D4.1 PRELIMINARY VERSION OF THE TEXT ANALYSIS COMPONENT, INCLUDING: NER, EVENT DETECTION AND SENTIMENT ANALYSIS

Grant Agreement nr	611057
Project acronym	EUMSSI
Start date of project (dur.)	December 1st 2013 (36 months)
Document due Date :	November 30th 2014 (+60 days) (12 months)
Actual date of delivery	December 2nd 2014
Leader	GFAI
Reply to	susannep@iai.uni-sb.de
Document status	Submitted

Project co-funded by ICT-7th Framework Programme from the European Commission

Project ref. no.	611057
Project acronym	EUMSSI
Project full title	Event Understanding through Multimodal Social Stream Interpretation
Document name	EUMSSI_D4.1_Preliminary version of the text analysis component.pdf
Security (distribution level)	PU – Public
Contractual date of delivery	November 30th 2014 (+60 days)
Actual date of delivery	December 2nd 2014
Deliverable name	Preliminary Version of the Text Analysis Component, including: NER, event detection and sentiment analysis
Type	P – Prototype
Status	Submitted
Version number	v1
Number of pages	60
WP /Task responsible	GFAI / GFAI & UPF
Author(s)	Susanne Preuss (GFAI), Maite Melero (UPF)
Other contributors	Mahmoud Gindiyeh (GFAI), Eelco Herder (LUH), Giang Tran Binh (LUH), Jens Grivolla (UPF)
EC Project Officer	Mrs. Aleksandra WESOLOWSKA Aleksandra.WESOLOWSKA@ec.europa.eu
Abstract	The deliverable reports on the resources and tools that have been gathered and installed for the preliminary version of the text analysis component
Keywords	Text analysis component, Named Entity Recognition, Named Entity Linking, Keyphrase Extraction, Relation Extraction, Topic modelling, Sentiment Analysis
Circulated to partners	Yes
Peer review completed	Yes
Peer-reviewed by	Eelco Herder (L3S)
Coordinator approval	Yes

Table of Contents

Tables

Figures

1. BACKGROUND

Deliverable D4.1 is described as follows, in the DoW:

A preliminary version of the text analysis component for integration in the multimodal platform, including preliminary versions of (1) a NER system that is capable of detecting names of all sorts in all text sorts and all languages the project deals with, (2) a component detecting / extracting events, relations and topics, for four languages and (3) a sentiment analysis module, for four languages.

D4.1 is a running prototype of the text analysis component to be integrated in the multimodal EUMSSI platform. This prototype is the result of the work performed within tasks 4.1, 4.2 and 4.3, during the first 12 months.

According to the DoW, Task 4.1 aims at *implementing and testing a NER system that is capable of detecting names in free text in English, French, Spanish and German.* The goal of task 4.2 is the *implementation of a processing component capable of detecting and extracting events, through the detection and extraction of topic, keyphrases and relations.* Task 4.3 aims at *implementing and testing a sentiment analysis module.*

Within the EUMSSI project, the output of the text analysis component (WP4) feeds into the cross-modal semantic representation framework (WP5; LUH, VSN, UPF) which serves as the basis for the contextualization and recommendation tools (WP6; UPF) whose specifications are guided by the user specifications developed in WP2.1 (DW, LUH, VSN). The text analysis component takes as input news-related data such as news articles provided by Deutsche Welle (DW), or news articles crawled from the web by LUH (WP5). Other text types such as the output of OCR (Optical Character Recognition) and ASR (Automatic Speech Recognition) are provided by IDIAP and LIUM respectively (WP3). Processing pipelines and annotation data formats are determined in consultation with all partners (WP2.2 System architecture, WP2.3 Data infrastructure and representation definition).

D4.1 is complemented by deliverable D4.2 which describes a preliminary version of the content extraction and analysis module for social media content.

D4.1 is followed by two further development stages: D4.3 in month 24 is the first functional version of the text analysis component, D4.5 in month 32 is the final version of the text analysis component.

2. INTRODUCTION

Within the EUMSSI architecture, the main task of the text analysis component is to extract structured information from unstructured text sources, that will be semantically enriched and aggregated to the multimodal interpretation platform, developed in WP5. The enriched semantic representation will then be used to build the two demonstrators in WP6.

