, al leer tu mensaje me he sentido identificada contigo. a mi tambien me prescribieron el taxotere y la herceptin@. te explico: el 2 de marzo fue mi primera sesion, me estrenaba en todo este mundo... la herceptina la tolere bien pero el taxotere al poco de entrar en vena me dio un eritema cutaneo y me lo quitaron enseguida.

+ taxotere

eritema

+ taxotere -> eritema [adverseEffect]

Figure 6. Example of Source Text presented by the Visualization module

[4] CIMA is an Online Information Center for Medicines. See http://www.aemps.gob.es/cima/ for more details.

[5] UMLS stands for „Unified Medical Language System. See http://www.nlm.nih.gov/research/umls/ for more details.

[6] MedDRA stands for Medical Dictionary for Regulatory Activities. See http://www.meddra.org/ for more information.

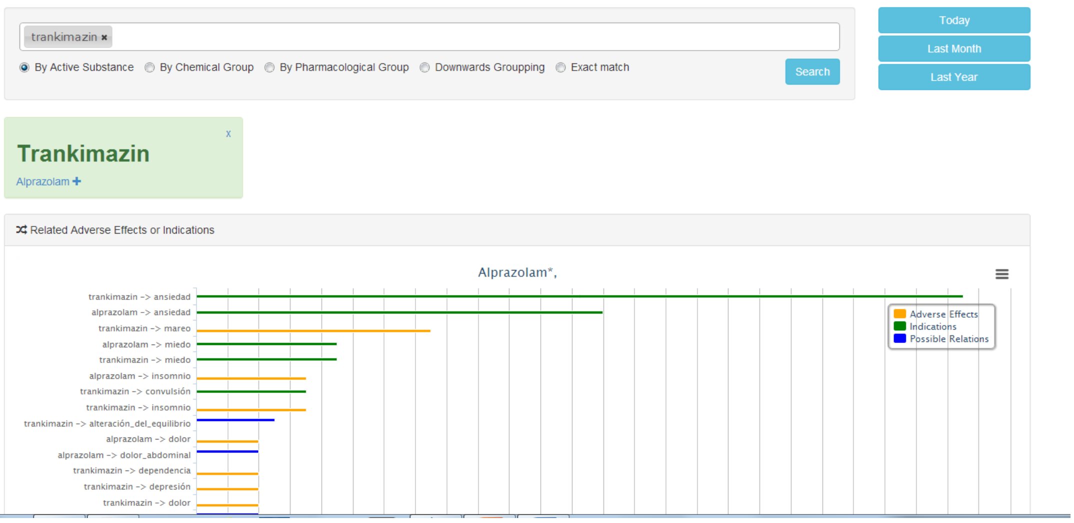


Figure 7. Graph showing aggregate count of known relations and possible new ones for the query: "trankimazin", grouping by Active Substance.

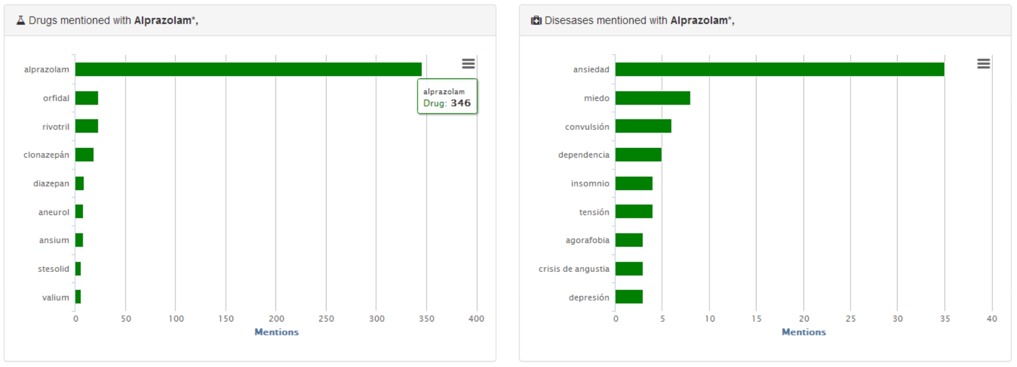


Figure 8. Graph showing co-occurrence aggregated information for the same query as in Figure 3

Finally, the system also presents information about the evolution of mentions against time with different granularity through a timeline graph like the one shown in Figure 9. All graphs have been developed using Highcharts [4].

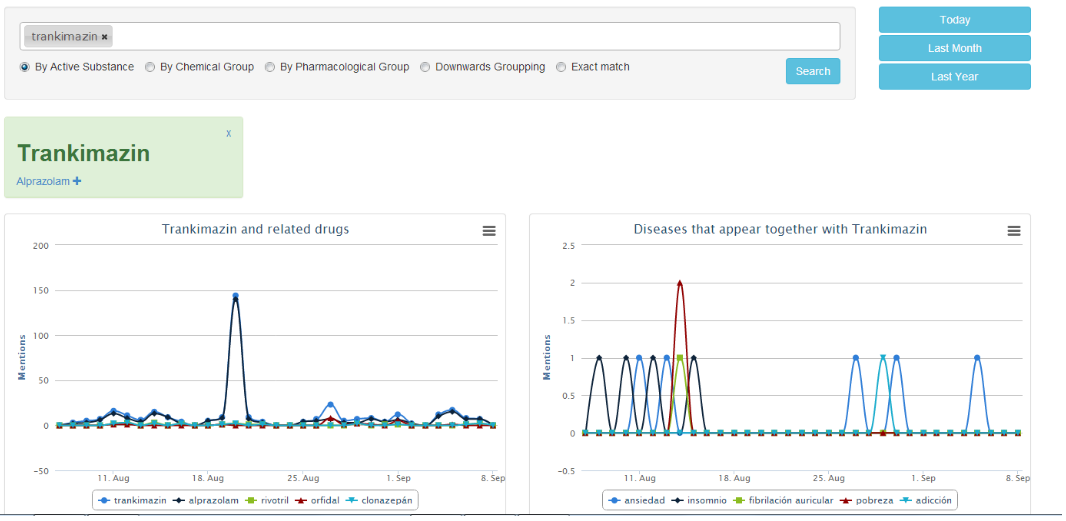


Figure 9. Graph showing time based evolution of mentions for the query: "trankimazim" groupping by Active Substance

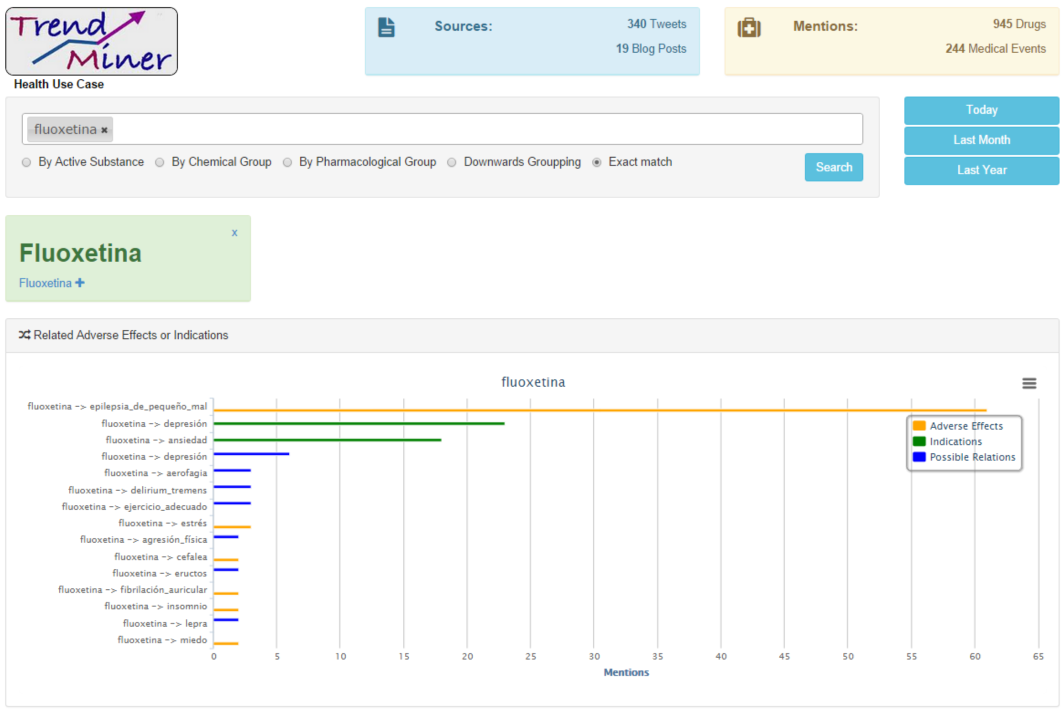


Figure 10. Example of search mode "Exact match"

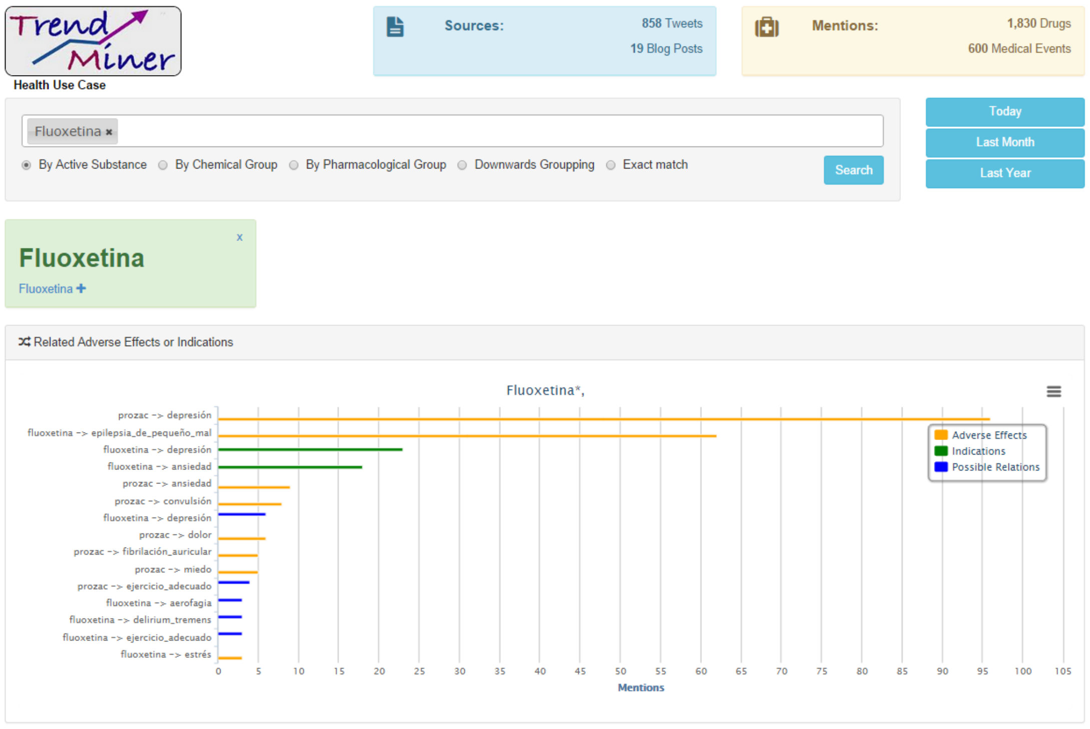


Figure 11. Example of search mode grouping "By Active Substance"

