FP7-ICT Strategic Targeted Research Project TrendMiner (No. 287863)
Large-scale, Cross-lingual Trend Mining and Summarisation of Real-time Media Streams

D10.1 Newly generated domain-specific language data and tools

José L. Martínez Fernández (Editor, DAEDALUS), Paloma Martínez (UC3M),
Maciej Ogrodniczuk (IPIPAN), Márton Miháltz (RILMTA)

Abstract

FP7-ICT Strategic Targeted Research Project TrendMiner (No. 287863)
D10.1 Newly generated domain-specific language data and tools (WP10)

This report describes the resources (knowledge bases, lexicons and corpora) and NLP tools integrated in the systems developed by the new TrendMiner partners DAEDALUS, IPIPAN, RILMTA and UC3M, across Health, Financial and Political use cases

Keyword list: NLP tools, lexicons, corpora and evaluation, use cases

Nature: Report
Dissemination: PU
Contractual date of delivery: 31.10.2014
Actual date of delivery: 31.10.2014
Reviewed By: DFKI
Web links: http://www.trendminer-project.eu; 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 Rvolution
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

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

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

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

Institute of Computer Science,
Polish Academy of Sciences
Jana Kazimierza 5
01-248 Warszawa
Contact person: Maciej Ogrodniczuk
E-mail: maciej.ogrodniczuk@ipipan.waw.pl

Research Institute for Linguistics,
Hungarian Academy of Sciences
Research Group for Language Technology
Benczúr u. 33, Budapest 1068 Hungary
Contact person: Tamás Váradi
E-mail: varadi.tamas@nytud.mta.hu

Executive Summary

This report describes the resources and linguistic processors developed for the new or extended use cases introduced by the new TrendMiner partners (IPIPAN, RILMTA, DAEDALUS and UC3M). The domains covered are health, finance, psychology and politics. For the financial and political domains, links and collaborations have been established to work done in the context of WP6 and WP7.

Concerning the health use case, a web service has been developed that shows how drugs, diseases and adverse effects of drugs are mentioned in social media, including blogs and social networks in Spanish. This deliverable describes the resources and NLP tools that have been implemented and integrated for this demonstrator, as well as the evaluation that has been performed.

Concerning the extended financial use case, the resources and linguistic processors developed and used for building a demonstrator are described. This demonstrator (a) identifies job relations between people and companies in news articles in Spanish and (b) analyses companies reputation in social networks, analysing messages written in Spanish. The results of this work are fully integrated in the TrendMiner platform (WP5).

Finally, the extended political use case includes two application domains: sentiment analysis of Polish political tweets and Hungarian socio-psychological analysis of social networks. For both of them we describe the adaptation work done in order to have the original linguistic processing tools running on user generated content texts.

For every use case, adaptation and extension of the TrendMiner ontologies (described in D2.1.2: Knowledge and Provenance Modelling and Stream Modelling) Multilingual resources and evaluation of knowledge modelling) developed in WP2 have been performed. Most of this work is described in D2.4: Integration of lexical and terminological data in the TrendMiner platform.

Contents

List of Acronyms

ADR	Adverse Drug Reaction
API	Application Programming Interface
ATC	Anatomical Therapeutic Chemical system
CRF	Conditional Random Field
CUI	Concept Unique Identifier
Fb	Facebook
GATE	General Architecture for Text Engineering
IAA	Inter Annotator Agreement
JAPE	Java Annotation Patterns Engine
LLT	Lowest Level Term
MP	Members of Parliament
NER	Named Entity Recognition
NLP	Natural Language Processing
NPCA	Narrative Psychological Content Analysis
PT	Preferred Term
RDF	Resource Description Framework
RID	Regressive Imagery Dictionary
SM	Social Media
SNP	Scientific Narrative Psychology
SO-PMI	Semantic Orientation-Pointwise Mutual Information
UMLS	Unified Medical Language System

List of Tables

List of Figures

1 Introduction

This report describes the resources and NLP tools developed for the new or extended use cases introduced by the new TrendMiner partners (IPIPAN, RILMTA, DAEDALUS and UC3M). The domains covered are health, finance, psychology and politics.

We present in this document the adapted language resources and tools deployed by the new partners in order to be able to extend the coverage of the TrendMiner platform. Integration work is described in D5.5 Deployment of web services for new use cases. Integration of lexical and terminological resources of the new partners in the TrendMiner platform is described in D2.4: Integration of lexical and terminological data in the TrendMiner platform.

2. Health Use Case

Current definitions of Social Media include several sources of user generated data, from Twitter to specialized blogs through Facebook. Users of these Web 2.0 applications share information about any subject, including issues related with their health condition. The number of people with Internet access seeking for health information through the net ranges from 70 to 75% in the U.S. Besides, 42% of them used social media to get information about health issues. Patients communicate each other about their feelings about a health problem, the way their bodies react to a given drug, how they mix different drugs to fight against some disease they have and many other issues related to their health situation. Nowadays, every company should be aware about opinions and mentions to them given by their customers in any of these Social Media; and health and pharma companies are not an exception to this. As an example of the importance of Social Media interactions in the health sector, according to a study developed by Price Waterhouse Coopers, 45% of consumers said information from Social Media would affect their decisions to seek a second opinion[1].

[1] PriceWaterhouseCoopers, Social media “likes” healthcare, http://www.pwc.com/us/en/health-industries/publications/health-care-social-media.jhtml

In this context, the aim is to apply text analytics processes to extract information from these real time Social Media streams relevant for the healthcare sector. The information to be extracted concerns drugs, adverse drugs reactions (ADR), drug indications and diseases mentions that could be interesting to medical centres, drug agencies, etc.. It also covers trend evolution of those mentions detected in patients' conversations in social media.

Text analytics processes to be applied for this purpose cannot be generic, but need to be adapted to the health domain. This requires specific dictionaries and ontologies covering drug, diseases, body part names and so on in order to support for example named-entity recognition, sentiment expressions used in relation with health topics, or colloquial expressions used for drugs, diseases and other entities. Concerning ADRs, it is well-known that they build an important health problem. Indeed, ADRs are the 4th cause of death in hospitalized patients (Wester et al., 2008). Thus, the field of pharmacovigilance has received a great deal of attention due to the high and growing incidence of drug safety incidents (Bond and Raehl, 2006) as well as to their high associated costs (van Der Hooft et al., 2006).

The objective in this new use case in TrendMiner was therefore to build a prototype for monitoring mentions of drugs, diseases, adverse effects, indications and other health related issues (see deliverable D5.5 for the prototype description). Next sections describe the implemented and adapted dictionaries and linguistic processors that have been integrated in this prototype.

2.1. Data sources

We use two sources of user generated data related to health issues: Twitter and Saluspot[2]. Comments and posts from these sources are annotated with the medical pipeline described in Section 1.5.

From Twitter, only tweets corresponding to certain filters are collected. Concretely, tweets that contain specific keywords like drug or disease names and are written in Spanish (the prototype currently collects tweets containing antidepressants and related drugs). As for October 14, 2014 a collection of 248,232 tweets was correspondingly indexed.

[2] https://www.saluspot.com/

The second data source is Saluspot, a Spanish website that allows its users to address free of charge and anonymously their doubts and information needs about health, lifestyle and drugs to thousands of registered doctors. Once a question is posted any of the registered, accredited doctors can answer and even multiple answers are possible. As for October 14, 2014 a collection of 41,985 comments was indexed.

2.2 Dictionaries

The health monitoring system uses a total of four dictionaries developed at UC3M; each of them intended to detect a different type of entity. In the first place, there is a dictionary which includes all the drugs included in the CIMA[3] resource, called drugsATC. Moreover, there is a dictionary including the totality of the adverse effects described in the MedDRA[4] resource, which receives the name of adrsMedDRA. These two dictionaries are generated querying the SpanishDrugEffect Database (see deliverable D2.4) with integrates CIMA, and MedDRA. Furthermore, the third dictionary contains the diseases obtained from the UMLS[5] resource, including Snomed-CT and MedDRA. Its name is diseasesUMLS. Finally, the dictionary called drugsGaz contains drug-related terms from different resources (Vademecum, MedlinePlus or Wikipedia). The primary sources for our dictionaries, CIMA, MedDRA, UMLS and ATC, are described in deliverable D2.4

2.2.1 drugsATC

Due to the fact that the ATC system does not include brand names, we associate active substances to their brand names in our dictionaries. The ATC code of each drug can be found in an extra field of the dictionary, adding thus value to each entry. In CIMA, each of the 16418 drugs is given an ATC code. This means that, despite the fact that the last level of the ATC code corresponds to an active ingredient in the drugsATC dictionary we still needed to establish correspondences among drugs (brand names) and the active substances they contain. In order to make this clear, an example is shown at Table 1. The aliases are those active substances that can be found

[3] http://www.aemps.gob.es/cima/
[4] http://www.meddra.org/
[5] http://www.nlm.nih.gov/research/umls/

in each of the brand drugs. See the description of the SpanishDrugEffect Database in Deliverable 2.4 to understand how drugs, ATC and active substances are related to.

entry	Drug	Aliases	Morphological tags	Semantic tags	ATC code
1163	eloine	Drospirerona/ etinilestradiol	NPUU-N- eloine	Sementity/class= instance@type= Top>Drug	SemId_list/ATC= G03AA12

Table 1: Examble of a drugsATC dictionary entry

We can see that *eloine* is the canonical expression of a drug whose dictionary id is 1163. This drug's composition includes two active substances: *drospirenona* and *etinilestradiol*. The ATC code associated with the drug *eloine* is G03AA12 as can be seen in the example. In the SpanishDrugEffect Database the ATC codes are also related to the active substances but this information is not included in the *drugATC* dictionary. In this example the ATC code of *etinilestradiol* is G03CA01 and the ATC code of *drospirenona* is G03AA12.

Thanks to the information that the ATC provides (therapeutic and chemical characteristics of the drug), the system is able to relate them to each other and is able to categorize them by active substance, chemical group or pharmacological group. This is possible due to the classification's hierarchy. ATC codes are divided in five levels. For example, the ATC code G03AA12 is divided into: Anatomical main group (G), therapeutic main group (03), therapeutic/pharmacological subgroup (A), chemical/therapeutic/pharmacological subgroup (A) and the chemical substance (12), forming the final G03AA12.

Every drug included in CIMA can be found in the tables Drug and DrugSynset (see SpanishDrugEffect Database in Deliverable 2.4). The first table contains the active substances (2,226), and the second one the brand drugs (6,352). Brand drug names were simplified before being included in the dictionary because many of them have very long names including dosages and/or administration and laboratory information. For example, *ESPIDIFEN 400 mg GRANULADO PARA SOLUCION ORAL SABOR ALBARICOQUE* is harmonized and represented as ESPIDIFEN. The DrugsATC dictionary has a total of 3,633 entries. This number corresponds to the distinct brand drugs included in CIMA.

2.2.2 adrsMedDRA

This dictionary has a total of 13,245 entries generated from the MedDRA resource. This number corresponds to the total number of Preferred Terms stored in the SpanishDrugEffect Database (see Deliverable 2.4). Every Preferred Term (PT) included in MedDRA can be found in the table Effect. Moreover, the table EffectSynset contains 48,504 terms (the 13,245 PT and 35,259 Lowest Level Terms (LLT)).

The Preferred Term of an effect is the one which is most probably used by a user, while the Lowest Level Terms give other ways of expressing this effect.

The canonical expressions of this dictionary are the 13,245 Preferred Terms (level 4 of MedDRA), whereas the aliases are those LLT (level 5) which are related to the PT throughout their MedDRA codes.

The extra information included in this dictionary is, in fact, the MedDRA code. MedDRA is a multilingual resource which uses this code to relate a term in all the different languages to which it is translated. As MedDRA is part of UMLS, this MedDRA code can be found assigned to a CUI (Concept Unique Identifier) which allows the system to find a term in different resources.

2.2.3 diseasesUMLS

This generated dictionary has a total of 42,547 entries. This number corresponds to the total number of Preferred Terms included in the UMLS which included either a term from Snomed CT or MedDRA.

UMLS is structured in several semantic categories (Substances, organisms, health care activity, etc.). Three of these categories (‘Diseases or syndrome’, ‘Mental or Behavioral Dysfunction’ and ‘Neoplastic process’) have been chosen in order to create a dictionary for diseases and symptoms.

From the set obtained for the resource, some of them were Preferred Terms and others Synonyms. The PTs were set as canonical expressions in the dictionary, and the Synonyms for these were considered as being the aliases.

As it happened with the two previously described dictionaries, this one also has extra information in every entry. In this case, this is the UMLS CUI, a code which relates a specific medical term (a disease in our resource), with a set of resources included in UMLS.

As a matter of fact, some terms are included in both the diseases dictionary and the adverse effect one. This is the case of *depresión* (*depression*). In the adrsMedDRA, it is related to its MedDRA code '10012378'. Thus, if checking this code in the UMLS resource, the CUI assigned to it is 'C0011570' (the code included in the diseasesUMLS dictionary).

2.2.4 drugsGaz

This dictionary has a total of 4,791 entries with drug-related terminology which was obtained by crawling different resources, like Vademecum[6] or MedlinePlus[7].

There are no aliases in this dictionary, as well as it has no extra information in the form of codes or any other data. By collecting information of other resources, more knowledge was obtained, and therefore the recall of the system could be increased.

2.3 Linguistics Processing: medical GATE pipeline

This section explains the different processing steps applied to the user comments and post extracted from social media in order to analyse Drugs, ADRs and Diseases. All this processing has been done with the help of the GATE NLP software (General Architecture for Text Engineering)

In order to process a document or a corpus of documents, GATE permits the generation of a set of processes ordered in a pipeline. This pipeline is based on a sequential execution of modules (see Figure 1) which are detailed in the next chapters.