As specified in the DoW, the main goal of the text analysis component in EUMSSI is to provide a text mining system able to extract named entities (persons, locations, organisations and temporal expressions), relations (or events), topics and opinions. The DoW also specifies that "*the implementation of this part consists mainly in taking available tools - owned by project partners or available from public domain - and enhance them with new features and domain adaptation*."

In the first year, a first version of a working text analysis module has been built by putting into place the core text analysis components such as named entity recognition (NER), named entity linking (NEL) and key phrase extraction, as well as a preliminary sentiment analysis component. The focus was on creating information structures that fulfill two purposes:

1. They can be used to improve the cross-media analysis results, beyond the potentials of the individual modules.
2. They serve as input for the contextualization and recommendation functionalities of the EUMSSI platform and demonstrators.

In this vein, initial experiments have been conducted with different text types such as (i) regular edited text, (ii) the output of optical character recognition (OCR), (iii) the output of automatic speech recognition (ASR) and (iv) user-generated text from social media. For sentiment analysis lexical resources have been gathered for inclusion into the system. Fine-tuning of models, enhancement of resources and language coverage is the goal of later project phases.

2.1. Requirements for tools used in EUMSSI

Given the multilingual and multimodal nature of EUMSSI and the architectural choices (use of UIMA), the tools used for text analysis have to fulfill several requirements:

1. *Language availability:* tools for all four EUMSSI languages (DE, EN, ES, FR) are needed. Thus, there are two options:
   a) One tool is available for all four languages or it can be trained for all four languages
   b) Different tools are used for different languages

   Generally the first option is preferred.
2. *Rich output features:* Tools that provide rich output features in the form of n-best candidate lists and scores are preferred since the output features can be used by the inference mechanisms for cross-media enhancement of the text analysis results and serve as input for the contextualisation and recommendation system.
3. *Rich parameter settings:* Tools with rich parameter settings that can be fine-tuned for different purposes are preferred.

4. *Program availability:* a downloadable version of the tool should exist, because web services do not guarantee continuous availability in the future.
5. *Integration into UIMA:* The text analysis is directly integrated into UIMA, which is written in Java. DKPro is a platform that provides many open-source UIMA-conformant tools for NLP (see 4.2). Hence, tools that already have a wrapper for DKPro or are written in Java are preferred.
6. *Adaptation to different text types:* ASR provides lower-cased output without any punctuation marks, OCR provides short, mostly sub-sentential text strings, social media streams provide highly colloquial and idiosyncratic text. Tools that can be applied to or trained for various text types are preferred.
7. *Output quality:* reasonable to excellent output quality is expected.

Given these requirements, the most suitable tools are selected.

2.2. Summary of the first year results

In this first year a survey of available tools has been performed and the following tools have been selected so far, according to the requirements expressed above: Stanford NER for Named Entity Recognition (NER), DBpediaSpotlight for named entity disambiguation (NED) and linking (NEL), and KEA for key phrase extraction. Stanford NER, DBpediaSpotlight and KEA key phrase extraction have been set up for all four EUMSSI languages and wrappers for integration into DKPro have been written if not available yet.
DBpedia Spotlight and OCR provide as output n-best candidate lists with scores, which can be used for cross-media and cross-tool enhancement. KEA also provides scores that can be used by the contextualisation and recommendation components.
For sentiment analysis an initial detector for polarity has been set up, and a benchmarking of existing lexical resources has been carried out.

As described in section 4.3.2, currently the text analysis processing chain contains segmentation/tokenization, lemmatization, part of speech tagging, NER, NED, NEL, key phrase extraction and detection of polarity. For ASR a special processing chain is used that trueCases the output of ASR which is in lower case letters. In a first approximation, StanfordNER is used as TrueCaser.

In the following sections, we address: description of the text corpus, named entity recognition, named entity disambiguation and linking, extraction of events, relations and topics via keyphrase extraction, sentiment analysis, analysis of other text types such as the output of OCR and ASR, the architecture of the text analysis component, and, finally, conclusions and next steps.

3. TEXT CORPUS DESCRIPTION

In addition to the multimedia corpus provided by Deutsche Welle (DW), corpora from major newspapers in the four languages covered by the project (*The Guardian, El País, Le Monde* and *Die Zeit*) and social media outlets (*You Tube, Twitter*) have been crawled, with a focus on the topic of fracking and energy transition in general. These corpora are used both for model training and as potential search space for the EUMSSI contextualization and recommendation system. Data crawling will be directly plugged into the platform so that collections are always up-to-date.
From the DW data, only articles with CLOBTEXT have been used. The Automatic Speech Recognition (ASR) data has been provided by LIUM, the Optical Character Recognition (OCR) data by IDIAP.