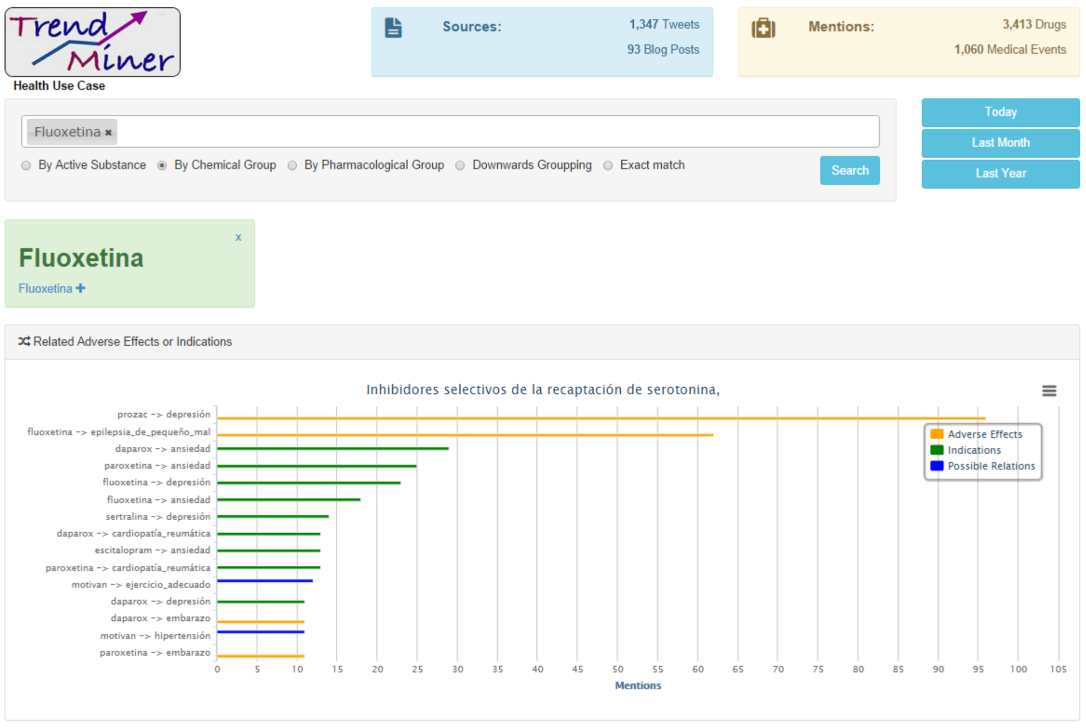

Sources: 1,347 Tweets, 93 Blog Posts
Mentions: 3,413 Drugs, 1,060 Medical Events
Fluoxetina
By Active Substance [✓ By Chemical Group] By Pharmacological Group Downwards Groupping Exact match

Figure 12. Example of search mode grouping "By Chemical Group"

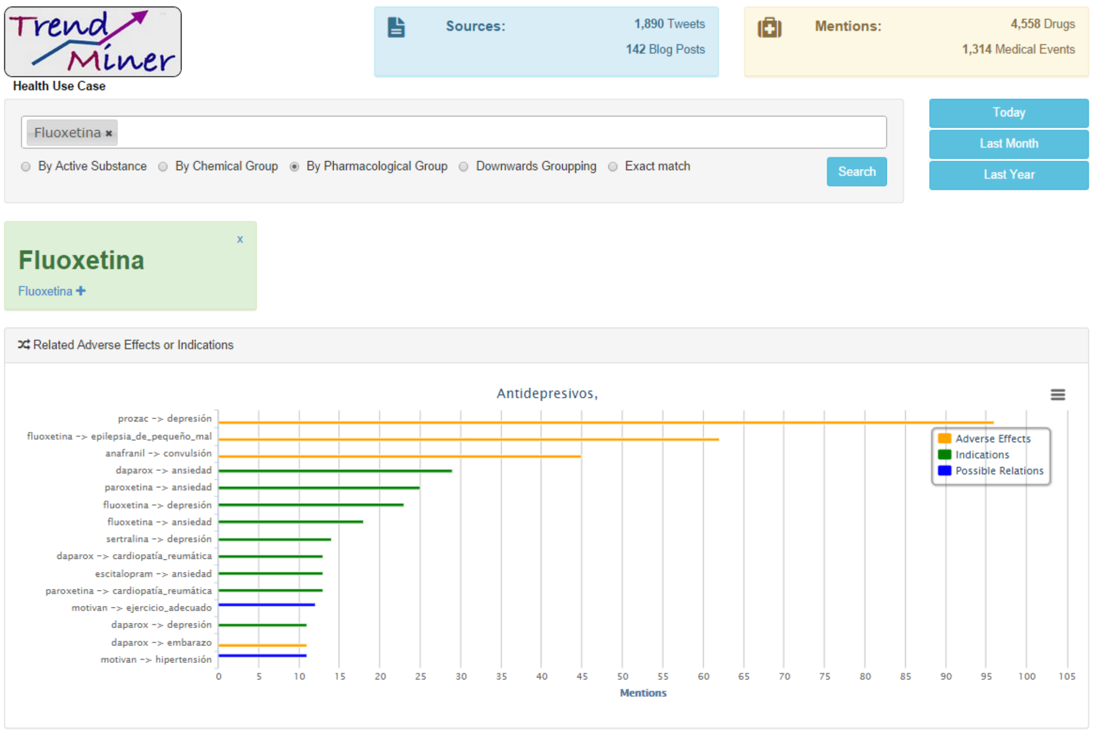

Sources: 1,890 Tweets, 142 Blog Posts
Mentions: 4,558 Drugs, 1,314 Medical Events
Fluoxetina
By Active Substance By Chemical Group [✓ By Pharmacological Group] Downwards Groupping Exact match

Figure 13. Example of search mode grouping "By Pharmacological Group"

2. A new financial use case for the TrendMiner user interface

Differently to the new ehealth use case in Spanish, the extensions of the financial use case for the Spanish language have been integrated in the TrendMiner platfrom.

2.1. Measuring companies reputation in Spanish

Nowadays, social media is one of the channels used by people to share thoughts, ideas, experiences and opinions about companies. Reputation measurement is not a new challenge, and there have been many initiatives to measure reputation in traditional news media, where press clipping companies provide companies with data about their presence in newspapers, TV channels and so on. Examples of those companies are Factiva[7], Lexis Nexis[8] (specialized in the legal domain) or, in the Spanish market, Acceso Group[9]. Of course, these companies have moved from the traditional media market to include Internet and user generated content.

Different models have been defined to standardize the way reputation is defined, allowing comparisons among different companies. The two most common models are RepTrak®[10] and Merco®[11]. Both have similar grounding approaches, based on the identification and definition of several dimensions to cover all stakeholders involved in the environment around a company. In this way, not only economical issues are covered, such as the quality of the offer provided by a company, but also its behaviour towards its staff or the involvement in environment care.

Daedalus has defined a reputation model, taking into account those experiences, with the following dimensions:

- Oferta (Offer)
- Trabajo (Work)
- Integridad (Integrity)
- Estrategia y liderazgo (Strategy and Leadership)
- Innovación y flexibilidad (Innovation and Flexibility)
- Responsabilidad Social (Social Responsibility)
- Situación Financiera (Financial Situation)

This model has not been developed as part of the TrendMiner project; instead, the goal was to test the model in the financial use case implemented in TrendMiner, combining it with sentiment analysis approaches.

2.1.1. Use case specification

The TrendMiner platform (including data storage and visualization) has been applied to a use case in the financial domain in Spanish. The first step has been the selection of the economical issue to follow. In mid September 2014, Orange, the telecommunication company, announced the acquisition of Jazztel, one of the relatively small mobile and Internet connections providers in Spain. The biggest telecommunications company in Spain is Telefónica, so specific searches were done including all three companies. The idea was to compare reputation among all three companies applying the available model, adding opinion analysis information in the process. So, several online newspapers were crawled during three days, along with Twitter (using the company names as search keywords) and technology blogs. The

[7] Factiva, http://new.dowjones.com/factiva/
[8] LexisNexis, http://www.lexisnexis.com
[9] Acceso Group, http://www.acceso.com
[10] The RepTrak Framework, http://www.reputationinstitute.com/about-reputation-institute/the-reptrak-framework
[11] Merco, http://www.merco.info/en/pages/1-que-es-merco

collection contains 1.549 documents. The exact content of the collection is detailed in deliverable 10.1 [8].

2.1.2. TrendMiner platform integration

Documents have been analyzed using the Textalytics Reputation and Sentiment analysis tools[12]. The result has been transformed to the format demanded by the 'RDF Import Service' developed as part of the TrendMiner platform[13]. Figure 14 shows the content of one of the documents with the format requested by TrendMiner 'RDF Import Service'.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sesame: <http://www.openrdf.org/schema/sesame#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix fn: <http://www.w3.org/2005/xpath-functions#> .