Morphosyntactic Parser: This module uses the external service Textalytics[8], a commercial tool from DAEDALUS for Named Entity Recognition (NER), which follows a dictionary-based approach to morpho-syntactically analyse the input text. The output is a GATE annotation set composed by all the annotations generated by Textalytics service.

Topics Analyzer: This module also uses the external service Textalytics and generates an annotation set that contains all the entities present in the text. The entities that it recognizes are obtained from the four self-defined dictionaries described in the former section in more details:

[6] http://www.vademecum.es/

[7] http://www.nlm.nih.gov/medlineplus/spanish/

[8] https://textalytics.com/

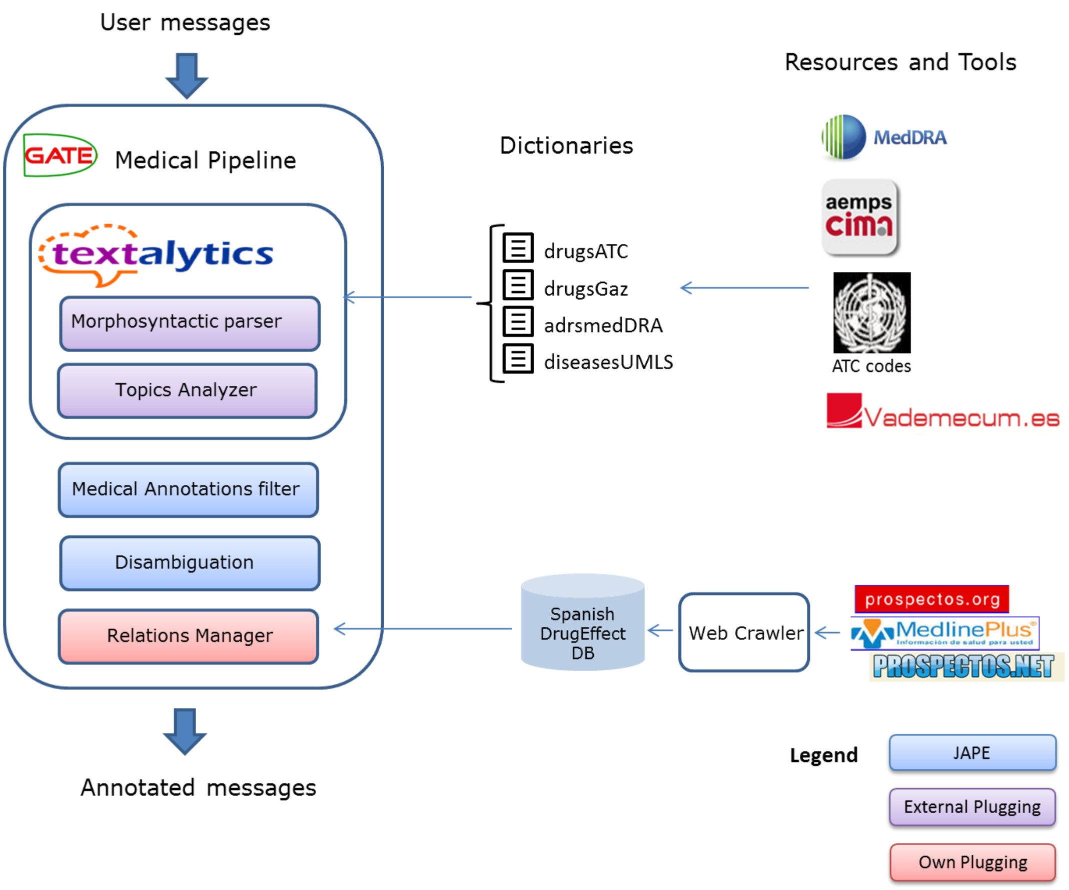


Figure 1: Medical Pipeline architecture implemented in GATE

- DrugsATC: a set of drugs obtained from CIMA associated with their active principles and their ATC codes.
- DrugsGaz: a self-defined set of drugs that were left in the previous dictionary. This dictionary does not have ATC Codes. Due to the fact that we defined the list of drugs, there was no possibility of obtaining automatically their related ATC codes.
- MedDra: a set of adverse effects retrieved from MEDDRA.
- DiseasesUMLS: a set of diseases obtained from the UMLS database.

Language Identification: This module uses the external service Textalytics and identifies the language in which the text is written and labels the (GATE) document with a feature (‘lang’). The language identifier is later used for filtering out texts that are not written in Spanish.

Medical Annotations Filter: This filter is a GATE JAPE (Java Annotation Patterns Engine). It filters all the entities that have been annotated by Topics Analyzer and which are not from the medical domain. There are three entities (from medical domain) that are kept: ‘Drug’, ‘Effect’ and ‘Disease’.

This filter is needed because the external service (Textalytics, Topics Analyzer) uses not only dictionaries defined by us but also general domain dictionaries, so that many non-related entities are recognized (and they must be filtered out for avoiding problems in later processing).

Disambiguation Filter: The disambiguation filters have been applied in order to avoid false positive cases that were produced by named entities that are also common nouns or verbs. There are characteristics that call for filtering an annotation or a whole text; some of them are:

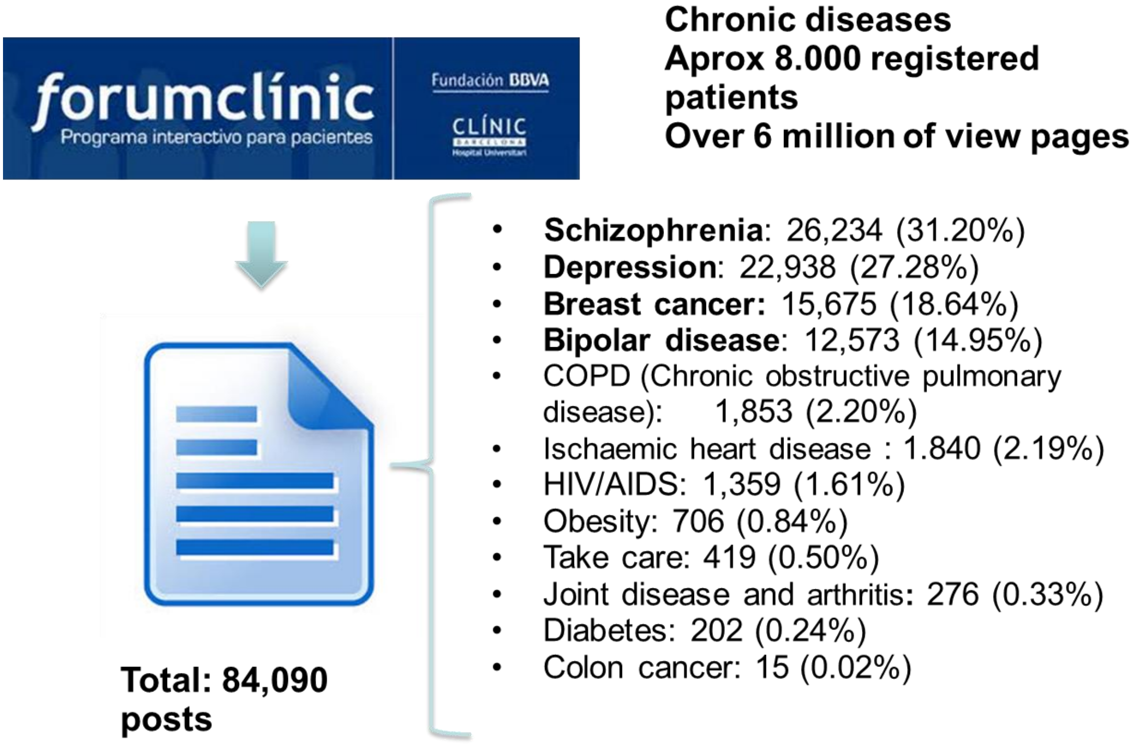
1. If the language of the comment is not Spanish, it is discarded.
2. If a part of the comment has a unique medical (drug, effect or disease) annotation, it is not discarded.
3. If a part of the text has a medical annotation and a morphosyntactic annotation then various cases must be considered:
    a. If there is another medical annotation in the whole text, the annotation is kept.
    b. If the text contains medical context (determined by a list of selected terms such as *diagnosticar*, *tomar*, *dosis*, *tratar*, *controlar*, *prescribir*, *producir*, *ingerir*, …..), then the annotation is kept.
    c. Otherwise, the annotation is deleted.

Relations Manager: This module analyses the annotated entities and determines the relations between drugs and effects. When a pair drug-effect is found, it is checked if this relation is also present in the SpanishDrugEffect Database. See (Segura-Bedmar et al. 2014b) for more details about this database. If it is stored in the database, it is annotated as a well-established relation (indicating its type: indication or adverse effect). If it is not present in the database, the relation is annotated as a possible relation between the drug and the effect.

2.4 Corpora and Evaluation

In order to evaluate the linguistic processor we have used a corpus extracted from Forumclínic[9], an interactive web page intended for patients to increase their degree of autonomy with respect to health issues, using the opportunities given by the newest Web technologies. Its target is to improve citizen's knowledge on health, diseases and their causes, as well as the efficiency and safety of the preventive treatments and medicines, so that they can get involved with the clinical decisions which attain them.

Forumclínic users are from all over the world, but a significant data is the fact that 46% of the webpage visits come from Spanish speaking countries in America. In total, the number of a million users was reached in 2011, and it maintains a steady increase since it was created, in 2007. See Figure 2 for information about structure and contents of this forum.



forumclínic
Programa interactivo para pacientes
Fundación BBVA
CLÍNIC BARCELONA Hospital Universitari

Chronic diseases
Aprox 8.000 registered patients
Over 6 million of view pages

- **Schizophrenia**: 26,234 (31.20%)
- **Depression**: 22,938 (27.28%)
- **Breast cancer:** 15,675 (18.64%)
- **Bipolar disease**: 12,573 (14.95%)
- COPD (Chronic obstructive pulmonary disease): 1,853 (2.20%)
- Ischaemic heart disease : 1.840 (2.19%)
- HIV/AIDS: 1,359 (1.61%)
- Obesity: 706 (0.84%)
- Take care: 419 (0.50%)
- Joint disease and arthritis: 276 (0.33%)
- Diabetes: 202 (0.24%)
- Colon cancer: 15 (0.02%)

Total: 84,090 posts

Figure 2: ForumClinic Structure

We created the first Spanish corpus annotated with drugs and effects (SpanishADR, see (Segura-Bedmar et al. 2014a)). This corpus was annotated by two annotators and consisted of 400 user messages collected from Forumclinic. The size of the corpus is

[9] http://www.forumclinic.org/

26,519 tokens, whereas each message contains an average of 3.15 annotations (0.48 drugs, 1.42 effects and 1.25 relations). Moreover, it contains 189 drug annotations, 568 effect annotations and 164 drug-effect relations. An assessment of the inter-annotator agreement (IAA) revealed that while drugs showed a high IAA (0.89), their effects pointed to moderate agreement (0.59). This may be due to drugs having specific names and being limited in number, however their effects are expressed by patients in many different ways due to the variability and richness of natural language. We have tested the pipeline described in Section 2.3 using this newly created SpanishADR Gold standard.

Concerning NER, Table 2 shows precision (P), recall (R) and F-measure evaluating drug recognition. The main source of false negatives for drugs seems to be that users often misspelled drug names. Some generic and brand drugs have complex names for patients. Some examples of misspelled drugs are *Avilify* (Abilify) or *Rivotril* (ribotril). While Textalytics provides a technique for fuzzy NER, this one needs to be better tuned to this specific domain. Another important source of errors was the abbreviations for drug families. For instance, *benzodiacepinas* (benzodiazepine) is commonly used as *benzos*, which is not included in our dictionary. An interesting source of errors to point out is the use of acronyms referring to a combination of two or more drugs. For instance, FEC is a combination of Fluorouracil, Epirubicin and Cyclophosphamide, three chemotherapy drugs used to treat breast cancer. Related to false positives some drug names such as *alcohol* (alcohol) or *oxígeno* (oxygen) can take a meaning different than the one of pharmaceutical substance. Another important cause of false positives is due to the use of drug family names as adjectives that specify an effect. This is the case of *sedante* (sedative) or *antidepresivo* (antidepressant), which can refer to a family of drugs, but also to the definition of an effect or disorder caused by a drug (sedative effects)

Drugs			
	R	P	F-Measure
strict	0,68	0,85	0,76
lenient	0,68	0,85	0,76

Table 2: Evaluation measures in drug recognition

Table 3 shows precision (P), recall (R) and F-measure evaluating effect recognition. The major cause of false negatives was the use of colloquial expressions to describe an effect. Phrases like *me deja ko* (it makes me KO) or *me cuesta más levantarme* (it's harder for me to wake up) were used by patients for expressing how they felt. These phrases are not included in the dictionary. The second highest cause of false negatives for effects was due to the different lexical variations of the same effect. For example, *depresión* (depression) is included in our dictionary, but their lexical variations such as *deprimido* (depress), *me deprimo* (I get depressed), *depresivo* (depressive) or *deprimente* (depressing) were not detected. Nominalization may be used to identify all the possible lexical variations of a same effect. Another important error source of false negatives was spelling mistakes (eg. *hemorrajia* instead of *hemorragia*). Many users have great difficulty in spelling unusual and complex technical terms. This error source may be handled by a more advanced matching method capable of dealing with the spelling error problem. The use of abbreviations (*depre* is an abbreviation for *depresión*) also produces false negatives. Techniques such as lemmatization and stemming may help to cope with this kind of abbreviations.