	EN		DE		FR		ES	
	Ori	Clean	Ori	Clean	Ori	Clean	Ori	Clean
DW	574.0	245.0	812.0	415.0	52.0	31.0	255.0	144.0
ASR output		3.6						
OCR output		~2500 samples		~2100 samples				

Table 1: Data size per source and language in MB and sample number

For *The Guardian*, *Die Zeit*, *Le Monde* and *El País* a total of 179.5MB have been downloaded so far. The data contains the following number of news articles:

newspaper	Number of articles
The Guardian (EN)	844
Die Zeit (DE)	536
Le Monde (FR)	112
El Pais (ES)	449

Table 2: Number of articles crawled from different newspapers

The news articles are collected from the corresponding newspapers' newsfeeds using a crawler that has been running since June 2014. The newsfeeds largely contain very recent article, with a small number of older articles in less active feeds. The distribution of news articles per month is as follows:

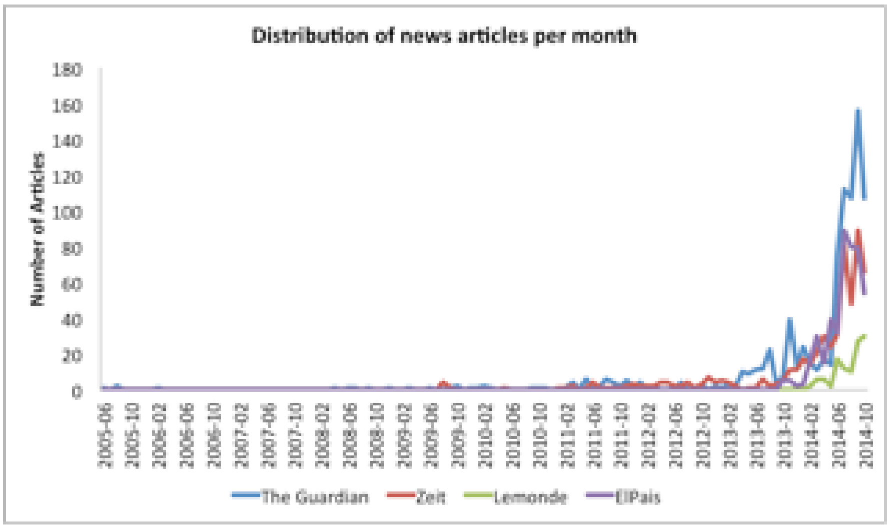


Figure 1: Distribution of new articles per month

The distribution of news articles by sources is as follows:

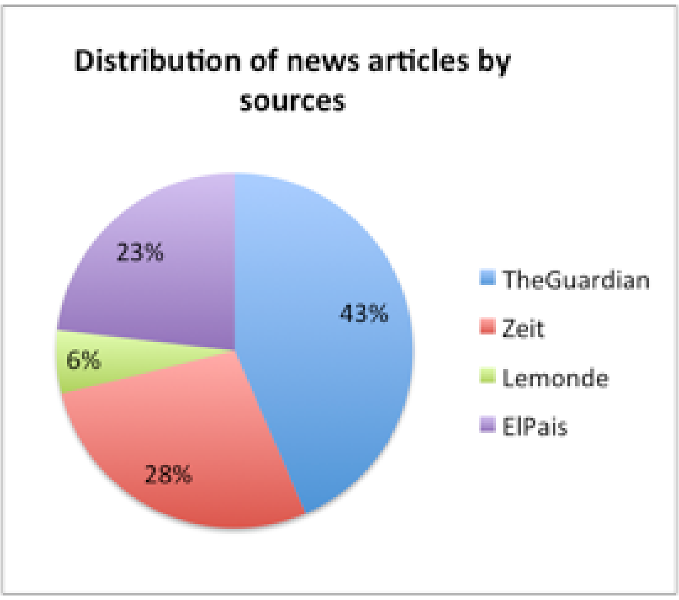


Figure 2: Distribution of news articles by sources

For more figures on the DW corpus see also deliverable D2.3.