<http://trendminer.eu/resources#tweet_514490682862739456>
        <http://purl.org/dc/elements/1.1/identifier> "514490682862739456" ;
        <http://purl.org/dc/terms/creator> "Beatriz" ;
        <http://purl.org/dc/terms/creator_id> "7735482" ;
        <http://purl.org/dc/elements/1.1/description> "En #Observatoriotecnológico de
@IndraCompany,Android One,llega
                                        Windows 9,Orange compra Jazztel,nuevos Apple el
21/10 http://t.co/c56eDDgc8H";        <http://purl.org/dc/elements/1.1/language> "es"
;
        <http://purl.org/dc/elements/1.1/sentiment> "0.0"^^xsd:float ;
        <http://purl.org/dc/elements/1.1/source> "twitter.com" ;
        <http://purl.org/dc/elements/1.1/date> "2014-09-23T21:05:12"^^xsd:dateTime ;
        <http://purl.org/dc/elements/1.1/hashtag> "#Observatoriotecnológico" ;
        <http://purl.org/dc/elements/1.1/location> "Madrid" ;
        <http://purl.org/dc/elements/1.1/location_uri>
<http://es.wikipedia.org/wiki/Madrid> ;
        <http://purl.org/dc/terms/ref_lab> "Orange" , "Jazztel" , "Apple" , "Indra" ;
        <http://purl.org/dc/terms/references> <es.wikipedia.org/wiki/Orange_S.A.> ,
<http://dbpedia.org/resource/Jazztel> ,