Effects			
	R	P	F-Measure
strict	0,43	0,75	0,54
lenient	0,47	0,83	0,6

Table 3: Evaluation measures in effect recognition

Table 4 shows precision (P), recall (R) and F-measure evaluating relation extraction taking in to account drug-effect pairs annotated in the corpus (the objective is to evaluate relation extraction task regardless of NER task). Regarding the false positives, a cause of error is SpanishDrugEffectDB could include incorrect relations due to the fact that it was automatically obtained and it has not been manually revised. Another source of errors is the lack of context resolution. This means that, despite correctly detecting a drug and an effect (according to the drug package information), the context of the text did not fulfil the requirements to properly consider it a relation. Moreover, the lack of co-reference resolution introduces another important source of error for false positives; terms such as *enfermedad*, *efecto*, *tratamiento* and other have to be solved. An interesting source of errors is the lack of negation resolution, which means that despite the fact that the user specifies that he/she did not experience an effect after taking a drug, the system annotates the relation. Finally, the complex sentences (coordinated and subordinated sentences) in a comment may mislead the system into annotating a relation which is not correct.

		SpanishDrugEffectDB			Coocurrences		
Window size		R	P	F-Measure	R	P	F-Measure
30	strict	0,08	0,57	0,14	0,63	0,44	0,52
	lenient	0,13	0,96	0,24	0,88	0,61	**0,72**
100	strict	0,1	0,34	0,16	0,74	0,26	0,38
	lenient	0,23	0,74	0,35	0,99	0,34	0,51
250	strict	0,12	0,32	0,17	0,17	0,75	0,33
	lenient	0,24	0,67	**0,36**	1	0,29	0,45

Table 4: Evaluation measures in relation extraction (over drug-effect annotated pairs in goldstandard)

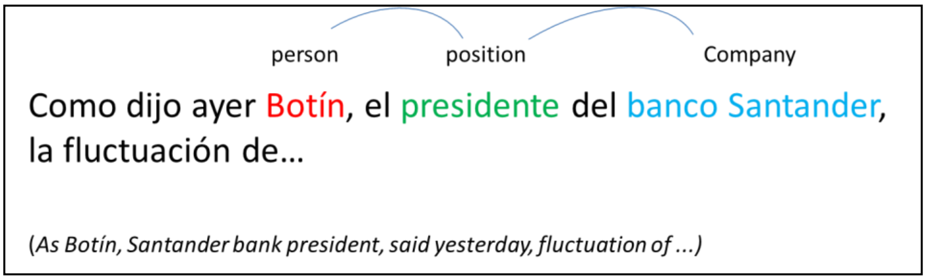
Finally, concerning false negatives Table 4 shows that a great number of drug-effect pairs appearing in the corpus are not covered by the SpanishDrugEffectDB (recall is very low), that is, this database does not include all drugs effects. Therefore, the corpus has only 164 relations and it is difficult to conclude about the database coverage. Other studies are reported in (Segura-Bedmar et al, 2014c) where a distant supervision method for relation extraction is analysed using the overall Forumclinic corpus (84,000 user comments) as training and testing dataset.

3. Finantial Use Case

3.1 Use Case 1: Detection of Job Positions in Companies

Sometimes it is difficult to know who is in charge of which position in a given company. Although there are some sources providing this kind of information (for example, stock market web sites), it is not always easy to obtain. So, a system to extract these data from news or blog entries text is proposed; it is very common for specialized online news sites to publish new designations in companies.

This use case has been defined to detect names of Persons, Positions and Companies. The aim is to get people together with the companies they work in. An example of this type of pattern is shown in Figure 3.



Como dijo ayer Botín, el presidente del banco Santander, la fluctuación de…

(As Botín, Santander bank president, said yesterday, fluctuation of …)

Figure 3: Example of financial pattern

This processing is performed by analysing linguistically the text and extracting the patterns based on JAPE (Java Annotation Patterns Engine) filters. The processing is done by means of a GATE pipeline.

3.1.1 Resources Description

A dictionary of Spanish job names obtained by manually extracting lists of job offers from different websites has been defined. Once the lists were obtained, we proceeded to perform a manual cleaning to remove specifications such as: *administrador jefe* (chief manager) and *administrador* (manager) because these combined positions are identified later with the filters explained below.

The dictionary is finally composed of:

- 52 terms with adjectives related to positions, for example, *adjunto* (assistant) *jefe* (chief), *corporativo* (corporate) etc.
- 18 names of positions that a person can work in, for example, *consultoría* (consultancy) or *gestión* (management)
- 57 names of roles that can qualify a person, for example, *president* (president), *director* (head), etc.

Existing ontologies, such as the OAEI initiative[10] or the work by the OEG Group[11], have been taken into account, but specific resources for the Spanish language have not been found,

3.1.2 Language Processing: GATE Pipeline

The different components of the language processing pipeline are shown in Figure 4 and their description are shown next.

- *Morphosyntactic Parser:* This module uses the external service Textalytics to analyze morphosyntactically the input text. The output is a GATE annotation set composed by all the annotations generated by the external service.
- *Positions Gazetteer*: This module generates an annotation set that contains all the entities present in the dictionaries: Positions, Adjectives and roles dictionaries used in this pipeline.

[10] Ontology Alignment Challenge, http://wissensnetze.ag-nbi.de/oaei/jobs.htm

[11] Human Resources Management Ontology, http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/99-hrmontology

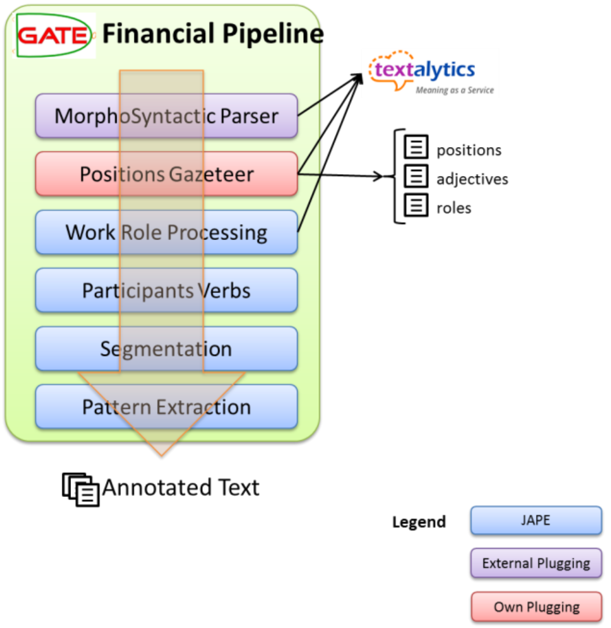


Figure 4: Financial Pipeline

- *JAPE Filters*: The different implemented JAPE filter are:
  1. WorkRoleProcessing: annotates the roles that are composed of several annotations of the gazetteer.
  2. ParticipantVerbs: annotates the verbs defined as participation (for example, *participar*, *asesorar*, *trabajar*, …..)
  3. DesignationVerbs: annotates the verbs defined as designation
  4. PersonCompany: extracts the person and Company entities from Textalytics morphosyntatic analyzer.
  5. Segmentation: annotates each delimiter between sentences.

- *Pattern Extraction*: Once all the information has been extracted and cleaned, a set of patterns are searched in order to detect person-company-role patterns. The different patterns that are analysed are:
  1. Person, Role Company.

     Bill Gates, president of Microsoft, …
  2. Role Company Person

     The president of Microsoft, Bill Gates, …
  3. Company/Role DesignationVerb(**Active** mode) Person Role Company
  4. Company/Role DesignationVerb(**Active** mode) Role Company Person

     Bill Gates/Microsoft has appointed Steve Ballmer as president of Microsoft

5. Company/Role DesignationVerb(**Pasive** mode) Role Company
   Steve Ballmer has been appointed as president of Microsoft
6. Person ParticipationVerb Company
   Steve Ballmer presides Microsoft since June.

3.2 Use case 2: Companies Reputation Analysis in Spanish

Another use case defined for the Spanish language in the financial domain has to do with reputation analysis of companies from online data. To do so, news texts are collected form online sources (online publications dealing with financial issues), mentions to given companies are extracted, the mood is analysed (by performing sentiment analysis) and an automatic text classification process around a reputation model is also done.

3.2.1 Reputation model

The reputation model used has not been developed as part of TrendMiner but applied to the use case, in order to see integration possibilities in the system already developed within TrendMiner. This model uses the following dimensions to classify mentions of a company:

i. *Oferta* (Offer)
ii. *Trabajo* (Work)
iii. *Integridad* (Integrity)
iv. *Estrategia y liderazgo* (Strategy and Leadership)
v. *Innovación y flexibilidad* (Innovation and Flexibility)
vi. *Responsabilidad Social* (Social Responsibility)
vii. *Situación Financiera* (Financial Situation)

Existing models, such as RepTrak® [12] or Merco®[13], have been taken into account

[12] The RepTrak Framework, http://www.reputationinstitute.com/about-reputation-institute/the-reptrak-framework
[13] Merco, http://www.merco.info/en/pages/1-que-es-merco

3.2.2 Sentiment analysis model

Regarding the sentiment analysis, models included in ontologies such as ONYX and MARL[14] have been considered (See D2.1.2, section 2.2.4 on details how the MARL model has been integrated in TrendMiner). To perform this sentiment extraction, the text provided is analysed to determine if it expresses a positive/negative/neutral sentiment. For this purpose, the local polarity of the different sentences in the text is identified and the relationship between them evaluated, resulting in a global polarity value for the whole text. Besides polarity at sentence and global level, the sentiment analysis integrated in TrendMiner uses advanced natural language processing techniques to also detect the polarity associated to both entities and concepts in the text. Ten different levels of polarity have been defined[15]:

P --, very weak positive
P -, weak positive
P, Positive
P+, strong positive
P++, very strong positive
N--, very weak negative
N-, weak negative
N, negative
N+, strong negative
N++, very strong negative

This sentiment analysis is combined with the reputational information to depict a figure of the reputational state of a given company as reflected in online media. The data collected for this TrendMiner Financial use case in Spanish is around two given companies: *Telefónica* and *Orange* (*Orange has launched an acquisition process on Jazztel last month*; the idea behind is to compare how *Telefónica* and *Orange* corporate image are affected by this situation).

[14] Eurosentiment project, http://eurosentiment.eu
[15] Those values will be harmonized with the values used in other use cases of TrendMiner.

3.2.2 Data sources

The document collection used in this scenario is formed by tweets, blog posts and news taken form RSS channels provided by online news media. So, around 2.000 texts have been processed according to the distribution included in Table 5.

Source	Number of Documents
Twitter	982
Jazztel	450
Orange	253
Telefónica	279
Blogs	517
News	50

Table 5: Data about text collections

All, these documents are in Spanish and have been accessed and pre-processed by services provided in WP5 and results of our processing is stored to the TrendMiner Repository through the 'Direct RDF Import' service developed also in the context of WP5 by the partner ONTO.

4. Polish Political Tweets use case

Responding to the need of socio-political analysts, politicians and journalists in Poland, eagerly awaiting the platform for effective monitoring of the newest data, a Polish Political Twitter monitoring use case scenario was implemented using TrendMiner architecture extended with the Polish natural language processing tools. Tweets from official accounts of parties, MPs, public institutions, political journalists and freelance analysts are systematically collected, automatically annotated with language tools to discover topics (represented by frequent noun phrases), named entities, and locations, and forwarded to TrendMiner visualisation interface. This allows for discovering new trends in the social media stream, tracking sentiment of communication and analysing topics of the public discourse over time.

4.1 Resources

The Sejm database

This database with all Polish Members of Parliaments (MPs) from 1991 until now was achieved from Sejm (the lower chamber of the Polish parliament). It contains complete information about all Sejm MPs and several cadencies of Senate, and it has been semi-manually supplemented with remaining records. The main source of the update was (Kaleta, 2010).

Money.pl

The Money.pl portal contains information about Polish parliamentarians from every cadency: http://www.money.pl/gospodarka/politycy/. Those Web pages have been crawled and data parsed by scripts.

Genetate data with Polish Members of the Europarliament

The Sejm database did not contain information about Members of Europarliament, so that this information was taken from Wikipedia:

http://pl.wikipedia.org/wiki/Polscy_pos%C5%82owie_do_Parlamentu_Europejskiego_2004-2009

http://pl.wikipedia.org/wiki/Polscy_pos%C5%82owie_do_Parlamentu_Europejskiego_2009-2014

http://pl.wikipedia.org/wiki/Polscy_pos%C5%82owie_do_Parlamentu_Europejskiego_2014-2019

Polish sentiment dictionary

The Polish sentiment dictionary (http://zil.ipipan.waw.pl/SlownikWydzwieku) contains information on sentiment of individual Polish lexemes computed using several classifiers and formulas described in (Wawer, 2012a), (Wawer, 2012b) and (Wawer and Rogozińska, 2012).Table 6 below displays some sample entries.

The first three columns reflect sentiment scores computed using supervised methods:

- Neutral (0) vs positive or negative (1).
- Negative (-1), neutral (0), positive (1).
- Very negative (-2), negative (-1), neutral (0), positive (1), very positive (2).

Entry	2-value-scale	3-value-scale	5-value-scale	SO-PMI
cudzoziemiec	0	0	0	-2.6
dziewczęco	1	1	2	-1.0
europejski	0	1	1	1.0
miasto	0	0	0	7.1
miły	1	1	2	6.8
okrutnie	1	-1	-2	-3.6
oryginalny	1	1	2	0.8
pretensjonalny	1	-1	2	-1.0

Table 6: Sample entries from the Polish sentiment dictionary

The last column is the SO-PMI (semantic orientation–pointwise mutual information) score calculated using the *svd*-based paradigm words selection.

The use of the dictionary and method of conversion of the scale into the target three-value TrendMiner tweet assessment it presented in section 4.3.3.