3.1. Text cleaning

The text analysis tools expect clean text as input. Since the different news sources use different markup, text cleaning routines have to be written for each news source with additional attention to possible variations for different languages. In a first step, basic text cleaning routines have been set up that simply remove all markup thus yielding an unstructured text body. (The title information is commonly stored in a separate text field). In a second step, more advanced text cleaning routines are developed that preserve paragraph structure and recognise subtitles/subheaders within the text body by inserting linebreaks. The internal structure of the text body and the subtitle information can be used when analysing longer texts and when visualizing results. The recognition of subtitles also improves key phrase assignment of the linguistically based GFAI tool (see section 6.2.5).

For removal of all markup the jericho parser[1] is used. For recognition of paragraph structure and subtitles, perl scripts are developped. Currently, a perl script has been written that recognizes paragraph structure and subtitles/subheaders within the text body of the DW data for all four EUMSSI languages. The DW text body is marked by the meta tag CLOBTEXT. In the DW markup, the main tags for paragraph and their frequencies are:

Frequency	Tag
376628	<p>
382075	<P>
1433894	</p>
3579	</P>

Table 3: Paragraph tags used in DW corpus and their frequency

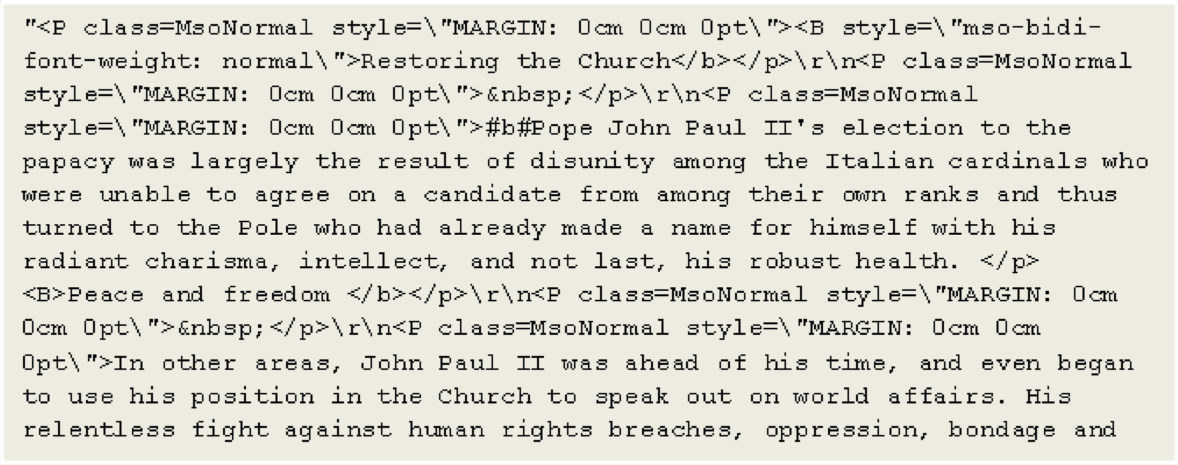
The main tags for subheaders/subtitles and their frequencies are:

Frequency	Tag
18	<B style=\"mso-bidi-font-weight: normal\" minmax_bound=\"true\">
27418	<B style=\"mso-bidi-font-weight: normal\">
222	<b style=\"mso-bidi-font-weight: normal\">
19144	<b>
54912	<B>
52	<STRONG minmax_bound=\"true\">
34274	<strong>
59420	<STRONG>