        <http://dbpedia.org/resource/Apple> ,
<http://es.wikipedia.org/wiki/Indra_Sistemas> ;
        <http://purl.org/dc/terms/subject> "Innovación" , "Finanzas" .
```

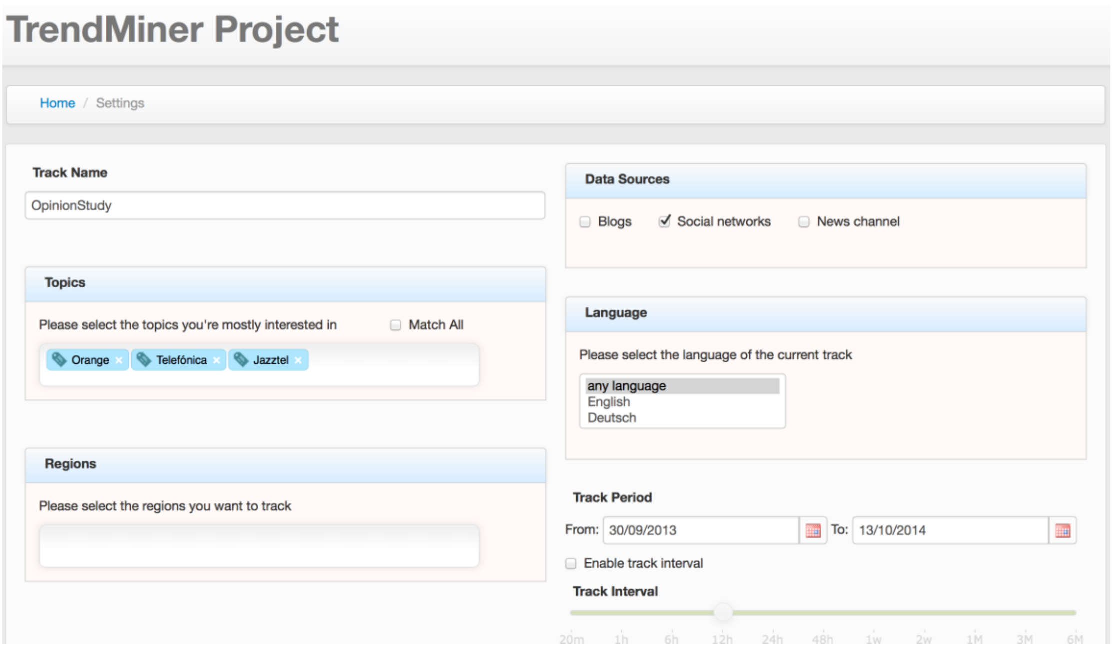
Figure 14. Example of a document processed with Textalytics and transformed to OWLIM.

The property 'dc-terms:subject' includes information about the text classification according to the dimensions defined in the reputation model. Besides, the 'dc-terms:references' property includes data about the different entities mentioned in the document. Finally, the result of the opinion analysis is included in the 'sentiment' property, through a value between -1.0 and 1.0 (negative to positive).

The user interface developed in TrendMiner (D4.2.1) can then be used to analyse the data stored, including reputation and sentiment. This analysis is described in the following section.

[12] See https://textalytics.com/semantic-api-media-analysis
[13] See see section 4 of D5.3.2 for more details

2.1.3. Reputation analysis through the TrendMiner platform



TrendMiner Project

Home / Settings

Track Name: OpinionStudy

Topics
Please select the topics you're mostly interested in ☐ Match All
Orange × Telefónica × Jazztel ×

Regions
Please select the regions you want to track

Data Sources
☐ Blogs ✓ Social networks ☐ News channel

Language
Please select the language of the current track
any language
English
Deutsch

Track Period
From: 30/09/2013 To: 13/10/2014
☐ Enable track interval
Track Interval
20m 1h 6h 12h 24h 48h 1w 2w 1M 3M 6M

Figure 15. Track definition screen for the opinion study with ‘Telefónica’, ‘Jazztel’ and ‘Orange’

With all the documents uploaded to the RDF Import Service of TrendMiner, a new track is defined using the interface provided. This track, called OpinionStudy, is defined around the keywords: Telefónica, Jazztel, Orange.

In this way, mentions can be compared among the three companies, along with the related sentiment and the concepts appearing together in the document collection.

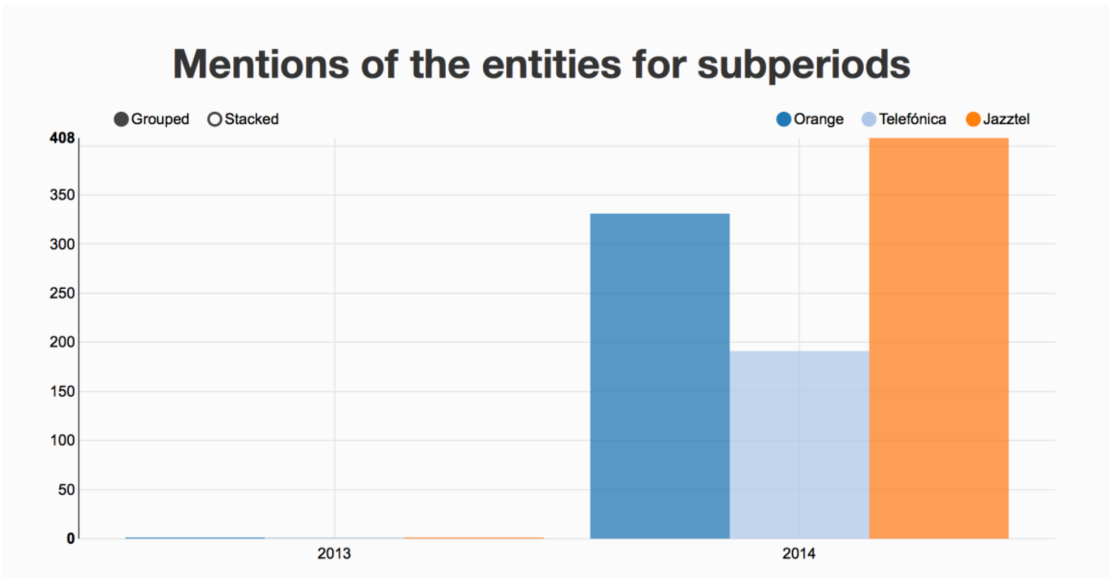


Figure 16. Mentions for each of the telecommunications companies under study

Figure 16 shows the difference in mentions for the three companies. As expected, Telefónica has fewer conversations, the news are about Jazztel and Orange. When sentiment analysis is considered, companies must be considered in an isolated track.

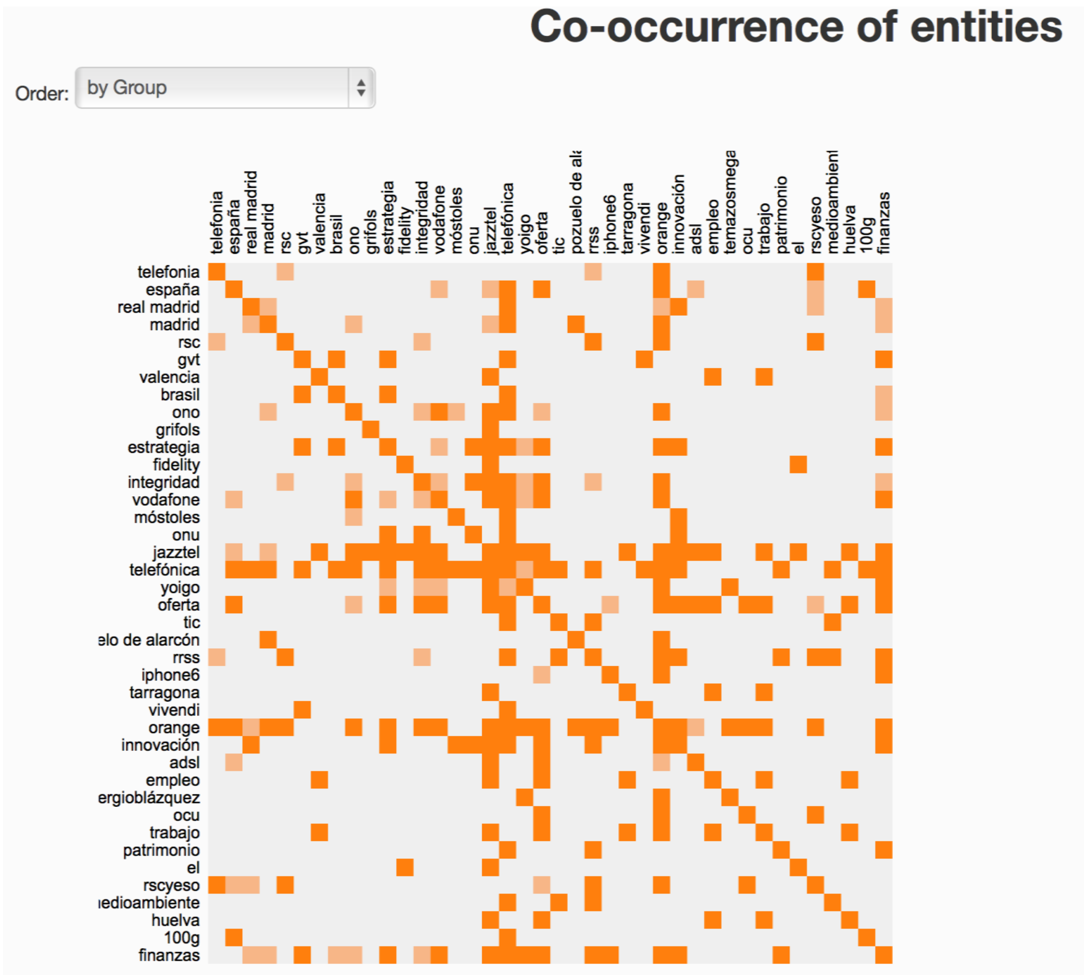


Figure 17. Word co-occurrence for the documents containing 'Telefónica' or 'Jazztel' or 'Orange'

If co-occurrence of entities is analysed (Figure 17), it is worth mentioning some examples: only Telefónica appears with the word 'medioambiente' (environment); the word 'empleo' (employment) is only mentioned with Jazztel; the acronym 'rsc' (corporate social responsibility), appears only with Orange while 'iphone6' appears with 'oferta' (offer) and 'Telefónica', while the rest of telecommunication companies in Spain can also sell the phone.

Regarding sentiment analysis, each company must be isolated in a track to get the sentiment for those documents where the given entity is appearing. Thanks to these tracks it is possible to see that Telefónica has a somehow positive mood, with a 0.19 value. On the other hand, Orange has a negative sentiment, with a -0.10 value; this could also be due to tweets about quality of service. Finally, Jazztel has a neutral sentiment, with a 0.03; this means that balanced messages (including positive and negative sentiments) have been crawled and included in the collection.

The different dimensions of the reputation model available can also be shown, comparing the number of mentions in each one. This is the view in Figure 18.

Messages related to the 'Offer' dimension are the most common ones, which makes sense (these include mobile offers, Internet access packages offers and so on). On the other hand, messages talking about employment are at second place in the list, while strategic related contents are less frequent.

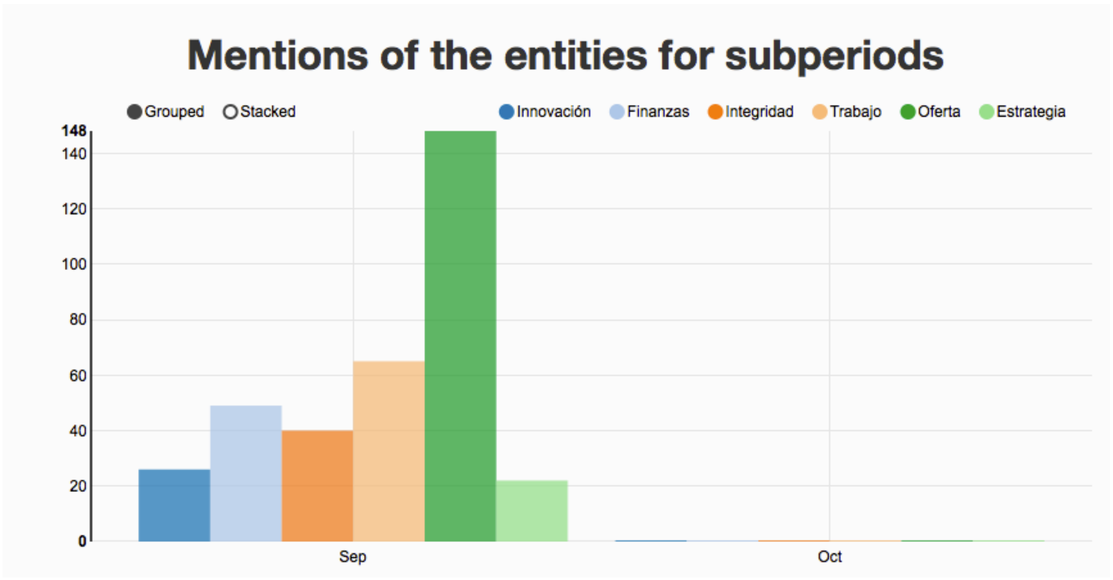


Figure 18. Number of documents related to the dimensions given by reputation model.

2.1.4. Conclusions

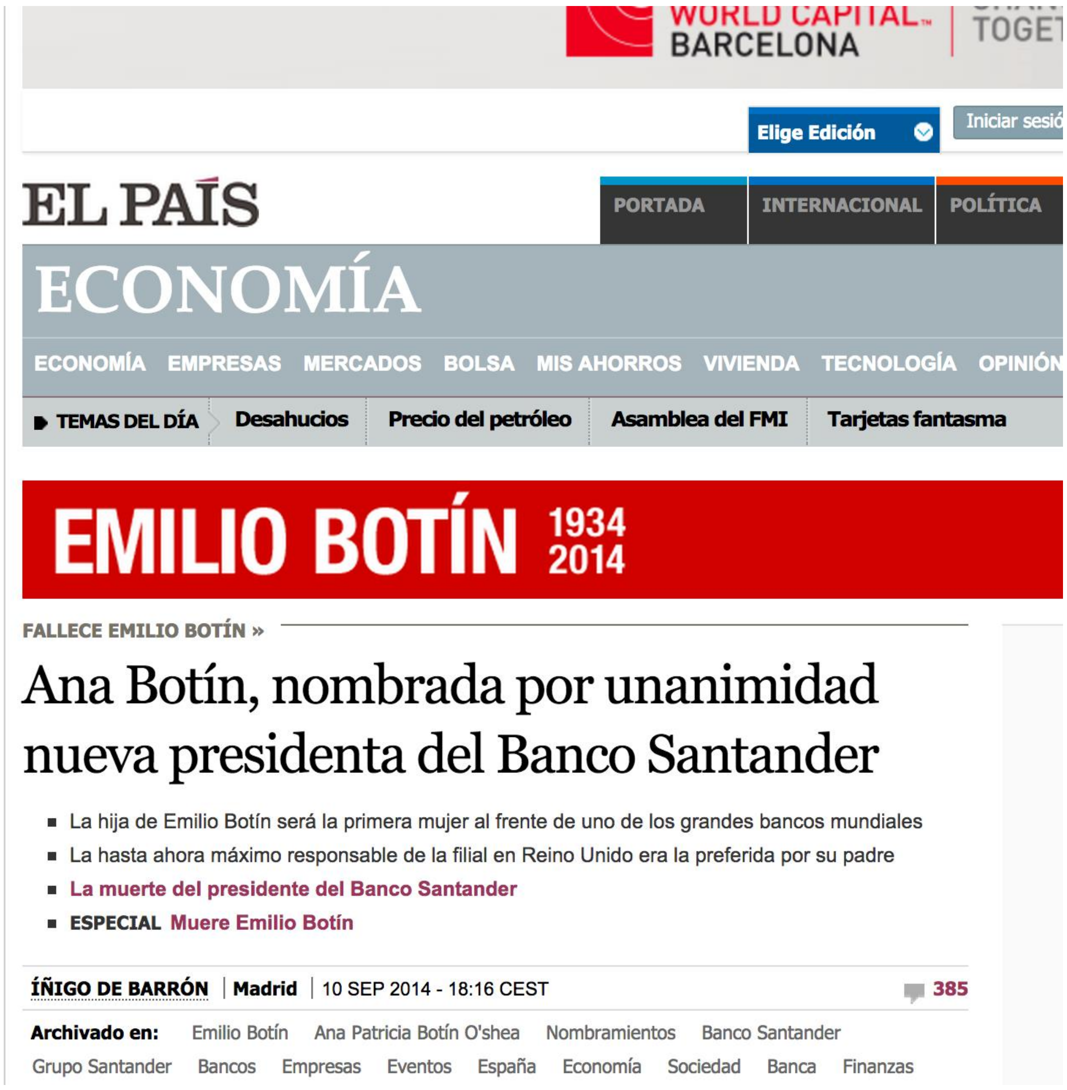
This experiment has been useful to demonstrate the potential of the TrendMiner platform. The combination of sentiment analysis with the tagging of texts according to the reputation model can be used as input data for a reputation analysis made by professionals. Our goal was to provide technology support to gather data for a reputation analysis in online social media, which has been covered in this use case. More information about this use case can be found in deliverable D10.1 [8].

2.2. Complementing the Spanish financial use case: detecting people positions in companies

2.2.1. Use case description

One of the most valuable data in the financial domain is the knowledge on who is in a given position in a company. Although it is possible to find this information through somehow official channels (for example, a stock market web site or a company providing financial data), there are many sites providing this information in the form of news. For example, for this proof of concept in TrendMiner, the site http://elpais.com/tag/nombramientos/a/ has been considered as an input source.

The application of linguistic processing, including named entity detection, to this kind of texts can lead to the identification of people and their positions in companies. For this purpose, a collection of job positions has been built and a set of patterns has been defined to extract the desired data. For more details about the technology applied to this use case, see the project deliverable D10.1 [8].



WORLD CAPITAL™ BARCELONA | TOGET

Elige Edición | Iniciar sesió

EL PAÍS | PORTADA | INTERNACIONAL | POLÍTICA

ECONOMÍA

ECONOMÍA EMPRESAS MERCADOS BOLSA MIS AHORROS VIVIENDA TECNOLOGÍA OPINIÓN

TEMAS DEL DÍA | Desahucios | Precio del petróleo | Asamblea del FMI | Tarjetas fantasma

EMILIO BOTÍN 1934 2014

FALLECE EMILIO BOTÍN »

Ana Botín, nombrada por unanimidad nueva presidenta del Banco Santander

- La hija de Emilio Botín será la primera mujer al frente de uno de los grandes bancos mundiales
- La hasta ahora máximo responsable de la filial en Reino Unido era la preferida por su padre
- La muerte del presidente del Banco Santander
- ESPECIAL Muere Emilio Botín

ÍÑIGO DE BARRÓN | Madrid | 10 SEP 2014 - 18:16 CEST 385

Archivado en: Emilio Botín Ana Patricia Botín O'shea Nombramientos Banco Santander Grupo Santander Bancos Empresas Eventos España Economía Sociedad Banca Finanzas

Figure 19. News example for the people-position-company relation extraction prototype

As an example, for the text shown in Figure 19, the output is:

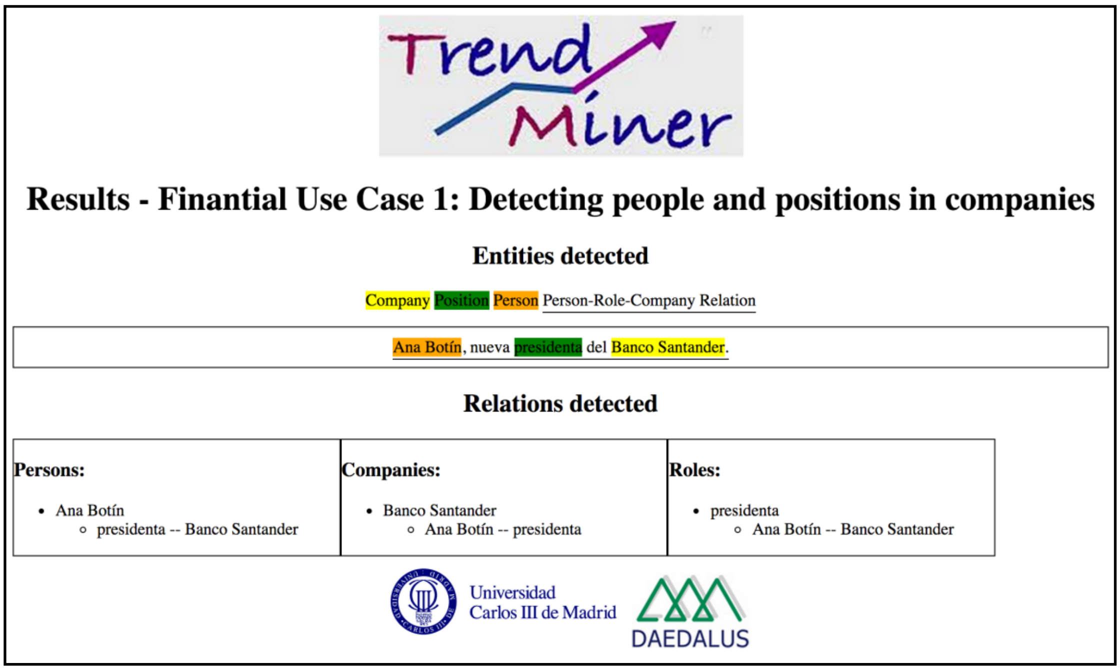
- *Person*: The name of the person – Ana Botín
- *Position*: The name of the position the person is in charge of – presidenta
- *Company*: The name of the company to which the person is working for – Banco Santander.

Figure 20 below shows the output provided by the prototype, available at http://163.117.129.57:8099/finantial.

2.2.2. Conclusions

The approach followed (described in detail in Deliverable D10.1 [8]) is an initial idea, just a proof of concept to see the feasibility of an analysis of this kind of texts based on linguistic patterns. The main problem in these approaches is the coverage of the patterns defined. There is always one sentence that is not covered by any of the patterns already available. For this reason, the focus has been made in the definition of an architecture that is easily extensible to new cases. In this prototype, patterns are defined through JAPE rules in a GATE application, which allows a seamless integration of within the Textalytics natural language processing APIs.

A full integration in TrendMiner will be possible after adapting the company ontology of TrendMiner to the Spanish language.



Results - Finantial Use Case 1: Detecting people and positions in companies

Entities detected

Company Position Person Person-Role-Company Relation

Ana Botín, nueva presidenta del Banco Santander.

Relations detected

Persons:	Companies:	Roles:
• Ana Botín ∘ presidenta -- Banco Santander	• Banco Santander ∘ Ana Botín -- presidenta	• presidenta ∘ Ana Botín -- Banco Santander

Figure 20. Result of the people-position-company relation extraction prototype

3. Political use case for the Polish language

3.1. Analyzing political tweets in Polish

This use case applies the tools described in [8] to perform a semantic tagging and analysis of tweets with political content in the Polish languages. This is an extension of TrendMiner to a new language in one of the domains included in the project. In the following sections, tools to gather tweets from Twitter are described along with an explanation on how available TrendMiner visualization capabilities have been reused.

3.2. Polish political Twitter account list

Before the beginning of the project there existed no common publicly available resource gathering Twitter accounts of Polish political players. In the scope of TrendMiner such list was created for the Polish Third Republic (years 1989-2014). The collection consists of members of upper and lower houses as well as Polish members of the European Parliament.

Initial version of the list

The initial version of the list combined the following available component resources:

– a list of political Twitter accounts by Mateusz Puszczyński (https://twitter.com/socjokracja/politycy/members)
– a list of Polish Twitter accounts (political section) by Klub Chwila (http://www.holdys.pl/polskitwitter/)

– TOP 50 Twitter political account list by wirtualnemedia.pl (http://www.wirtualnemedia.pl/artykul/top50-polskich-politykow-na-twitterze-palikot-sikorski-i-tusk-na-czele)
– TOP 50 Twitter political journalist accounts by wirtualnemedia.pl (http://www.wirtualnemedia.pl/artykul/kuzniar-piasecki-i-sekielski-liderami-dziennikarskiego-twittera-najaktywniejsi-janecki-warzecha-i-pereira-top50)
– data offered by ePaństwo Foundation (http://epf.org.pl/app/webroot/api/dane/).

Extending the list with automatically retrieved accounts

The initial version of the list was supplemented with accounts automatically retrieved from Twitter and then manually verified by experts:

– based on the entries from the Polish Political Ontology all Twitter accounts compatible with the first name-surname pattern were collected (20 accounts on average, due to many fakes)
– for every candidate account every *friend* and *follower* of the account was retrieved (in Twitter terminology: when A is observing B, B is a friend of A and A is a follower of B)
– the list was filtered using the methodology of the self-confirming structure, i.e. by tracking only accounts followed by verified accounts and being followers and friends of verified accounts of the largest Polish political parties.
– finally the remaining candidate accounts were manually verified and all inactive accounts (by people who did not publish any tweets retrievable by Twitter API) were removed.

Currently the list contains 766 accounts.

3.3. Tweet collection tools

This report presents how Polish political tweets are collected for the TrendMiner project. As the main purpose of the project is to focus on political tweets, we have chosen to collect only tweets from a selected set of users, which are likely to tweet about politics. Our application monitors their accounts and fetches their tweets on a regular basis.

3.3.1. Collecting scheme

Starting point for implementing our tweet collecting application was a set of Python scripts, developed within TrendMiner: `twitter-collection-utils` project[14]. These scripts allow interacting with the Twitter REST API[15] to fetch Twitter data. Our version of the tweet collecting tool is called Tweet Collector.

Via Twitter REST API, the tools monitor on a regular basis a selected set of users for new activity, downloading their new tweets and storing them in a database. Using this data, further processing may be done, but it is outside of the scope of this report. Here, we present only information what we collect and in which way.

[14] `https://github.com/danielpreotiuc/twitter-collection-utils`
[15] `https://dev.twitter.com/docs/api/1.1`

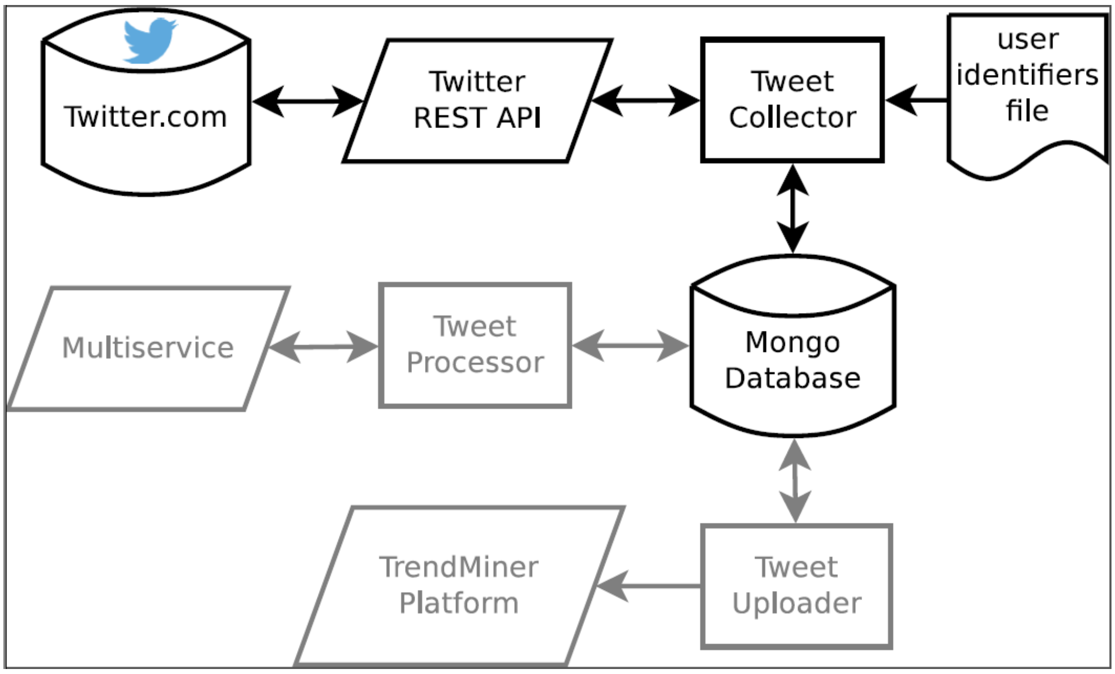


Figure 21. Tweet collection scheme

3.3.1.1. Twitter account and rate limiting

To use Twitter API, one needs to have a developer Twitter account, as well as to register an application to be identified with API requests. Such account may be easily obtained from http://dev.twitter.com, where we get a set of credentials for accessing the API.

Unfortunately, free of charge requests are limited – a number of requests of each type have a specific limit for any 15 minute window. When we try to fetch more data than we are allowed to, we simply have to wait until the time window passes. In our collecting script, we use only two API methods:

- GET statuses/user_timeline – for fetching tweets of a particular user,
- GET application/rate_limit_status – for checking the number of requests within a fixed time interval.

The first method currently has a limit of 300 requests per 15 minutes, the second one – 180 requests per 15 minutes.

3.3.1.2. Collecting script

Our version of collecting script is developed in Python and publicly available at https://github.com/mkopec87/trendminer-pl/ repository in python-twitter-collector project. The project contains several helper scripts, but the one responsible for collecting tweets on a regular basis is monitor_users.py. It may be executed as follows:

```
python monitor_users.py user_list twittertokens db_details
```

where:

- `user_list` is a file with tab-separated user list, where first column contains unique Twitter user (numerical) id, other columns are not obligatory,
- `twittertokens` is a file with our credentials to use the API,

- `db_details` is a string, specifying a connection to database to store the tweets, in form of database-name:collection-name.

Compared to the original TrendMiner twitter-collection-utils project, the main difference is the usage of Mongo database for data storage, instead of plain text files.

3.3.1.3. API usage

As stated in the previous section, for fetching tweets of a particular user, we use `GET statuses/ user_timeline` API method. It allows fetching up to 3,200 of a user's most recent tweets, but no more than 200 per single request. This, combined with the rate limiting information, creates an upper bound on tweet downloading speed – we cannot obtain more than 240,000 tweets per hour, even if we would have enough users producing such amount of tweets.

API method may use user's `screen_name` or id to identify request for his/her tweets. We use id_str, which is a string version of id. Use of integer version of id is discouraged[16], as there are problems with very long numbers. Usage of `screen_name` would also be worse, as it may be changed by the user, while `id_str` cannot.

We control which tweets should be fetched using `since_id` and `max_id`, which control time span we are interested in. We start with `since_id` equal to the last downloaded tweet's id of a given user in a previous run of the program (or 1 if we haven't downloaded any yet). This id is simply taken from the stored tweets in `target_dir` folder. As tweets are downloaded from the most recent ones, if we download a maximum number (200), we need to use `max_id` to further fetch tweets older than the most recent 200 tweets, but newer than our most recent tweet from the previous run. Such procedure is followed until we fetch all new tweets of a user, then we move on to the next user from our list.

If any request is rejected, the application needs to check rate limiting using `GET application/ rate_limit_status` and wait required number of seconds. If requests for any user give more than 4 errors in a row, we omit that user in given run of the program.

3.3.1.4. Discarded tweets

Not all tweets obtained by the API are well-formed. We have decided to ignore tweets, which lack any of the following attributes:

- id_str value,
- created_at value,
- user value,
    - user[id_str] value,
    - user[screen_name] value.

[16] `https://dev.twitter.com/docs/twitter-ids-json-and-snowflake`

3.3.1.5. Database details

To allow easy access to tweets, we store them in a MongoDB database[17]. As Mongo stores data as documents, we simply save JSON objects obtained from the API into the database. Tweet collection has a unique index on id_str field created by:

```
   db.tweets.ensureIndex( { "id_str": 1 }, { unique: true } )
```

The database is located at glass.ipipan.waw.pl server, and its name is trendminer, tweets are stored in tweets collection.

To allow easier data filtering, we add to the original JSON a `created_at_mongo` field, which is the created_`at` field converted to date format used by MongoDB. We also put a chronological sorting index on that field by:

```
        db.tweets.ensureIndex( { "created_at_mongo": -1 } )
```

Two other indices are useful in further processing:

```
db.tweets.ensureIndex( { "analysis.language.value": 1 } )
db.tweets.ensureIndex( { "user.id_str": 1 } )
```

Because the usage of integer id fields is discouraged, we remove both the id and `user[id]` fields prior to saving JSON to the database.

The database is not publicly available due to Twitter Terms of Service.

3.3.2. Current setting

Our application is scheduled to run every hour on a server using CRON job:

```
0 * * * * PYTHONPATH=/home/pzi/local/lib/python2.7/site-
packages ; export PYTHONPATH ; cd
/mnt/d1/trendminer/twitter/twitter-collection-utils-master ;
python monitor-users.py user-file-common 0 common >
logs/common.log
```

If a collecting application instance is already running when we want to run it again, we skip that run.

3.3.2.1. Followed accounts

Our list of users to follow is the Polish political Twitter account list. It currently contains over 700 user ids. It is stored in the SVN inside twitter-collector project. As the list was changed during time, it may have impact on the number of collected tweets, yet from May to July 2014 the list was not extended, therefore volume of new tweets from these months is equal to the number of tweets produced in that period.

3.3.2.2. Data volume

In August 2014, database contained over 1,800,000 tweets. Almost 100,000 of them were created in July 2014, about 135,000 in June 2014; therefore we may currently expect a volume of about 100,000-150,000 tweets/month. Figure 22 presents tweet counts from each month from January 2011 to July 2014.

[17] `http://www.mongodb.org/`

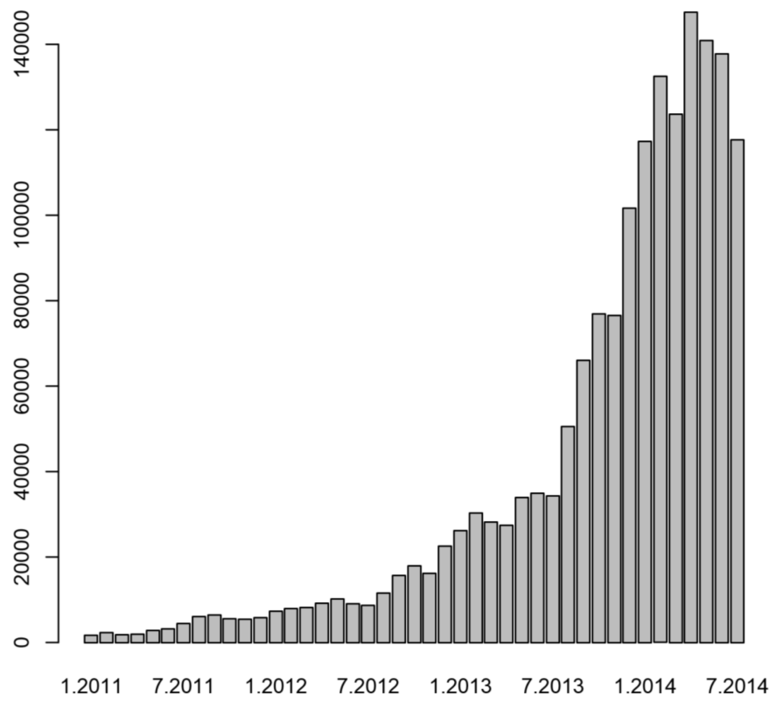


Figure 22. Tweet counts by month

3.4. Visualisation of Polish tweets

This section describes the process of visualisation of Polish political tweets as performed by the TrendMiner UI frontend.

3.4.1. Architecture

Polish political tweets are collected into a database on a regular basis. They are also processed via an NLP pipeline and results of that NLP analysis are also stored in the same database. Both, tweet collection and NLP processing are described in detail in separate sections. Here we assume tweets are already collected and enriched with the results of NLP analysis.

Visualization of tweets is performed in a TrendMiner UI service. The process of uploading NLP-processed tweets to that service is presented in Figure 23 and described in detail in the following sections.

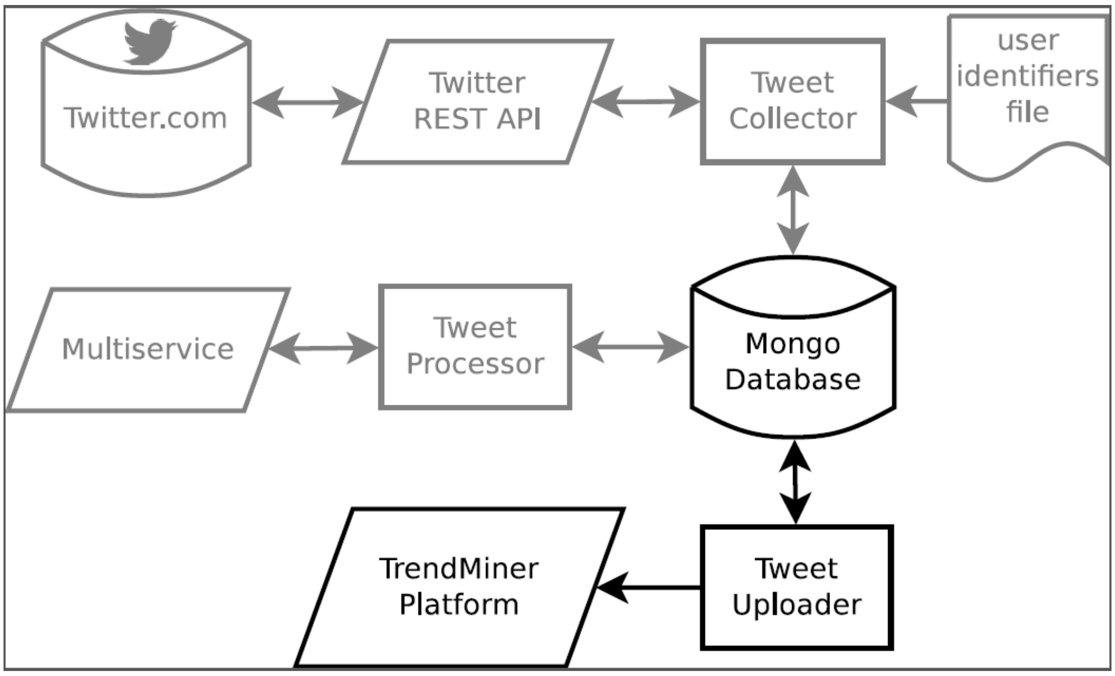


Figure 23. Visualization scheme

3.4.1.1. Visualisation deployment server

The server which performs visualization of Polish tweets is located at[18]:

```
    http://ipipan-uc.ontotext.com/TrendMiner/tracks.html
```

Our tool sends data directly to the data warehouse of that server. Data may also be sent manually, using cURL with the command:

```
curl -X POST -H "Content -Type: text/turtle" --data
"@RDF_DATA_FILE" --user ipipan:ipipan1420 http://ipipan-
uc.ontotext.com/tm-data-service-v3/rdf
```

Where RDF_DATA_FILE is a file in RDF output format of Polish tweets, described in detail in the next section.

For development purposes, data at the server may be deleted using the command:

```
curl -X DELETE --user ipipan:ipipan1420 http://ipipan-
uc.ontotext.com/tm-data-service-v3/admin/resetstore
```

3.4.1.2. RDF output format

Tweets are sent to the TrendMiner platform server in RDF format. One upload request may contain several tweets. The request starts with namespaces definition lines:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc-terms: <http://purl.org/dc/terms/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix geo-pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
```

Tweets are represented in RDF (here using the Turtle syntax). Each tweet is started by a blank line, followed by a line similar to this one:

```
<http://trendminer.eu/resources#tweet_3273675 >
```

[18] Credentials can be obtained upon request. Please contact Mateusz Kopeć, MSc. mateusz.kopec@ipipan.waw.pl

where 3273675 is the identifier of given tweet as given by the tweet API. After that we have a number of lines containing attributes of that tweet. Each line is similar to the one below:

```
dc-terms:creator "waldydzikowski" ;
```

Table 1 below describes all tweet attributes.

Attribute name	**Content**	**Example value**
dc:identifier	tweet identifier	"3273675"
dc-terms:creator	author’s screen name	"waldydzikowski"
dc:description	original tweet content	"@KrzysztofLisek Dziekuje, w koncu sie zdecydowalem;) #wojna"
dc:date	timestamp	"2013-04-25T10:24:39"^^xsd:dateTime
dc:hashtag	hashtags	#wojna
dc:source	source	"twitter.com"
dc:language	language	"pl"
dc:sentiment	sentiment of the tweet	"1.0"^^xsd:float
dc:location	location mentioned in tweet	"Warsaw"
dc:location_uri	URI of a location mentioned in tweet	<http://dbpedia.org/resource/Warsaw>
dc-terms:references	URI of an entity mentioned in tweet	<http://dbpedia.org/resource/SoftBank>
dc-terms:ref_lab	label for an entity mentioned in tweet	"Softbank"
dc-terms:subject	topic of tweet	"polityka"
<http://trendminer.eu/hasTokens>	space separated list of tokens which might be normalized, lemmatized or whatever is necessary to feed the clustering tool properly.	KrzysztofLisek dziękować koniec zdecydować ;) wojna

Table 1: Attributes of a single tweet

The first five attribute values come directly from the Twitter API (`tweet_json` is the API tweet object) and are not a result of any NLP analysis:

1. dc:identifier – it’s a copy of tweet_json[id_str].
2. dc-terms:creator – it’s a copy of tweet_json[user][screen_name].
3. dc:description – it’s a copy of tweet_json[text].
4. dc:date – it’s a copy of tweet_json[created_at], converted to another date format (notice the lack of timezone information).
5. dc:hashtag – it’s an information easily extracted from tweet_json[entities][hashtags].

The next two attributes are constants in Polish NLP processing:

1. dc:source – it is always "twitter.com".
2. dc:language – it is always "pl". We do not process and visualise tweets for which the detected language is other than Polish. They are simply discarded.

The following six attributes are the result of Polish NLP tweet processing, described in deliverable 10.1 [8]:

1. `dc:sentiment` – represents sentiment of a tweet in a numeric value from interval [−1,1], where -1 stands for negative sentiment and 1 stands for positive sentiment.
2. `dc:location` – is location mentioned in tweet, represented as Polish location name.
3. `dc:location_uri` – URI to the location in dc:location attribute.
4. `dc-terms:references` – URI to an entity mentioned in a tweet.
5. `dc-terms:ref_lab` – Polish label for an entity mentioned in a tweet.
6. `dc-terms:subject` – subject of a tweet in Polish, may be a multi-word text.

All attributes except dc:sentiment from these six can have multiple values, for example several locations may be mentioned in a tweet, as well as several entities or subjects. In such case, we may output multiple values for such attribute, separating them by commas.

Obligatory attributes

A tweet won't be accepted or presented by TrendMiner platform, if it misses any of the following attributes:

- dc:identifier,
- dc:date,
- dc:language,
- dc-terms:creator,
- dc:description.

Sending location GPS coordinates

TrendMiner UI visualises locations of tweets on a map. For such purpose, we need to send GPS coordinates for the locations, which we resolved as tweet attributes. We send location data along with tweet data, starting location data with location URI, and then specifying following attributes:

- `<http://www.w3.org/2000/01/rdf-schema#label>` – label for location,
- `geo-pos:lat` – float representing latitude,
- `geo-pos:long` – float representing longitude.

An example of a file sent to the server is shown below:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc-terms: <http://purl.org/dc/terms/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix geo-pos: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
<http://trendminer.eu/resources#tweet_327367546749718528 >
dc:identifier "327367546749718528" ;
```