4.2 NLP tools

4.2.1 Morphological analyser

The linguistic processing on segmentation, lemmatization and tagging levels is executed by PoliTA (http://zil.ipipan.waw.pl/PoliTa), a meta-tagger which uses many individual POS taggers to combine their decisions and produce a potentially better output than any of the individual methods by themselves. It combines 4 existing Polish taggers:

Pantera (Acedański, 2010) is an adaptation of the Brill's algorithm to morphologically rich languages, such as Polish. Pantera includes several techniques of improving the tagging of inflectional languages, such as multi-pass tagging and transformation templates.

- WMBT (Radziszewski and Śniatowski, 2011) is a memory based tagger, which disambiguates the set of possible tags in multiple tiers. The number of tiers is equal to the number of attributes in the tagset, including the grammatical class. Tokens in each of the individual tiers are classified using a k-NN classifier.
- WCRFT (Radziszewski, 2013) is a tiered tagger, based on Conditional Random Fields (CRF), a mathematical model similar to Hidden Markov Models. A separate CRF model is used to disambiguate distinct grammatical attributes.

– Concraft (Waszczuk, 2012) is another approach to adaptation of CRFs to the problem of POS tagging. In Concraft, the CRF layers are mutually dependent and the results of disambiguation from one of the layers may propagate to an-other. The first of the two layers used by the tagger includes tags related to POS, case and person, while the second contains all other grammatical categories.

4.2.2 Noun phrase extractor

NP extraction is performed by Spejd – an Open Source Shallow Parsing and Disambiguation Engine (http://zil.ipipan.waw.pl/Spejd/) based on a fully uniform formalism both for constituency partial parsing and for morphosyntactic disambiguation — the same grammar rule may contain structure-building operations, as well as morphosyntactic correction and disambiguation operations. The formalism and the engine are more flexible than either the usual shallow parsing formalisms, which assume disambiguated input, or the usual unification-based formalisms, which couple disambiguation (via unification) with structure building. Current applications of Spejd include rule-based disambiguation, detection of multiword expressions, valence acquisition, and sentiment analysis. The functionality can be further extended by adding external lexical resources.

Spejd processing is powered by a grammar of Polish developed by Katarzyna Głowińska for the task of syntactic group recognition in the National Corpus of Polish (http://www.nkjp.pl/).

4.2.3 Named entity recognizer

Named entity recognition is performed by Nerf (http://zil.ipipan.waw.pl/Nerf), a statistical CRF-based named entity recognizer trained over 1-million manually annotated sub-corpus of the National Corpus of Polish and successfully used in the process of automated annotation of its total 1 billion segments.

4.2.4 Sentiment analyzer

Sentipejd (Buczyński and Wawer, 2008) is a Spejd-based sentiment analyzer created by extension of the morphosyntactic tagset with a semantic category, expressing properties of positive or negative sentiment. The rules for sentiment extraction were created semi-automatically with the help of statistical methods of collocation extraction. First, a list of word bigrams with the highest value of Frequency biased Symmetric Conditional Probability (Buczyński, 2006) was created, to find collocations which are both common in the corpora and strongly dependent. A simple heuristics was used to discard proper names from the results. Then, the collocations were manually generalised into rules and assigned sentiment values.

Within TrendMiner project Sentipejd has been made available as a web-service and included in an already existing framework for publishing language-related resources called Multiservice (Ogrodniczuk and Lenart, 2012).

The multiservice tool supplements a shallow grammar of sentiment patterns with the second component: the sentiment dictionary of 3268 lexemes (the same as the one used for Tweet processing). This tool is capable of detecting and computing simple compositional sentiment structures, such as mixtures of various semantic word classes and part-of-speech types. Therefore, the classes of words taken into account in the rules include not only sentiment, but also negations, nullifiers, amplifiers, high- and low intensity markers. In the JSON output of the tool, we add sentiment as yet another tag type, along with morphosyntactic tags. The multiservice sentiment component is aimed at more syntactically elaborate and clean (less noisy) language than Tweeter, such as news. TrendMiner contribution has been integration of the tool within the Thrift framework and enrichment of the JSON output layer to include sentiment which makes it compatible with other de facto standard tools for processing Polish, available in common National Corpus of Polish-based format (Przepiórkowski et al. 2012).

4.3 Tweet language processing chain

This section describes an efficient processing pipeline for Polish political tweets. Key observation to develop TWEET PROCESSOR was that the tweets we are working with are written using a quite high-quality language, contrary to most Twitter data examples presented in the literature. It seems to be the result of a carefully selected set of users, whose messages we are dealing with (although the key to that selection was not the language quality, but the topics mentioned by these users).

4.3.1 Architecture

Twitter political data are obtained by monitoring all tweets of few hundred users from the Polish political Twitter account list. This procedure produces up to few million tweets per month; therefore our pipeline should be able to cope with such volume of data. The whole process of collecting tweets is presented in more detail in deliverable D5.5: Deployment of Web services for new cases.

The input to our processing pipeline is a Mongo database[16], in which collected tweets are stored. General view on architecture is presented in Figure 5. Tweets are fetched from Mongo database and processed with TWEET PROCESSOR tool, which uses standalone MULTISERVICE server to perform some of the NLP tasks. The output of our pipeline is the same Mongo database, in which we update original tweets content with results of the NLP analysis.

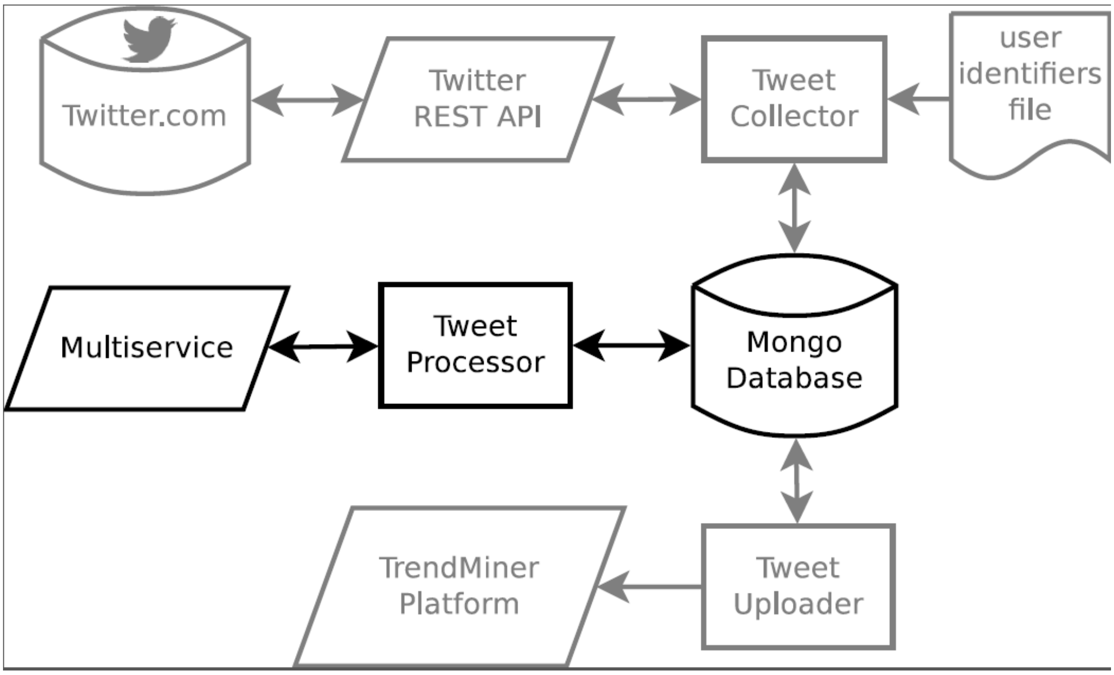


Figure 5: NLP architecture

4.3.2 Existing TrendMiner solution for English

A Java tool for processing social media data in English was created as part of the TrendMiner project (Pretiouc et al., 2012). Source code may be downloaded from:

http://github.com/sinjax/trendminer.

[16] http://www.mongodb.org/

The project contains several modules, from which we are interested in TWITTERPREPROCESSINGTOOL module, responsible for NLP tweet processing. This program may be run either as a command line tool, or as a distributed tool that runs in the Hadoop Map-Reduce framework. This report focuses on adjusting the command line tool version to be able to efficiently process Polish tweets. Authors report the speed of 0.5 mln tweets processed through a tokenization and language detection pipeline on a single core in 1 hour, which should be more than sufficient for our data volume, making distributed processing unnecessary.

TWITTERPREPROCESSINGTOOL handles multiple input/output formats. As Twitter data comes in JSON format, the tool uses JSON internally – each step in the pipeline adds new fields to the record in a special “analysis” field. Available pipeline steps (realized by separate modules) are:

- LANG_ID – language detection is performed by an OPENIMAJ[17] library reimplementation of the LANGID.PY[18] tool with the default trained model (for 97 languages). This language detection doesn’t depend on any other processing steps, assumes one language/tweet and doesn’t rely on user’s self-reported profile language.
- TOKENISE – tokenization performed by a Twitter-specific tokenizer, an OPENIMAJ implementation based on the twokenise by Brendan O’Connor. Tokenizer works through a chainable set of regular expressions, handling:
- URLs,
- strange usage of punctuation,
- emoticons,
- hashtags, retweets, @ mentions,
- abbreviations, dates.

[17] See (Hare et al., 2011) and http://www.openimaj.org.

[18] See (Lui and Baldwin, 2012) and at https://github.com/saffsd/langid.py.

- PORTER_STEM – Porter stemming (Porter, 1980) performed only for English tweets, relying on already tokenised input. Implemented in Lucene[19] library.
- REMOVE_STOPWORDS – token filering, removing all stopwords found in a stopword predefined list. Lists for several languages available, but not for Polish.
- SENTIMENT – detection of words in the tweet text, belonging to two lists: list of words with positive sentiment and list of words with negative sentiment. Only available for English.

TWITTERPREPROCESSINGTOOL was used as a starting point for development of Polish TWEET PROCESSOR tool. Main changes included:

- possibility to use MongoDB as input and output data store,
- implementation of Polish-specific processing modules,
- code simplification,
- possibility to "break out of the pipeline" when one of the modules returned error or decided that further processing is not necessary (this is useful, when language detection module detects language other than Polish),
- batch processing of multiple tweets by single module (this speeds up significantly the communication between modules and remote servers, because we do not have to create a separate network request for every single tweet).

4.3.3 Polish Tweet Processor

The task TWEET PROCESSOR is to augment the tweets present in the Mongo database with the results of NLP analysis. A single run of the tool performs the following steps:

1. Select from the database tweets:
    - either not yet processed,
    - or processed with an earlier version of the TWEET PROCESSOR.

   The tweets with the newest creation time have the highest priority for processing.

[19] http://lucene.apache.org

2. Process tweets.
3. Save the results of the analysis in the database.

For visualization purposes (visualization is described in D5.5, Section 3.4), the NLP pipeline must provide the following values for each tweet in Polish (tweet_json is the JSON Mongo object of a tweet):

1. Language of a tweet must be detected and stored in tweet_json[analysis][language][value] as string. If it is not "pl", tweet should not be NLP processed.
2. Sentiment of a tweet in a numeric value from interval [−1,1], where –1 stands for negative sentiment and 1 stands for positive sentiment. This value should be stored in tweet_json[analysis][sentiment][value] field (the method of recalculation of the different scales is explained later in this section).
3. Locations mentioned in the tweet, stored as list of locations in tweet_json[analysis][locations][values] field, each item on that list being a map and having URI (uri), label (label), geographical latitude (latitude) and longtitude (longtitude) fields.
4. Entities mentioned in a tweet, stored as list of entities in tweet_json[analysis][references][values] field, each item on that list being a map and having label (label) and URI (uri) fields.
5. Topics of a tweet in Polish, each topic represented by a keyword or multi-word phrase. These topics should be stored in a list in tweet_json[analysis][topics][values] field, each item being a string representing a single topic.
6. For clustering purposes, tokens parsed from tweet text in a list in tweet_json[analysis][tokens] field, each item on that list being a map, having lemmatized form in base field.

These requirements are fulfilled by our tool via a pipeline of modules:

- Language detection module – provides language of the tweet,
- Preprocessing module – cleans original tweet text to make it correct Polish language,

- Multiservice module – processes cleaned tweet text with state-of-the art NLP tools for Polish,
- Sentiment analysis module – analyses the sentiment of the tweet,
- Entity detection module – detects named entities mentioned in tweet,
- Topics detection module – provides topics mentioned in tweet,
- Location detection module – detects location names mentioned in tweet.

Each module may use the output of previous modules. Details of the module implementation are described in the next sections.

Language detection module

We use language detection from TWITTERPREPROCESSINGTOOL performed by an OPENIMAJ library reimplementation of the LANGID.PY tool with the default trained model for 97 languages. The algorithm is used on original tweet text content. If language detected was not Polish, we do not process the tweet by next modules, as further processing would be inaccurate.

The precision of language detector is very high, as out of manually examined 3000 tweets which were marked as Polish, all were marked correctly. Recall is yet to be investigated, but it is certainly above 85%, as this is the amount of all tweets we are collecting which are detected as written in Polish.

Preprocessing module

Manual analysis of few thousand of Polish tweets written by political commentators led to an observation, that language used in that discourse is rather well-formed and close to the language quality of news articles. This was quite surprising (for example, named entities were almost always written correctly in terms of capital letters), but at the same time promising, as NLP tools created for general purpose should perform well on such data.

Nevertheless, before performing NLP analysis, some preprocessing was needed, mainly because of specific Twitter mechanisms. Our preprocessing module does several important cleaning steps:

- removes words which were truncated by Twitter as exceeding maximum tweet character size,
- cleans parts of the message that indicates that a post is retweeted,
- removes URL-s from tweets,
- “decodes“ hashtags, trying to replace them with regular words (by splitting on camelCase with spaces and removing # sign,
- replaces user mentions with user names, indicated in user profiles.

Then, the tweet text is passed to LANGUAGETOOL[20], an open-source, rule-based proofreading tool, which is mainly used to insert missing diacritics or correct misspelled words.

Multiservice module

After the tweet text is cleaned by our preprocessing module, we process it with a pipeline of state-of-the-art NLP tools for Polish, offered by MULTISERVICE[21] – a robust linguistic Web service for Polish, combining several mature offline linguistic tools in a common online platform. We use the following pipeline:

1. Segmentation and part of speech tagging by WCRFT[22],
2. Shallow parsing by SPEJD[23],
3. Named entity recognition by NERF[24],
4. Mention detection by MENTION DETECTOR[25].

[20] See (Miłkowski, 2010) and http://languagetool.org.

[21] See (Ogrodniczuk and Lenart, 2012) and http://glass.ipipan.waw.pl/multiservice/.

[22] See (Radziszewski, 2013) and http://nlp.pwr.wroc.pl/redmine/projects/wcrft/wiki.

[23] See (Przepiórkowski and Buczyński, 2007) and http://zil.ipipan.waw.pl/Spejd.

[24] See (Waszczuk et al., 2013) and http://zil.ipipan.waw.pl/Nerf.

[25] See (Kopeć, 2014) and http://zil.ipipan.waw.pl/MentionDetector.

Results of that processing are later used by subsequent modules. For performance reasons, requests are sent to MULTISERVICE in batches, containing configurable number of tweets.

Sentiment analysis module

Sentiment of a tweet is calculated as a real value within the interval [−1,1]:

$$S = \frac{1.5t_{pos} - t_{neg}}{t}$$

where

- S is the sentiment value,
- t is the number of tokens in tweet,
- t_{pos} is the number of tokens in tweet having positive sentiment,
- t_{neg} is the number of tokens in tweet having negative sentiment.

In the Polish Tweet Processor application, we include the 3-value version of the dictionary (negative (-1), neutral (0), positive (1)). Computing overall sentiment of a tweet requires us to take into account not only the sentiment orientation of individual words (here, positive or negative) but also a ratio of negative and positive words to all words in this tweet. For example, tweets with multiple neutral words and only one negative words should have higher scores than tweets composed of mostly negative words and only several neutral ones.

Each tweet is tokenised by a POS tagger within the Multiservice module, positive and negative tokens are recognized using a list of 3277 lemmatized word forms with associated sentiment. The bias in the Internet discourse towards negative sentiment is balanced by having a 1.5 weight in favour of the positive sentiment, as reported by multiple papers and corpus frequency studies.

Entity detection module

Entities in tweet text are obtained in two ways: first, from the named entity recogniser in Multiservice module. Second, all user mentions (which were transformed by the preprocessing module to user names) are marked as entities.

To match them against Polish political ontology, a simple text search of lemmatized form of the entity is performed against the ontology. If a match is found, a URI to the entity in the ontology is saved with the entity, otherwise only entity lemmatized form as the entity label.

Lemmatization of multi-word phrases for Polish is not a trivial task, the way it was performed is described in the next section.

Topic detection module

We decided to treat all entities (not necessarily named entities) in the text as its topics, not limiting ourselves to a constant topic set. This enables to observe new topics emerge from the Internet discussions. Therefore we used Multiservice's mention detector, a preliminary step for coreference resolution, as the topic provider.

To discard some non-frequent mentions to be marked as topics, our tool stores a dictionary of all mentions found in all already processed tweets along with their counts. This allows us to set an arbitrary threshold, above which we decide, that given entity was mentioned sufficient number of times to treat it as a well-formed topic.

On the other hand, we also discard some common stop-word mentions, such as for example pronouns.

Similar to the entity detection module, we again return lemmatized forms of single- and multi-word expressions. Single word expressions have lemmatized forms provided by the tagger, but to find correct lemmatized forms of multi-word expressions, we use corpus-based approach. Along with frequency statistics of mentions, we also store frequencies of their inflected forms. Out of all inflected forms with syntactic head marked by the tagger as having nominal case, the most frequent one is taken to be the lemmatized form of the whole expression. Such approach yielded promising results in our initial experiments.

Location detection module

Location detection module tries to match in DBPEDIA all named entities of geographical type found by the named entity recogniser in tweet text. If such match is found, we extract GPS coordinates of that location, if they are present in the database.

Again, we use corpus-based multi-word expression lemmatization to provide correct location labels and facilitate their search in DBPEDIA.

4.4 Corpora and Evaluation

Tweets from a selected set of users, which are likely to tweet about politics have been collected. The application monitors their accounts and fetches their tweets on a regular basis. 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.

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.

4.4.1. Twitter Language Analysis

Our approach to analyze text data was to manually inspect a sample of it and perform a categorization of common phenomena, which distinguish language of our Twitter users from that of general Polish writers.

For that purpose, we selected 10,000 most recent tweets from our data. Then, from this set we were sampled one tweet per each user, again starting from the most recent ones. When we already sampled tweets for each user, who published a tweet in our 10,000 tweet sample, we started sampling again from that sample, taking one tweet per each user (skipping tweets already taken). This approach was repeated until we obtained 3,000 tweets dataset. Presented way of selecting tweets to that dataset was supposed to maintain high variability in language use, as many different authors produced tweets in our sample, yet still having a higher number of tweets of more active users.

This 3,000 tweets dataset was manually inspected and annotated with some common language phenomena, which represent a challenge for NLP tools. For each tweet, we marked several categories of errors, explained below.

Missing diacritical marks

A word without diacritics may have a different meaning or no meaning at all, which phenomenon increases the difficulty of text processing. Missing diacritics is a subclass of a general spelling error, but whenever adding diacritics was sufficient for getting a proper interpretation of a word (e.g. mąz –> mąż), the tweet was marked with that category.

Missing foreign diacritics (as in exposé, Müller) were not marked although they are regularly corrected by spellcheckers, yet they are very common because of problem to obtain such characters with a standard Polish keyboard.

Whenever both forms (with and without diacritics) were acceptable in the content, the more probable variant was selected, as in "@mblaszczak do JK powiedział chyba "prowokacja"? Odczytuje z ruchu warg", when 3rd person "odczytuje" is less likely to be used in this context than 1st person "odczytuję").

In a few cases this class also lists other related problems, such as excess ("pisałęm" -> "pisałem") or foreign diacritics ("pomarzyč" –> "pomarzyć").

Abbreviations

Almost all abbreviations were counted, including the following subcategories:

- abbreviations of named entities ("FB" –> "Facebook", "GW" –> "Gazeta Wyborcza", "PiS", "PO")
- initials of people names ("J.K." –> "Jarosław Kaczyński", "JVR" –> "Jan Vincent Rostowski"), also including ad-hoc abbreviations ("PDT" –> "Premier Donald Tusk", "PEK" –> "Premier Ewa Kopacz")
- foreign abbreviations frequently used in Polish ("CIT", "NATO", "OK")
- currency symbols ("zł", "€")
- abbreviations without the (obligatory) dot at the end ("nt.", "prof")
- ad-hoc abbreviations of common words, resolvable from the context ("dzienn." –> "dziennikarz", "dokł" –> "dokładnie")
- certain proper names, initially formed as abbreviations ("TVP", "TVP2", "TVN", "CO2", but not "ZET" in "Radio ZET").

Some abbreviations may not be present in morphological dictionaries, which implies unrecognised words in tweet content. Abbreviations of English or Polish slang expressions were excluded from this group and counted as slang words ("other" group). Code or brand names ("F16", "BMW") which are not acronyms were also not treated as abbreviations.

Trimmed words

Maximum size of a tweet is 140 characters. Sometimes users try to publish longer messages, in such case they are often automatically trimmed in the middle of a word, username, hashtag or URL. Some applications which act as an intermediate between the user and Twitter do trim not necessarily the first word which exceeds the maximum tweet size, but do it in a different way.

In any case, such trimming is marked with triple dots, often leaving only first part of a word, making it difficult to understand.

Case problems

Entity names started with lowercase or unnecessary capitalization of a whole word. Such case problems increase the difficulty of finding named entities in the text.

Lowercase letter in the beginning of the sentence was not counted as case problem.

Spelling errors

Errors other than missing diacritics; one of the following subcategories:

- misplaced or missing letters („swrdecznie” –> „serdecznie”, “członkowstwo” –> “członkostwo”)
- words stuck together due to missing separating spaces (but punctuation problems such as an extra space between a word and a comma were not counted)
- words separated due to excess space (“był by” –> “byłby”, ”gospodarkama” –> “gospodarka ma”)
- repetitions of letters or their sequences (“okeeeeeej” –> “okej”, “Hmmmm” –> “Hmm”; not necessarily a spelling error, but not frequent enough to form a separate class).

Emoticons

Presence of emoticons is something difficult for NLP tools created for traditional written texts, such as books or news articles. Their presence requires special treatment, especially to interpret the sentiment of a tweet.

Foreign language

Most of non-Polish expressions in Polish tweets were English (“dream team”), but German and Latin was also observed. Several subcategories of foreign language use can be distinguished:

- single foreign words (“community”), also those functioning as slang expressions (“sorry”, “nerd”), possibly inflected (“Sammyego”, „iPhone’owi”)
- foreign phrases or sentences, both ad hoc interjections (“I like it”) and quotations (“Ora et labora”)
- titles (“Assassin's Creed Identity”)
- polonized foreign words other than named entities („retłitują”, “hendszejk”, “słitfocia”).

Other

Other interesting observed phenomena included:

- neologisms (“sorkokorki”, “Kopaczinho”)
- Polish (“kminię”, “Bolandzie”, “Lemingradu”, “Łomatko”, “pzdr” –> “Pozdrawiam) and foreign slang words and abbreviations (“OMG” –> “Oh My God”, “rl” –> “real life”), sometimes noted in the dictionaries of slang
- new words, still not present in the morphosyntactic dictionary, but likely to be included (“smartfon”, “audiobook”)
- compound words, with lack of interpretation probably resulting from misconfiguration or missing prefix in morphosyntactic dictionary (“homopropaganda”, “nadredaktor”)
- less frequent forms of common words, evidently missing from the morphosyntactic dictionary (“zmolestowanego”)
- non-standard transcription of common words ("nie-by-wa-łe-go”)
- inflected forms of named entities, particularly adjectives (“palikotowy”)
- forms intentionally distorted for stylistic reasons (“Swiętokrzysko”, “szłem”, “pachły, “wiater”, “jedenu”).
-

Statistics of issues

Table 7 presents the share of problems in presented categories in our 3,000 tweet sample.

Most common problem for NLP tools are the abbreviations. Over 26% of tweets contained at least one abbreviation, which makes it crucial to try to replace such words with their full forms.

The most common spelling error (occurring in over 10% of tweets) is missing Polish diacritic marks (hardly understandable in smartphone era) which make an important statement that this issue must be handled in the very beginning of text processing.

Emoticons probably shouldn't be considered an error, but still it is a feature of Twitter language not present in typical written texts. Their high frequency (12% of tweets) makes interpretation of emotions represented by emoticons an influential task.

Error category	**% of tweets with this problem**
Abbreviations	26.23
Missing diacritical marks	10.93
Emoticons	12.00
Trimmed words	6.03
Spelling errors	3.37
Foreign language	2.93
Other	2.77
Case problems	1.27
Any	49.4

Table 7: Categorization of unrecognized words representing typical problems

Trimmed words is also a frequent factor in Twitter communications – the strict limit on the number of characters quite often (6%) cuts some tweet content in a way, which requires either dropping trimmed words, or trying to guess, which word was trimmed.

Very interesting result of our manual analysis is that spelling errors other than missing diacritics are quite rare (about 3%). Probably it is the result of careful selection of monitored Twitter users (many official accounts of political parties of actors), which use rather correct language. It also applies to case problems, which are very infrequent. Compelling fact, that missing diacritics seems to have a different status than other type of language errors, as our selected users are rather careful in writing, yet are not so strict about using proper diacritic signs.

Foreign words occurred in less than 3% of tweets.

This analysis led us to the conclusion, that NLP tools created for general Polish language should be effective for our data type, given some preprocessing fixing diacritics and trimmed words, expanding abbreviations and correctly parsing emoticons.

4.4.2. Evaluation of lemmatization

Effective processing of Polish tweets by a language processing chain must include some kind of content interpretation and spelling correction – to enable use of higher-level tools such as noun phrase detectors and lemmatization of extracted noun phrases – to present extracted content to the user.

Our pipeline extracts topics from tweet content by finding expressions in tweet text. Important thing for topic visualization is finding a correct lemmatized form of such expressions.

In case of single-word expressions, lemmatization is performed via a morphosyntactic dictionary with help of a POS tagger. The case of multi-word phrases is a lot more difficult: in that case we can do a word-by-word lemmatization, but most often it is not a correct lemmatized form of the whole phrase.

As we are interested only in topic expressions which do occur in multiple tweets, our approach to lemmatization of MWEs was corpus-based. The idea was to collect the number of occurrences of all MWEs (in the inflected form they occurred in text) in our Twitter database, alongside with a word-by-word lemmatization and information, whether the inflected form was analysed by NLP tools as having its syntactic head in nominative case and singular number. In such case, it is likely that the inflected form of MWE is a lemmatization of that expression.

With such data, we were able to find for a MWE its lemmatized form simply by taking the most frequent inflected form (with the same word-by-word lemma as our query MWE) from the corpus, assuming we look only at forms with syntactic head in nominative case and singular number.

This procedure was evaluated by taking 1000 random MWE, occurring at least 10 times in our Twitter corpus, and checking validity of lemmas which were proposed for them by our algorithm.

The results are encouraging, as all 1000 lemmas were good morphologically. The only issue found was the problem with in correct capitalization, which for 332 MWEs was not perfect. For example, a common noun group may receive first capital letter, if it was most frequently used in nominative case and single number at the beginning of a sentence in our corpus.

This issue, however, is less serious than presenting to end user of our topic detection tool an incorrectly inflected lemma.