```
dc-terms:creator "waldydzikowski" ;
dc:description   "@KrzysztofLisek  Dziekuje  ,  w  koncu  sie
zdecydowalem;) Pozdrawiam" ;
dc:language "pl" ;
dc:sentiment "0.7"^^xsd:float ;
dc:source "twitter.com" ;
dc:location "Warszawa" ;
dc:location_uri <http://dbpedia.org/resource/Warsaw> ;
dc:date "2014-04-25T10:24:39"^^xsd:dateTime ;
dc-terms:references <http://dbpedia.org/resource/SoftBank > ;
dc-terms:ref_lab  "Softbank"  ;  dc-terms:subject  "polityka"  ,
"wybory"  ;  <http://trendminer.eu/hasTokens  >  "@KrzysztofLisek
Dziekuje , ← - w koncu sie zdecydowalem ;) Pozdrawiam".
<http://trendminer.eu/resources#tweet_121235135415313213 > ← -
dc:identifier "121235135415313213" ;
dc-terms:creator "Polityka300" ;
dc:description  "Jak  wazne  jest  nowe  stanowisko  Tuska?  W  2014
r. do szefa RE Putin dzwonil... raz http://t.co/G8xS3L8K2q" ;
dc:language "pl" ;
dc:sentiment "-0.1"^^xsd:float ;
dc:source "twitter.com" ;
dc:location "Warszawa" ;
dc:location_uri <http://dbpedia.org/resource/Warsaw> ; dc:date
"2014-04-25T13:24:39"^^xsd:dateTime ;
dc-terms:references
<http://dbpedia.org/resource/EuropeanCouncil>                ,
<http://dbpedia.org/resource/Tusk>;
dc-terms:ref_lab "Rada Europejska" , "Tusk" ;
dc-terms:subject  "wybory"  ;  <http://trendminer.eu/hasTokens  >
"jak  wazny  byc  nowy  stanowisko  Tusk  ?  w  2014  R  .  do  szef  RE
Putin dzwonic . . . ← - raz http://t.co/G8xS3L8K2q" .
<http://dbpedia.org/resource/Warsaw>
<http://www.w3.org/2000/01/rdf-schema#label> "Warszawa" ;
geo-pos:lat "52.0"^^xsd:float ;
geo-pos:long "21.0"^^xsd:float .
```

3.4.3. Tweet uploading tool

The tweet uploading tool is a Java application and stored in `svn://chopin.ipipan.waw.pl/trendminer/tools/ptp/ptp-upload project`. It is supposed to be run on a regular basis to find all tweets processed with NLP chain of specified version and not already uploaded, convert them to format presented before and upload to the TrendMiner platform server.

3.4.3.1. Assumptions about data input

Each tweet, which we want to upload to visualisation server, should be present in Mongo database with following attributes:

- `id_str` – tweet id,
- `text` – original tweet content,
- `created_at` – tweet creation time in Twitter API format,
- `analysis field`, containing subfields:
    - o `version_id` – version id of NLP process used for analysis,
    - o `language.value` – detected language,

  - `uploaded_ontotext` – flag marking if the text was already uploaded to the TrendMiner platform,
  - `sentiment.value` – sentiment value,
  - `topics.values` – list of topics detected in tweet,
- `references.values` – list of entities mentioned in tweet, each being a map with:
  - `label` – label for an entity,
  - `uri` – URI for that entity,
- `locations.values` – list of locations mentioned in tweet, each being a map with:
  - `label` – label for that location,
  - `uri` – URI for that location,
  - `longitude` – longitude for that location,
  - latitude – latitude for that location,
- `tokens` – list of tokens parsed in tweet content, each being a map with:
  - base – lemmatized token,
  - user field, containing subfields:
    - `id_str` – user id,
    - `screen_name` – user screen name,
- `entities.hashtags` field, containing:
  - `text` – hashtag value, without the '#' sign.

3.4.3.2. Upload process

The Mongo database may contain tweets processed with various Tweet Processor versions, which is marked for each tweet in analysis.version_id attribute. For running the tweet uploading tool, we need to specify which version is the one we want to upload for visualization. The Tweet Uploader works as follows:

1. Selects tweets in the database which have:
   - particular analysis.version_id,
   - analysis.language.value equal to "pl",
   - analysis.uploaded_ontotext not equal to true.
2. Prepares tweets by converting them to the RDF format accepted by the visualisation server.
3. Sends RDF data to the TrendMiner platform server.
4. Updates in the database the field analysis.uploaded_ontotext to true for each tweet successfully sent.

3.4.3.3. Visualisation example

Below, in Figure 24 and Figure 25, we display as examples of the visualisation of the analysis generated by the political use case applied to Polish streaming data results related to a recent development in the Polish political life.
After it was announced (on the 30th August) that the Polish Prime Minister Donald Tusk would be the next President of the European Council, several names were mentioned as potential followers for the position of Prime Minister: Tomasz Siemoniak, Elżbieta Bieńkowska, Grzegorz Schetyna, Paweł Graś and Ewa Kopacz, who was eventually elected (on September 22). Bieńkowska was leading in the first period since most political commentators regarded her as the best candidate; then Schetyna's name hit the headlines when he came back from political exile (he became

a big opponent to Tusk once, so Tusk minimized his role in the party to protect his position). We can also see how Kopacz's position was constantly growing until the moment when it was decided that she would be the Primer Minister. Then Schetyna is coming back since he was designated as the Minister of Foreign Affairs.

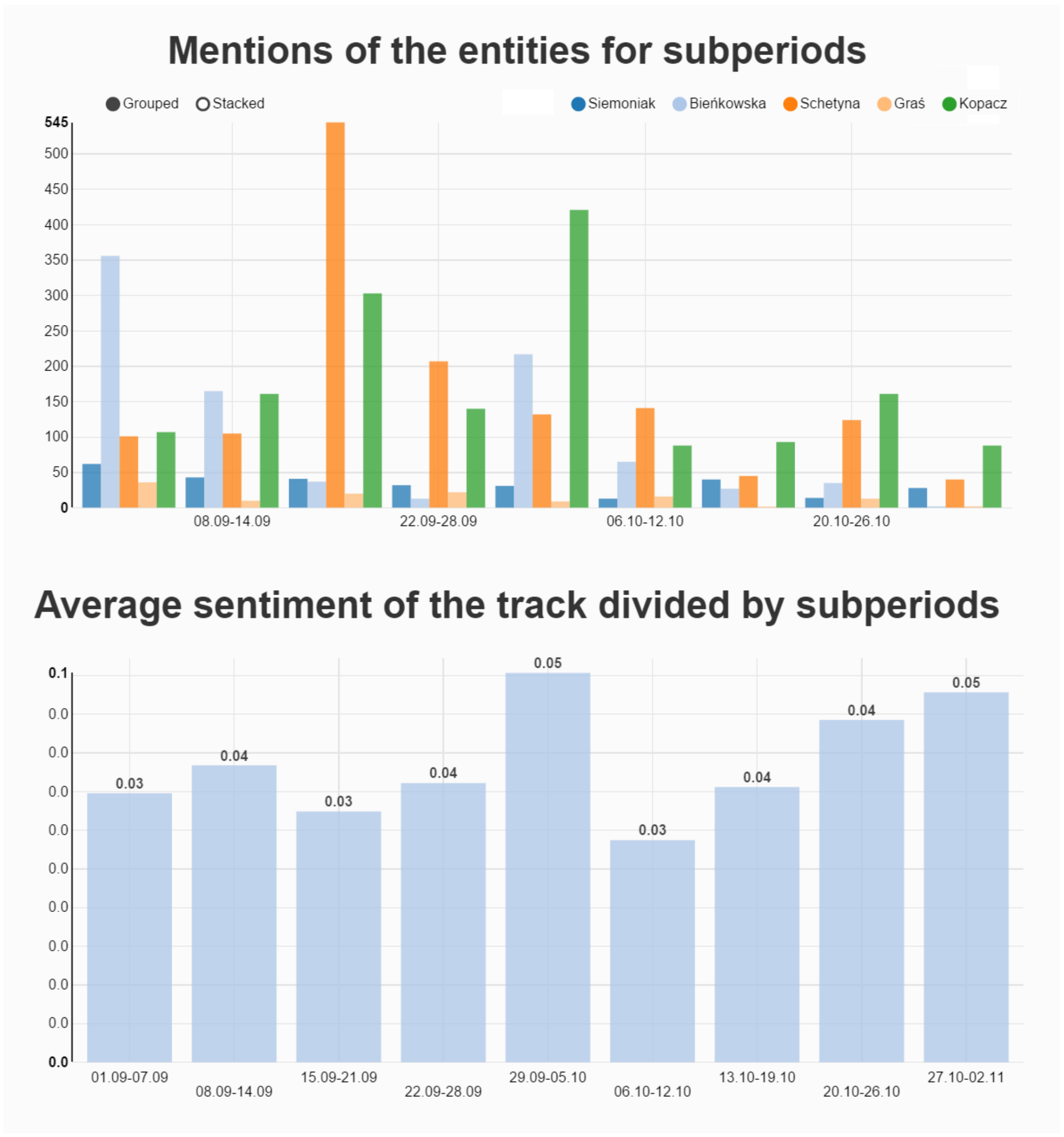


Figure 24. Mentions of Polish politicians named in relation with the nomination of the new Primer Minister (elected on September 22nd 2014)

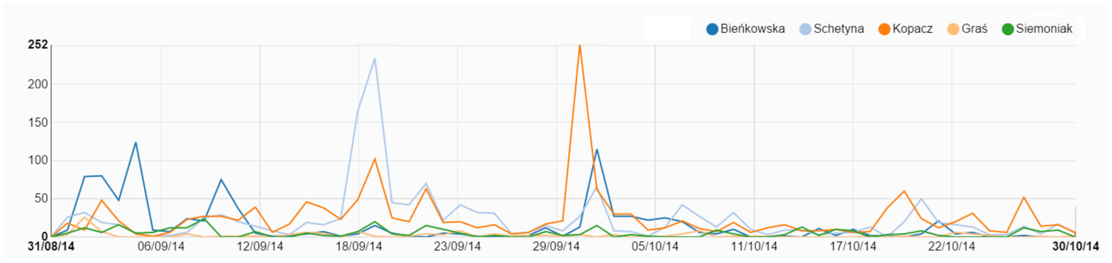


Figure 25. Time line selector for the user

4. Hungarian Political Use Case

As the scenario of the extension of the political use case to Hungarian social media data has many similarities with the Polish political use case described in Section 3 above, and as the Hungarian NLP adaptation efforts are described in details in Deliverable 10.1 [8], we concentrate in this section in mentioning the particular aspects of this extension of the political use case.

This extension of the political use case of Trendminer implements the collection and processing of Hungarian public Facebook comments from Hungarian politicians' and political organizations' pages. The comments undergo basic NLP processing, named entity identification, sentiment analysis and the analysis of unique sociopsychological indicators including *primordial vs. conceptual thinking, agency and communion, optimism/pessimism and individualism/collectivism* (see [8] for more details).

The data sources comprise 1341 different Facebook pages of Hungarian political actors involved in the 2010 Hungarian parliamental elections, 2014 Hungarian parliamental elections and the 2014 European Parliament elections in Hungary. Section 5.1 in [8] describes in details how these pages were selected. These sources were also linked to entries in the Hungarian political ontology (described in detail in section 1.3 in [10]).

The Facebook Graph API was used to collect public posts and their public comments from these sources on a regular basis in JSON format. Since Facebook as a new data source was introduced to TrendMiner in this use case, a new data collecting solution needed to be implemented, which runs independent of TrendMiner's social media (Twitter) message harvesting architecture (see section 5.1 in [8] for more details). The comments and their metadata are saved in a MySQL database to enable efficient storage and arbitrary querying (data model and details are given in Deliverable 10.1 [8]).

Each downloaded comment is processed by a pipeline that performs basic NLP tasks (tokenization, morphological analysis, part-of-speech tagging, lemmatization), named entity recognition and content analysis (sentiment + 4 other psychological/sociopsychological indicators). These processes use existing Hungarian NLP tools that were adapted to the special language used in the social media messages, in combination with lexicons and grammars developed for the NooJ[19] language analysis framework. Again, these tools operate independently of the TrendMiner processing architecture on servers hosted by RILMTA. For more details on the NLP tools and their adaptation to Facebook comments language see section 5.2 in [8]. For more information on the sentiment analysis and other sociopsychological analysis methods, see section 5.3 in [8].

The results of the analys and annotation tools are processed to extract entity mentions and various indicators (sentiment score, regressive imagery dictionary primary/secondary annotation counts, agency score, communion score, individualism score, optimism score) for each comment which are then stored in the database (see section 5.4 in [8] for more details). Finally, each comment and associated sentiment

[19] http://nooj4nlp.net/pages/nooj.html

score, mentioned entities with their ontology links and other metadata is converted to RDF representation and uploaded to the TrendMiner use case visualization service using the provided RESTful API endpoints. The details of this process are described in section 1.3 of [11].

5. Conclusions

In this deliverable we have presented a series of webservices developed in the last year of the project as extensions to the original description of work of TrendMiner The web services are supporting interaction of the prototypes developed by the new partners (IPIPAN, RILMTA, UC3M and DAEDALUS), who joined the project consortiumin the 3rd and last year, with the users.

We described in some details the new prototypes, which are dealing with a new domain (eHealth) and with the extensions of already integrated use cases (the financial use case in WP6 and the political use case in WP7). We have show the different levels of integration of the new resulting data and of the new components in the TrendMiner platform, whose final state is described in the Deliverable D5.3.2 (http://www.trendminer-project.eu/images/d5.3.2_final.pdf)

6. Bibliography and references

[1] Elasticsearch, 2014. *Elasticsearch.* [Online]
Available at: http://www.elasticsearch.org/

[2] Enfemenino.com, 2014. *enfemenino.com.* [Online]
Available at: http://www.enfemenino.com/

[3] Forumclinic, 2013. *Forumclinic.* [Online]
Available at: http://www.forumclinic.org/

[4] Highcharts, 2014. *Highcharts.* [Online]
Available at: http://www.highcharts.com/

[5] Saluspot, 2014. *Saluspot.* [Online]
Available at: https://www.saluspot.com/

[6] Sheffield, T. U. o., 2014. *GATE- General Architecture for Text Engineering.* [Online]
Available at: https://gate.ac.uk/

[7] Textalytics, 2014. *Core Textalytics APIs.* [Online]
Available at: https://textalytics.com/api-core-language-analysis

[8] D10.1. Newly generated domain-specific language data and tools, 2014.
Available at http://www.trendminer-project.eu/images/d10.1.pdf

[9] Twitter, 2014. *Twitter Developers.* [Online]
Available at: https://dev.twitter.com/overview/documentation

[10] TrendMiner, Integration of all lexical and terminological data in the TRENDMINER platform, D2.4 Public TrendMiner Deliverable, WorkPackage 2, 2014.

[11] TrendMiner, Integration of All Annotations Generated by the New NLP Tools, D9.1 Public TrendMiner Deliverable, WorkPackage 9, 2014.