4.4.3. Evaluation of Spelling correction

Since most errors result from missing diacritic marks or wrong spelling, the obvious step is to integrate an existing automatic spellchecker for Polish to introduce the corrections.

Spelling correction issue for Polish is a difficult task due to inflection resulting in high number of distinct word forms. PoliMorf (http://zil.ipipan.waw.pl/PoliMorf), the largest morphological dictionary of Polish contains 44,341 lexemes corresponding to 4,223,981 word forms and 6,578,143 morphosyntactic interpretations. Without taking diacritics into consideration they are likely to be homographic which makes such tasks as adding diacritical marks difficult in frequent cases, but there are no evaluation data available, so we targeted evaluation of the best available spellchecking tool for Polish in context of tweet processing.

Data source

Using the 3,000 tweet sample (see section 2.2) as evaluation data, we performed manual evaluation of the sample and subsequent variants of the tool. 776 “lexical errors” were identified in the sample, corresponding to the following two types:

- misspelled words, including abbreviations and named entities (Polasat –> Polsat)
- words with missing diacritical marks (zapytac –> zapytać).

Other types of extra-lexical errors (punctuation, grammatical, usage, stylistic errors) were not marked.

LanguageTool

LanguageTool (Miłkowski 2010) is a language-independent rule-based open source proof-reading software able to detect frequent context-dependent spelling mistakes, as well as grammatical, punctuation, usage, and stylistic errors. It is regarded as the most extensive resource of this type for Polish, features hundreds of thousands of downloads and is available as a standalone tool as well as a plugin for LibreOffice/OpenOffice and Firefox.

LanguageTool correction rules are extensive which results in introducing errors for new words (“smartfonów” –> “smart fonów”), named entities (“Baracka Obamy” –> “Baranka Obawy”) and non-standard abbreviations (“pracow.” –> “placów.”) in the out-of-the-box solution (referred to as version LT0 later in this section). This verification resulted in evaluating two other settings of the tool:

- running only on words which are not entirely capitalized – which corresponds to a setting where all errors except for those in words regarded as abbreviations are corrected (LT1)
- running only on words which are not starting with a capital letter – which corresponds to a setting where all errors except for those in words regarded as named entities are corrected (LT2).

TrendMiner solution

Another version of the spell-checking solution (later referred to as TM) was created based on assumption that majority of errors are diacritic-related, therefore fixing only this problem could solve many issues without introducing new problems likely to be caused with extensive spelling correction.

TM solution is a baseline algorithm, using morphosyntactic dictionary to extract all possible strings, which by addition of some number of diacritics may represent a valid word, present in the dictionary. This gives a mapping from strings to possibilities for diacritic insertion, which produces a valid word. We also add a special case of not adding any diacritics, if the string without diacritics is already valid.

When a string we have in our mapping occurs in text, we have two options:

– leave it unchanged, if there is such option in the mapping,

– replace it with a chosen word from possible options in the mapping.

To have an efficient way to select valid replacement (or no replacement), we implemented a baseline algorithm, using a unigram frequency count, extracted for a large corpus. Using such data, we may simply select the option which produces most frequent word in our reference corpus.

The algorithm works in a different way depending on presence of any diacritic signs in a tweet we are correcting:

1. If the tweet does not have any diacritic signs, we allow to add diacritics to valid words (in this way the word “mowie” may be corrected to “mówię”, even that it is a valid dative form of a noun “mowa”).
2. Otherwise, we only try to add diacritics to strings, which are not valid words.

TM solution requires two resources: dictionary with valid word forms (we are using PoliMorf) and unigram frequencies (we are using unigrams from a 300,000,000 word balanced NKJP subcorpus).

Tool comparison

Table 8 and Table 9 show the figures correspondant to evaluation of LT and TM aproaches.

	LT0	LT1	LT2	TM
Undetected errors	126	164	268	184
Detected and corrected	614	576	472	589
Wrongly corrected	695	483	178	228

Table 8: Error correction statistics for all investigated settings

	LT0	LT1	LT2	TM
Precision	0.47	0.54	0.73	0.72
Recall	0.83	0.78	0.64	0.76
F1	0.60	0.64	0.68	0.74

Table 9: Evaluation of relevance of all investigated settings

The TM solution proves best; yet, it could still be improved, for example for using context larger than unigrams, or implementing a more sophisticated approach to decide, if the tweet author is likely to omit diacritics or not.

4.4.4. Analysis of Hashtag and username decoding

Hashtags and Twitter user names are frequently used in content not only as reference markers, but at the same time (mostly because of the 140 character limit) as part of communication, so it is not uncommon to see "Tylko u nas, minister @KosiniakKamysz podaje szczegóły propozycji z expose Ewy Kopacz."

To be able to effectively pass the communication to the language processing chain, hashtags and user names must be converted to plain text and used as standard words. Current solution applies the following methods to the process:

- account identifiers are replaced with account name retrieved from Twitter,
- hashtags decoded, with spaces inserted in place of camel case

5. Socio-psychological Analysis of Social Media Messages in Politics

Social Media (SM) is becoming an increasingly important channel for communications in politics. In Hungary, Facebook is the dominant SM platform, with 4.27M registered Hungarian users (59.2% penetration of 7.2M people with internet access, which represent 43% of the total population)[26]. No politician or political organization can afford to miss the opportunity of extending their influences by

[26] As of Dec. 31, 2013 (source: http://www.internetworldstats.com/europa.htm)

regularly publishing status update messages (posts) on their Facebook pages that are potentially accessible by all Facebook users (i.e. marked as "public"). Most political actors enable discussions (commenting) on their pages, which means other Facebook users are able to publicly respond to (post comments about) the original posts or to each other's responses. This constitutes a vast and vivid source of political or politics-inspired discussions, debates, expressions of sentiment, support or dissent, attended by and influencing large crowds of social media users – who also happen to be real-life voters.

In this use case of TrendMiner, RILMTA developed a set of tools and resources to enable the collection and analysis of Hungarian public Facebook comments written in response to public posts published on the pages of Hungarian politicians and political organizations. Besides the identification of relevant entities and sentiment polarity in these messages, our investigations focused on methods for detecting and quantifying several additional psychological and socio-psychological phenomena including *primordial vs. conceptual thinking, agency and communion, optimism/pessimism and individualism/collectivism*. In our hope, this will lead to better answers to questions such as: how can political actors shape public sentiment via their public SM messages? What are the trends in the reactions to these messages for the various political actors? How do the trends in these online discussions correlate to real-life political actions and events, such as elections and votes? How do political communication and discussions shape the psychological states and social values of various SM user groups?

5.1 Data Sources

We identified 1341 different Facebook pages that belong to Hungarian political organizations (parties, their regional and associated branches etc.) and politicians (candidates and elected representatives, party officials, members of Parliament etc.) of years 2013 and 2014. We used both official pages (administered by the agents of the political actors the pages are about) and fan pages (administered by independent communities).

There were 3 major election campaigns in Hungary in 2014: general elections for seats in the National Assembly (Hungarian Parliament) in April, elections for seats in the European Parliament in May, and municipal elections in October. We therefore collected the names and Fb (Facebook) pages of:

- members of Hungarian Parliament in the 2010-2014 term
- candidates for the 2014 Hungarian Parliament elections (only individual electoral candidates (OEV))
- members of Hungarian Parliament for the 2014-2018 term
- candidates for the 2014 European Parliament Elections in Hungary
- winners of seats in the European Parliament in 2014 in Hungary

We did not collect data for the October 2014 municipal elections for lack of time (the project ended in October 2014). The sources used were *valasztas.hu* (official election data) and Hungarian Wikipedia. The data was also used to compile a database of Hungarian political actors in 2013-2014, which was imported to the TrendMiner Political Ontology (See section 1.3 in Deliverable D2.4).

Facebook Graph API was used to collect public posts and their public comments from these sources on a regular basis (once a week). One week after each harvest, another script was used to check on new comments that arrived to already downloaded posts. Posts and comments dated in the period 01.10.2013 – 02.09.2014 were collected. The Table 10 summarizes the size properties of the data set.

For each downloaded post, the following metadata was stored: ids of the containing page, id of the post, date and time of creation, message text, author user id, number of shares, download date and time. For each downloaded comment, the following metadata was stored: ids of the containing page, the post it was written in response to, id of the comment it was written in response to, id of the comment, date and time of creation, message text, author user id, date and time of download. For all data sources (Facebook pages) the following data was stored: id of the Fb page, URL, title of the page, type of the entity the page is about (person or organization), normalized name of the entity the page is about, normalized name of the party the entity is associated with, participation in the 2010 and 2014 Hungarian and European elections (for persons, also the normalized name of the party that nominated the person, and the results of the election (won the seat or not)).

Facebook pages monitored	1,341
posts collected	141,825
comments collected	1,939,356
comment authors	226,500
sentences in comments	4,079,339
tokens in comments	46,211,723
average comment length (tokens)	23.83

Table 10: Size properties of Facebook collection

All this data (together with annotations and scores described in Section 5.3 is organized in a MySQL database which allows unified storage and the formulation of arbitrary queries for future analyses. The database schema is illustrated in Figure 6.[27]

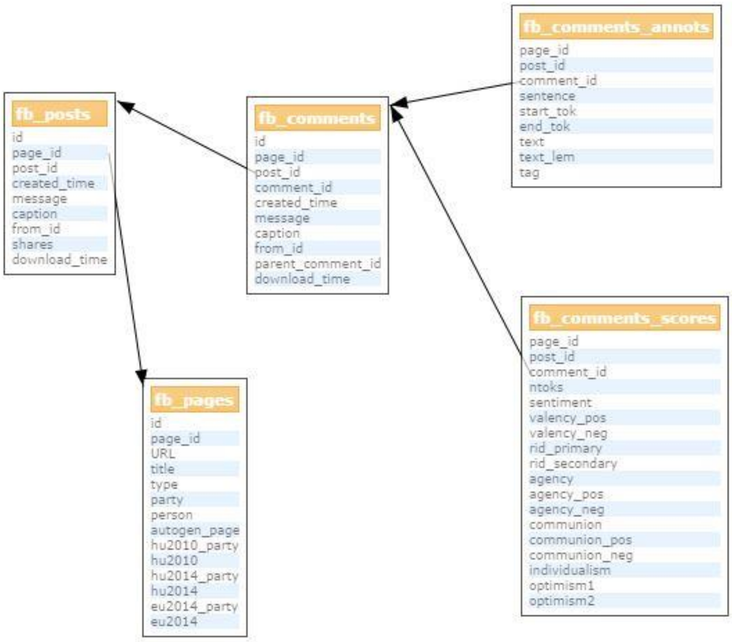


Figure 6: Database schema

5.2 Processing Pipeline

[27] Mapping of this database schema to ontology languages is on the way as well.

Each incoming Facebook comment, after being saved to the database is exported and processed by the following pipeline (in blocks of 500 to enable batch parallel processing):

1. Basic NLP
   - sentence segmentation and tokenizetion (*huntoken*)
   - morphological analysis (*hunmorph*)
   - part-of-speech tagging (*hunpos*)
   - lemma and morphological analysis disambiguation (own script)
   - named entity recognition (*Java NooJ*)

2. Content analysis
   - annotation of expressions of sentiment (emotional polarity) and 4 other sociopsychological dimensions (*Java NooJ*)

3. Processing results
   - calculating scores for document & storing scores and all annotations in database
   - Linking named entities to ontology entity URIs and uploading document to visualization server in RDF format

5.2.1 Domain-adaptation of the NLP tools

The fundamental steps in the NLP pipeline (tokenizaton, morphological analysis and PoS-tagging) were carried out by existing open source tools available in the *hun** package (http://mokk.bme.hu/eszkozok/). However, all of these tools were developed for a linguistic domain (using standard language texts, mostly newswire) that is different from the language used in Facebook comments. The latter has a high tendency for phenomena like typos and spelling mistakes, non-standard punctuation use, use of slang expressions, emoticons and other creative uses of characters, substitution of Hungarian accented characters by their Latin-1 codepage variants etc. For this reason, our readily available tools suffered from degradation in performance, and had to be adapted to the SM language domain.

To properly investigate the problems arising from switching from standard language to SM language, we created an **investigation corpus** of 1.2 million Facebook comments. The corpus was analyzed by the vanilla NLP tools, and unknown tokens were collected into frequency lists (see Table 11):

Comments (documents) in corpus	1,244,322
Sentences	2,504,514
Tokens	29,022,748
Unknown tokens	2,256,642
Unknown token types	694,764

Table 11: Characteristics of Facebook collection

Unknown types occurring with a frequency of 15 or higher (about 14,000 types) were manually inspected (with reference to their contexts). This investigation yielded a) an overview of common problems that had regularity, b) lists of unknown and important words and abbreviations, slang terms, name variants etc. In the following, we give account of all these observations and how they were accommodated into the domain-adaptation of the tools. The process involved both the implementation of methods to harmonize non-standard input to standard format and the extension of the tools to handle non-standard input.

On the level of **tokenization**, the following issues were resolved by adding pre- and post-processing filters to the *huntoken* tool:

- Missing spaces around punctuation were inserted, e.g.: *End of a sentence.Beginning of another sentence.* – here the unknown token "*sentence.Beginning"* would have been produced.
- Multiplicated punctuation, e.g. *part one…. part two* would be tokenized as: ("part", "one.", "…", "part", "two") – here the extra period from the ellipsis would be attached to the preceding word ("one.") causing the morphological analyzer to fail. Punctuation sequences of length > 3 were converted to length 1.
- Some slang words represent contractions of two or more standard language words, e.g. *asszem* = *azt hiszem*, these were normalized using a list.

- For some interjections and onomatopoeic words, it is common for SM authors to write with the final consonant multiplicated for added sentiment, e.g. *pfffffffff, uffffff, ejjjjjjj*. These were normalized with regular expressions.
- A list of commonly misspelled words and their corrected form was used to correct tokens that were otherwise known words for the morphological analyser.
- Some URLS were split into several tokens, corrected by a regular expression.
- Large numbers were split by decimal groups, e.g. 125 000, corrected by a regular expression.
- Emoticons were split into multiple tokens, e.g. ": D", corrected by using a list of emoticons.
- Some bugs in huntoken were fixed, e.g. in some cases it produced not well-formed XML output.

There were a large number of tokens in the SM sample corpus that were unknown to the *hunmorph* **morphological analyzer** tool, but which were not misspelled known words but existing valid words that are were not in the analyzer's lexicon. Items with high frequencies and relevance to the political use case were categorized into lists:

- emoticons
- abbreviations
- proper names
- slang words and expressions
- other unknown words (compounds etc.)

Since Hungarian is a highly inflectional language with a rich morphology, simply using a list of new word forms would not have been sufficient since each word lemma may appear in numerous inflected forms. Instead, for each unknown but correct proper name, slang word or other regular unknown word, another word was manually produced that was 1) already known for the analyzer (and not a derivational form), 2) had the same part-of-speech and subcategory, 3) was analogous to the unknown word in its inflection, i.e. it belonged to the same morphological paradigm. The desired part-of-speech of the unknown word–analogous word pair was also added manually. A tool called *hunmorph_appender* was developed to process this list and automatically extend the *hunmorph* analyzer's lexicon by looking up known analogous words with their parts of speeches in the lexicon and copying their morphological feature descriptions for the new entries. After recompiling *hunmorph*'s database with the added new items, *hunmorph* was able to recognize any possible (inflected) forms of the newly added words.

Another tool was developed to improve the linguistic quality of **lemmatization**. The *hunmorph* morphological analyzer produces a list of possible lemmas and corresponding analyses (codes that describe inflectional or derivational operations on the lemma that lead to the observed word form). This output is ambiguous: 1.94 analyses on average for all tokens based on a sample of 27,132,830 tokens with analyses, or 2.47 analyses on average for content words (based on 8,274,085 content word tokens with analyses)). On the other hand, the *hunpos* part-of-speech tagger tool is able to assign an unambiguous part-of-speech tag + inflectional information for each token using its statistical model. We use this information and a number of linguistically motivated heuristics (selecting minimal number of compounding and derivational operations, matching capitalization with surface form etc.) in a python script to choose a single morphological analysis and its corresponding lemma.

The source code of all the tools mentioned above that were developed to improve the quality of the standard language NLP tools is available from https://github.com/mmihaltz/trendminer-hunlp. Evaluation of the performance the domain-adaptation of the tools is presented in Section 5.5.1.

5.2.2 Named Entity Recognition

For the identification of relevant named entities (names of political actors: persons (politicians) and organizations (parties)) in the Facebook comments an experiment was conducted using *HunTag* (https://github.com/recski/HunTag/), an open-source maximum entropy classifier tool trained to resolve Hungarian named entities (Simon, 20013). However, HunTag showed a major performance loss when running on Facebook comments, probably because HunTag's NER model, like the other available standard Hungarian NLP tools, was trained on standard language newswire documents. Based on manual introspection, HunTag's output showed an unacceptable rate of , false positive NEs, entity type miscategorizations and even entity boundary recognition problems. A decision was made to use a domain-specific, lexicon-based NE recognition tool better suited for our purpose. A lexicon and grammar was created using the NooJ text analysis tool (see below) which is able to recognize the names (using lemmatization), name variations (e.g. first name and last name or just last name. abbreviation etc.) and common nicknames of the politicians and political parties identified for the domain of the project (Section 5.1). It is also used to annotate political actors with their party affiliations (e.g. name of the party that mandated a members of parliament etc.) For the development of the lexicon, the corpus described in the next section was employed.

5.3 Content Analysis

Scientific Narrative Psychology (SNP) is a complex approach to text- based theory building and longitudinal, quantitative assessment of psychological phenomena in the analysis of self-and group narratives in the field of social, personality and clinical psychology (László, 2008). The method of SNP, Narrative Psychological Content Analysis (NPCA), is integrated with Natural Language Processing (NLP) technology. Work for the socio-psychological content analysis of Facebook comments in the political use case was carried out in cooperation with researchers of the Narrative Psychology Research Group at the Institute Of Cognitive Neuroscience And Psychology, Hungarian Academy of Sciences. The content analysis modules developed for the political use case build on earlier work (Ehmann et al, 2012, Ehman et al 2014, László et 2013) and extend it to the domain of SM discourse in politics. This enhances current approaches to sentiment analysis in SM text analysis that only focus on one psychological aspect, emotional polarity.

For the content analysis of Facebook comments, the open source Java NooJ tool (http://nooj4nlp.net) was used. NooJ enables the creation of lexicons and grammars that can be compiled into finite state automatons usable for lexical, morphological and syntactic annotation and concordance generation. NooJ is able to import and rely on annotations produced by outside NLP tools. It was successfully employed for Hungarian psychological content analysis in the past (Ehmann et al, 2012, Ehman et al 2014, László et 2013).

Since NooJ and Java NooJ are both single-user tools operated through a graphical user interface, we used the API available in the source code of Java NooJ to create a command-line version of NooJ that is able to perform most features of the GUI. This way NooJ can be embedded in large-scale batch processing systems[28].

To facilitate the creation of custom lexicons and grammars for content analysis, a **sample corpus** was created that contains comments from 569 different Hungarian politics-related Facebook pages (summary in table below).

Comments in corpus	**176,398**
Tokens	5,458,497

The corpus was analyzed with the standard NLP tools. Frequency lists of word forms, lemmas, lemmas with PoS-tags, named entities etc. were created. They were used to aid the creation of the named entity, emotional polarity and agency-communion lexicons (details in the following sections).

In the following sections, all the content analysis NooJ modules are described in detail.

5.3.1 Emotional Polarity

This module performs the annotation of emotions and their polarities (positive or negative; also known as emotional valency in the psychology literature) that can be used to calculate the general sentiment value of a comment. The grammar uses a lexicon of 500 positive and 420 negative nouns, verbs, adjectives, adverbs, emoticons and multi-word expressions. It also uses a number of rules to treat elements of context that might affect polarity (e.g. negation). The lexicons were constructed by employing 6 independent human annotators to code words in the sample corpus that occurred with a frequency of 100 or more (about 3500) for polarity. In the cases where at least 4 annotators agreed, a seventh annotator made the final decision.

[28] The source code for nooj-cmd is available from http://github.com/tkb-/nooj-cmd, the compiled binary is downloadable from http://bitbucket.org/tkb-/nooj-cmd/downloads.

Next, an evaluation corpus was created manually to check and improve the quality of the annotations produced by the polarity grammar. A collection of 337 facebook comments were each annotated by 3 independent human annotators for positive and negative polarity expressions, and a 4th annotator made final judgements. The emotional polarity NooJ module was used to annotate the same documents automatically, and the results were compared to the manual annotation. Based on this, a number of improvements were made to the lexicon and the rules, and a list of frequent misspelled word forms was created to be used with the adapted tokenizer (See Section 5.2.1).

5.3.2 Agency and Communion

According to current social theory, there are 2 fundamental dimensions in social values:

- *Communion* describes the moral and emotional aspect of an individual’s relations to other group members, individuals or groups. Expressions of cooperation, social benefit, honesty, self-sacrifice etc. (moral aspects) and affection, friendship, respect, love etc. (emotional aspect).
- *Agency* is the efficiency of an individual’s goal-orientated behavior in terms of their personal goals: motivation (ambition, persistence, sense of purpose, will etc.), competence (intelligence, skills, cunning, professional qualities etc.) and control (assertivity, success, power etc.)

The agency/communion NooJ grammar is responsible for annotating expressions of agency or communion in the texts of comments. Both dimensions can further be divided into positive or negative values. The module’s lexicon was created by coding words with a frequency of at least 100 (about 3500 words) for agency or communion by 3 + 3 independent judges. A seventh annotator made final decisions about inclusion in cases of 100% agreement. The lexicon contains 650 different expressions with morphological properties.

5.3.3 Regressive Imagery Dictionary

The Regressive Imagery Dictionary (RID) is a classic content analysis lexicon (Martindale, 1990) created to uncover psychological processes reflected in the text. The underlying theory proposes 2 basic categories of thinking:

- Primordial (primary): associative, concrete, and takes little account of reality, found in fantasy, dreams, reverie
- Conceptual (secondary): abstract, logical, reality oriented, aimed at problem solving

The Regressive Imagery Dictionary for English contains about 3000 words divided into 29 categories designed to measure primordial content and another set of 7 categories designed to measure conceptual thought. The Hungarian version was created by carefully localizing English concepts and implementing it as a NooJ grammar (Pólya and Szász, 2013).

5.3.4 Optimism-Pessimism

Individuals differ in the way past, present or future events dominate their thinking. When a person's thinking is dominated by the past, they are likely to view the world unchangeable. Thinking dominated by the present indicates the importance of realistically attainable goals, while future-dominated thinking usually sees open possibilities. The NooJ grammar for optimism-pessimism annotates expressions of time using morphological information (verb tenses) and by recognizing some temporal expressions. Based on these, 2 variants of an optimism indicator can be calculated (higher values, higher degrees of optimism):

- |future_tense_verbs| / |past_tense_verbs|
- |present_tense_verbs| / |past_tense_verbs|

5.3.5 Individualism-collectivism

Individualism represents the importance of the category of a person when thinking about the world. The level of individualism is indicated by the use of personal pronouns. Frequent use of pronouns shows the importance of the category of the person, this is what's prominent, which indicates a high level of individualism. Lower frequency of personal pronouns signals that the category of person is more in the background; in this case the environment is in the foreground, which means that the level of individualism is lower. This NooJ grammar relies on part-of-speech and morphological information only to annotate personal pronouns and verbs or nouns with personal inflections. Counting these and calculating the ratio between the former and the latter provides a measure of individualism. A higher score indicates higher degree of individualism.

5.4 Processing annotations

After the general NLP, named entity and psychological content annotations are complete, each annotated document is processed:

- all annotated text segments are stored in the database, with reference to the document (page, post, comment) id, the index of the containing sentence, starting and ending token positions and the annotation tag. The annotated text is stored both as it was in the original document and both with the head word (last token) lemmatized.
- The counts for the different content annotation types and some derived measures are stored for each document together with the document length (token count) to enable normalization:
    - sentiment score = (| positive_valency | - | negative_valency |) / token_count
    - positive sentiment annotation count
    - negative sentiment annotation count
    - RID primary annotation count
    - RID secondary annotation count
    - agency score = (| positive_agency | - | negative_agency |) / token_count
    - positive agency annotation count
    - negative agency annotation count
    - communion score = (| positive_communion | - | negative_communion |) / token_count
    - positive communion annotation count
    - negative communion annotation count

- individualism score = | personal_pronouns | / (|nouns_poss_infl| + | verbs_pers_infl |)
- optimism score 1 = |future_tense_verbs| / |past_tense_verbs|
- optimism score 2 = |present_tense_verbs| / |past_tense_verbs|

5.5 Evaluation

5.5.1 Evaluation of Domain-adaptation

To assess the quality of the adaptation of our NLP tool to the social media domain, we carried out a simple procedure. A corpus of comments was assembled for evaluation that was disjunct from (had none of the comments used in) the development corpus described in Section 5.2.1. It was processed by both the original (standard text) and the adapted (social media language compatible) versions of our basic NLP tools (tokenizer, PoS-tagger, morphological analyzer and lemma/analysis disambiguator), and the number of unknown tokens (no lemma or morphological information) was counted. We expected the adaptation to show a decrease in the number of unknowns, which would enable higher theoretical recall rates for the successive annotation levels. The findings are summarized in Table 12.

	With original tools	With adapted tools
Comments (documents) in corpus	457,059	
Sentences	941,106	
Tokens	10,687,624	
Unknown tokens	1,378,703 (12.9%)	9,415,796 (8.81%)
Unknown token types	308,610	306,502

Table 12: Corpora used for and results of the evaluation of the domain adaptation of the Hungarian NLP tools to Facebook language

5.5.2 Named Entity Recognition and Psychological Annotation Quality

In order to evaluate the quality of the annotation tools created for the use case, manually annotated sets were created to function as gold standards to compare against automatic annotation. Three different sets were created for the different annotation tasks (see Table 13 below). Each set was created to reflect the distribution of comments for each political party that is observed in the complete 1.9M-comment 1-

year use case corpus: FIDESZ-KDNP 25.2%, EGYÜTT-2014 19.3%, JOBBIK 19.2%, MSZP 16.6%, DK 12.5%, PM 4.2%, LMP 2.9%.

	Set 1	**Set 2**	**Set 3**
Annotation level(s)	NER	Emotional polarity	Agency-communion, Optimism-pessimism, Individualism-collectivism
Number of comments	337	336	672
Number of tokens	7,615	7,540	17,924

Table 13: The three gold standard sets used for the evaluation of named entity and psychological annotation

Named Entity Recognition

The gold standard set consists of 337 comments (7,615 tokens). For each comment, 7 binary decisions were made by the human annotators which show whether or not a named entity that corresponds to one of 7 political parties in this Use Case (FIDESZ-KDNP, EGYÜTT-2014, JOBBIK, MSZP, DK, PM, LMP) is present in the comment's text (at least once). (This method was chosen instead of tagging entity occurences in text for efficiency reasons.) Coding was carried out by 2+1 annotators for each comment (the third annotator was the judge in situations where the first two disagreed). The NER NooJ module was used to automatically annotate this set. The following indicators were calculated and summed for each comment:

- True positives (tp): number of (comment_id, party_name) pairs in the machine output that were also present in the gold standard
- False positives (fp): number of (comment_id, party_name) pairs in the machine output that were not present in the gold standard
- False negatives (fn): number of (comment_id, party_name) pairs in the gold standard that were not present in the machine output

Precision, recall and F1 score were calculated based on these, shown in Table 14.

TP	FP	FN	Precision	Recall	F1 score
60	1	45	0.9836	0.5714	0.7229

Table 14: Precision, recall and F1 score of named entity (party affiliation) recognition (TP: true positive, FP: false positive, FN: false negatives)

Emotional Polarity (Sentiment)

The gold standard set for the evaluation of emotional polarity (sentiment) annotation consists of 336 comments (7,540 tokens). The comment messages were segmented into clauses by a simple algorithm (splitting sentences along the following punctuation characters: \n|\.|,|:|;|\?|!|\–), yielding 1295 clauses. Each clause was annotated by 3 human annotators with 2 integer values: number of positive and number of negative expressions in the text. The Emotional Polarity NooJ module was used to automatically annotate the clauses, and the number of positive and negative annotations were counted in each clause. Then, the following indicators were calculated and summed for each clause, separately for the positive and the negative annotations (gs: count of positive/negative annotations in the gold standard, mo: count of positive/negative annotations in the machine output):

- True positives (tp): if gs >= mo: tp = mo, if gs < mo: tp = gs
- False positives (fp): if gs >= mo: fp = 0, if gs < mo: fp = mo-gs
- False negatives (fn): if gs <= mo: fn = 0, if gs > mo: fn = gs-mo

Precision, recall and F1 score were calculated based on these indicators for positive annotations, negative annotations and combined (Table 15). It should be noted that comparing only the counts of annotations instead of the actual annotated text segments is prone to prone some error (e.g. when the count of annotations is correct for a text unit but the machine marked different text segments than the gold standard). However, on the one hand the text units for which annotation counts were compared were short (clauses), so such errors are unlikely; and on the other hand, sentiment score calculation –which is the ultimate goal of emotional polarity annotation-- is based only on the counts of annotations.

	TP	FP	FN	Precision	Recall	F1 score
positive	142	30	53	0.8256	0.7450	0.7738
negative	124	61	107	0.6703	0.5368	0.5962
combined	266	91	160	0.7451	0.6244	0.6794

Table 15: precision, recall and F1 score of positive and negative emotional polarity annotation

Besides the quality assessment of emotional polarity annotations, we were also interested in how accurately sentiment can be measured based on these annotations. A score was calculated in the gold standard and in the machine output for each clause by the following formula which represents only the polarity (but not the intensity) of sentiment:

-1 if | positive_expressions | - | negative_expressions | < 0
score = 1 if | positive_expressions | - | negative_expressions | > 0
0 if | positive_expressions | - | negative_expressions | == 0

Accuracy was calculated by dividing the number of clauses for which the score was equal for both machine output and gold standard by the number of all clauses (Table 16)

Clauses with correct sentiment polarity	All clauses	Accuracy
1096	1295	0.8463

Table 16: Accuracy of sentiment polarity recognition

Agency and Communion

The gold standard consisting of 672 comments was split into 3188 clauses (see previous section) and each was annotated with 4 binary values indicating the presence or absence of four classes in the clause (positive agency, negative agency, positive communion, negative communion). 3 annotators coded for agency, 3 other annotators coded for communion. Automatic annotations by the agency-communion NooJ grammar were converted to binary values and compared to the gold standard and evaluated as a binary classification problem for each class (Table 17).

	TP	FP	FN	Precision	Recall	F1
Agency: positive	84	35	115	0.7059	0.6943	0.5283
Agency: Negative	25	13	73	0.6579	0.2551	0.3676
Agency combined	109	48	188	0.6943	0.3670	0.4802
Communion: positive	48	25	307	0.6575	0.8205	0.2243
Communion: negative	80	3	500	0.9639	0.1380	0.2413
Communion combined	128	28	807	0.8205	0.1369	0.2347

Table 17: precision, recall and F1 score for positive and negative agency and communion annotations

Like emotional polarity, agency and communion both have polarity dimensions from which a score can be calculated which represents polarity (values -1, 0, 1, see previous section). Accuracy was calculated for agency and communion polarity recognition by dividing the number of clauses with correct polarity compared to the gold standard by the total number of clauses in the whole dataset (Table 18).

	Clauses with correct sentiment polarity	All clauses	Accuracy
Agency	2954	3188	0.9265
Communion	2661	3188	0.8347

Table 18: Accuracy of agency and communion polarity recognition

Optimism and Pessimism

For the evaluation of the optimism/pessimism module the same set of comments was used as for agency/communion evaluation (672 comments divided into 3188 clauses). 3 human annotators tagged subsets of the corpus with binary flags corresponding to the presence or absence of each of the linguistic markers used for calculating the optimism scores: past tense verbs, non-past tense verbs (present tense) and future tense verbs. Table 19 summarizes precision, recall and F1 measures from the evaluation of the automatic annotation against the gold standard set.

Marker	Annotated GS clauses	TP	FP	TN	FN	Precision	Recall	F1
Verb_past	1799	187	50	1545	12	0.7890	0.9397	0.8578
Verb_present	1850	248	542	991	20	0.3140	0.9254	0.4688
Verb_future	3188	61	125	2971	30	0.3280	0.6703	0.4404

Table 19: Precision, recall and F1 score of the annotation of linguistic markers used for optimism/pessimism score calculation

Individualism and Collectivism

The evaluation of the individualism/collectivism module used the same corpus (3188 clauses) and same evaluation methodology as the evaluation of the optimism/pessimism module. Subsets were annotated for the presence/absence of linguistic markers: personal pronouns (PP) and nouns or verbs with personal inflection suffixes (SUFF). Table 20 contains the results of comparing automatic annotations to the gold standard set.

Marker	Annotated GS clauses	TP	FP	TN	FN	Precision	Recall	F1
PP	2099	63	116	1885	33	0.3520	0.6563	0.4582
SUFF	797	306	90	343	17	0.7727	0.9474	0.8512

Table 20: **Precision, recall and F1 score of linguistic markers used in the individualism/collectivism module (PP: personal pronouns, SUFF: verbs and nouns with personal inflections)**

6. Relevance to TrendMiner

Relevance to the project objectives

Data described in D10.1 are crucial for several scientific and technological objectives of the project, i.e.: Delivered lexical and linguistic processors integrated in new real-time trend mining and summarisation methods for steam media developed by Trendminer, i.e, the political and health use cases provide means of tracking and monitoring of events and entities over different social media (Twitter, Facebook, blogs, forums) in different languages (Polish, Hungarian, Spanish)

Relation to other workpackages of TrendMiner

D10.1 provides content related to the following TrendMiner workpackages:

- **WP5 (See D5.5: Deployment of Web services for new cases** provides information about Web services making use of resources and NLP tools described in D10.1. This deliverable describes the integration of the use cases by the new partners)

- **WP9 (See D9.1: Integration of annotation generated by annotation tools** contains detailed information of annotated collections using lexical and NLP tools described in D10.1. This deliverable describes the integration of annotations generated by the new partners)

- **WP2 (See D2.4: Integration of all lexical and terminological data in the TRENDMINER platform** contains information on project ontologies and terminological data cross- and multilingually harmonized within D10.1. This deliverable describes how the lexical and terminological data from the new partners are integrated in the ontological and terminological framework of TrendMiner)

Bibliography and references

Acedański, S. (2010). A morphosyntactic Brill tagger for inflectional languages. In Advances in Natural Language Processing, pp. 3–14.

Buczyński A. (2006). Wybrane zastosowania programu Kolokacje do badań lingwistycznych. W: A. Duszak, E. Gajel, U. Okulska, Korpusy w angielsko-polskim językoznawstwie kontrastywnym. Teoria i Praktyka. Kraków 2006, pp. 427-448.

Buczyński A. and Wawer A. (2008). Automated classification of product review sentiments in Polish. In Kłopotek M.A., Przepiórkowski A., Wierzchoń S.T., and Trojanowski K., (eds.), Intelligent Information Systems, pp. 211–215, Warsaw. Akademicka Oficyna Wydawnicza EXIT.

CA. Bond and Cynthia L. Raehl. 2006. Adverse drug reactions in United States hospitals. Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy, 26(5):601-608

Cornelis S. van Der Hooft, Miriam CJM Sturkenboom, Kees van Grootheest, Herre J. Kingma and Bruno HCh Stricker. 2006. Adverse drug reaction-related hospitalisations. Drug Safety, 29(2):161-168

Ehmann B., P. Lendvai, T. Pólya, O. Vincze, M. Miháltz, L. Tihanyi, T. Váradi, J. László 2012, Narrative Psychological Application of Semantic Role Labeling. In: Kristina Vučković, Božo Bekavac, Max Silberztein (eds.): Automatic Processing of Various Levels of Linguistic Phenomena: Selected Papers from the NooJ 2011 International Conference, Pages 195-204, Dubrovnik, Croatia, Cambridge Scholars Publishing, Faculty of Humanities and Social Sciences, Cambridge, 3/2012

Ehmann Bea, Csertő István, Ferenczhalmy Réka, Fülöp Éva, Hargitai Rita, Kővágó Pál, Pólya Tibor, Szalai Katalin, Vincze Orsolya, László János: Narratív kategoriális tartalomelemzés: a NARRCAT. In: Tanács Attila, Varga Viktor, Vincze Veronika (szerk.) X. Magyar Számítógépes Nyelvészeti Konferencia: MSZNY 2014. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2014. pp. 136-147.

Hare J. S, Samangooei S., and Dupplaw D. P. (2011). Openimaj and im- ageterrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In Proceedings of the 19th ACM international conference on Multimedia, MM ’11, pages 691–694, New York, NY, USA. ACM.

Kaleta P. (2010). Ludzie Władzy Polski Niepodległej 1989-2009, Zabrze.

Karin Wester, Anna K. Jönsson, Olav Spigset, Henrik Druid and Staffan Hägg. 2008. Incidence of fatal adverse drug reactions: a population based study. British journal of clinical pharmacology, 65(4):573-579

Kopeć M. (2014). Zero subject detection for Polish. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers, pp. 221–225, Gothenburg, Sweden. Association for Computational Linguistics.

László János, Csertő István, Fülöp Éva, Ferenczhalmy Réka, Hargitai Rita, Lendvai Piroska, Péley Bernadette, Pólya Tibor, Szalai Katalin, Vincze Orsolya, Ehmann Bea

László, J. 2008, The Science of Stories: An introduction to Narrative Psychology. London, New York: Routledge.

Lui M. and Baldwin T. (2012). Langid.py: An off -the-shelf language identification tool. In Proceedings of the ACL 2012 System Demonstrations, ACL '12, pp. 25–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. (Eds.): Proceedings of the 2007 International NooJ Conference, 214-227

Martindale, C. (1990). The clockwork muse: The predictability of artistic change. New York: Basic Books.

Miłkowski M. (2010). Developing an open-source, rule-based proofreading tool. Software: Practice and Experience, 40(7):543–566.

Narrative Language as an Expression of Individual and Group Identity: The Narrative Categorical Content Analysis. SAGE OPEN 3:(2) Paper 2158244013492084. 12 p. (2013)

O'Connor B., Krieger M., and Ahn D. (2010). TweetMotif: Exploratory Search and Topic Summarization for Twitter. In William W. Cohen, Samuel Gosling, William W. Cohen, and Samuel Gosling (eds.), ICWSM. The AAAI Press.

Ogrodniczuk M., Lenart M. (2012). Web Service integration platform for Polish linguistic resources. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 1164–1168, Istanbul, Turkey. ELRA.

Pólya T., Szász L.: A Regresszív Képzeleti Szótár magyar nyelvű változatának kidolgozása. IX. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2013) 124–132.

Porter M. (1980). An algorithm for suffix stripping. Program, 14(3):130–137.

Przepiórkowski A., and Buczyński A. (2007). Spejd: Shallow Parsing and Disambiguation Engine. In Zygmunt Vetulani (ed.), Proceedings of the 3rd Language & Technology Conference, pp. 340–344, Poznań, Poland.

Przepiórkowski A., Bańko M., Górski R. L., and Lewandowska-Tomaszczyk B. (2012). Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw.

Radziszewski, A. (2013). A tiered CRF tagger for Polish. In H. Rybiński, M. Kryszkiewicz, M. Niezgódka, R. Bembenik, and Ł. Skonieczny (eds.), Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions. Springer Verlag.

Radziszewski, A. and Śniatowski, T. (2011). A Memory-Based Tagger for Polish. In Proceedings of the LTC 2011. Tagger available at http://nlp.pwr.wroc.pl/redmine/projects/wmbt/wiki/.

Segura-Bedmar I, Revert R, Martínez P. (2014a) Detecting drugs and adverse events from Spanish social media streams. In *Proceedings of LOUHI 2014, ACL, 106-115.*

Segura-Bedmar I, Peña-González S, Martínez P (2014b): Extracting drug indications and adverse drug reactions from Spanish health social media. In Proceedings of BioNLP 2014, 98-106.

Segura-Bedmar I, Revert R, Martínez P., Moreno-Schneider, J (2014c). Exploring Spanish Health Social Media for detecting drug effects, BMC Medical Informatics and Decision Systems, accepted manuscript.

Silberztein, M. 2008, Complex Annotations with NooJ. In: Blanco, X. and Silberztein, Simon, E. (2013): Approaches to Hungarian Named Entity Recognition. PhD dissertation. Budapest University of Technology and Economics, Budapest, 2013.

Waszczuk J., Głowińska K., Savary A., Przepiórkowski A., and Lenart M. (2013). Annotation tools for syntax and named entities in the National Corpus of Polish. International Journal of Data Mining, Modelling and Management, 5(2):103–122.

Waszczuk, J. (2012). Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language. In Proceedings of the 24th International Conference on Computational Linguistics (COLING2012), pp. 2789–2804, Mumbai, India.