Table 4: Subheader tags used in DW corpus and their frequencies

[1] http://jericho.htmlparser.net/docs/index.html

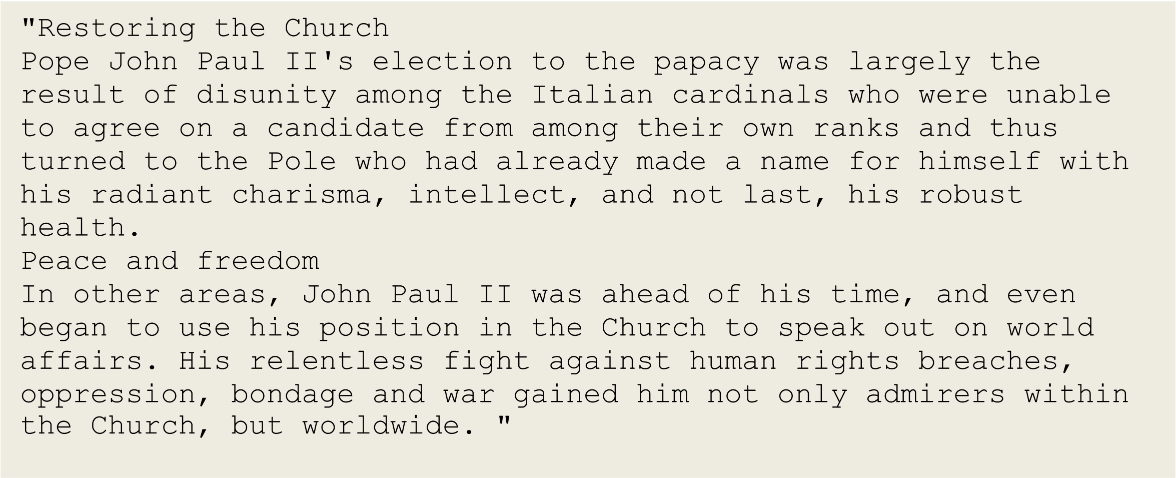
Example of a DW document fragment with markup:



```
"<P class=MsoNormal style=\"MARGIN: 0cm 0cm 0pt\"><B style=\"mso-bidi-
font-weight: normal\">Restoring the Church</b></p>\r\n<P class=MsoNormal
style=\"MARGIN: 0cm 0cm 0pt\">&nbsp;</p>\r\n<P class=MsoNormal
style=\"MARGIN: 0cm 0cm 0pt\">#b#Pope John Paul II's election to the
papacy was largely the result of disunity among the Italian cardinals who
were unable to agree on a candidate from among their own ranks and thus
turned to the Pole who had already made a name for himself with his
radiant charisma, intellect, and not last, his robust health. </p>
<B>Peace and freedom </b></p>\r\n<P class=MsoNormal style=\"MARGIN: 0cm
0cm 0pt\">&nbsp;</p>\r\n<P class=MsoNormal style=\"MARGIN: 0cm 0cm
0pt\">In other areas, John Paul II was ahead of his time, and even began
to use his position in the Church to speak out on world affairs. His
relentless fight against human rights breaches, oppression, bondage and
```

Figure 3: DW document with markup

The same document fragment after clean-up:



```
"Restoring the Church
Pope John Paul II's election to the papacy was largely the
result of disunity among the Italian cardinals who were unable
to agree on a candidate from among their own ranks and thus
turned to the Pole who had already made a name for himself with
his radiant charisma, intellect, and not last, his robust
health.
Peace and freedom
In other areas, John Paul II was ahead of his time, and even
began to use his position in the Church to speak out on world
affairs. His relentless fight against human rights breaches,
oppression, bondage and war gained him not only admirers within
the Church, but worldwide. "
```

Figure 4: DW document without markup

Furthermore, generic titles such as "*Documentaries and Reports*" have been identified. They are not passed on to the text analysis components. This improves e.g. key phrase training and extraction. For the DW corpus, titles that appear more than 100 times have been marked as generic titles. There are 67 generic titles in the English DW corpus.

For the Guardian texts the jericho parser has been used to remove all markup in the text body which is in the meta tag *article-body-blocks*. The contents of the meta tags *title* and *description* are used as title information. Currently no subtitles or paragraph structure is recognized. The author-annotated keywords are available in the meta tag *keywords*.

3.2. Next steps

Next steps are:

1. clean text versions for the crawled newspaper corpora of *El País, Le Monde* and *Die Zeit*;
2. improved clean text versions with paragraph structure and subtitle/subheader information for *The Guardian, El País, Le Monde* and *Die Zeit.*

4. ARCHITECTURE OF THE TEXT ANALYSIS COMPONENT

4.1. UIMA Platform

As explained in D5.3[2], the EUMSSI platform relies heavily on Apache UIMA (Unstructured Information Management Architecture)[3] and its native data structure, the CAS (Common Analysis Structure). Thus, on one hand the UIMA architecture is used to regulate the workflow between the different multimodal components, and, on the other, the CAS is used to store the aligned layers of analysis results. In the case of the Text Analysis component, it is completely implemented within the UIMA framework. In the case of the audio and video components, only their input/output workflow is managed by UIMA.
The UIMA CAS representation is a good fit for the needs of the EUMSSI project as it has a number of interesting characteristics:

- Annotations are stored "stand-off", meaning that the original content is not modified in any way by adding annotations. Rather, the annotations are entirely separate and reference the original content by offsets
- Annotations can be defined freely by defining a "type system" that specifies the types of annotations (such as Person, Keyphrase, Face, etc.) and the corresponding attributes (e.g. dbpediaUrl, canonicalRepresentation, ...)
- Source content can be included in the CAS (particularly for text content) or referenced as external content via URIs (e.g. for multimedia content)

Linguistic analysis components, or annotators, basically take two forms:

- Native "Analysis Engines", directly written for UIMA, or
- External existing analysis components integrated in UIMA by using "wrappers" that translate their inputs/outputs.

4.2. DKPro Linguistic Repository and Type System

DKPro Core is a collection of software components for natural language processing (NLP) based on the Apache UIMA framework.
Many NLP tools are already freely available in the NLP research community. DKPro Core provides UIMA components wrapping these tools (and some original tools) so they can be used interchangeably in UIMA processing pipelines. DKPro Core builds heavily on uimaFIT which allows for rapid and easy development of NLP processing pipelines, for wrapping existing tools and for creating original UIMA components.
Many of the tools used in the EUMSSI Text Analysis module come from the DKPro repository. Also the type system used for text analysis in EUMSSI is based on the DKPro type system in order to ensure compatibility with third party components. Types

[2] EUMSSI D5.3 Preliminary version of data integrated platform.
[3] http://uima.apache.org

not covered by DKPro are defined in a separate type system for the EUMSSI project, aligned with the metadata schema described in D2.3[4].

4.3. Linguistic processing chains or pipelines

4.3.1. *UIMA-based pipeline for Text Analysis*

Figure 5 shows the functional diagram of a UIMA pipeline, from input (using a Collection Reader), going through several processing components (or Analysis Engines), to the CAS Consumers which take the annotated data (stored in the CAS) and output it in the desired format (in the case of EUMSSI, databases and full-text indices for search).

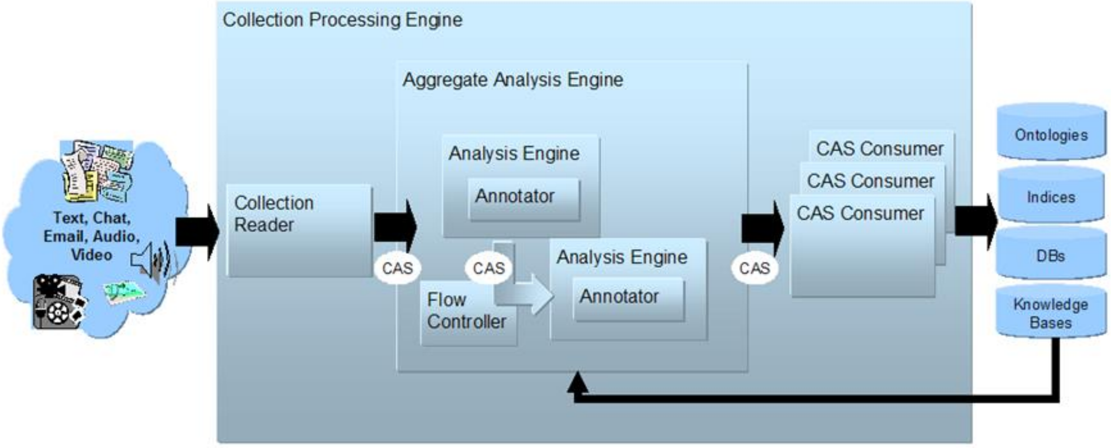


Figure 5: UIMA general pipeline

The UIMA-based linguistic pipeline comprises basic linguistic processing, such as sentence splitting, tokenization, part-of-speech tagging, and lemmatization, as well as semantic and sentiment annotation. Available open-source analysis modules, such as those provided by the OpenNLP project, have been used in conjunction with in-house developed language resources, and adapted to the task.

UIMA representation under CAS objects allows preserving the documents integrity since annotations are added as stand-off metadata. Thus, the original information is not modified while the metadata is enriched with each processing iteration.

In a nutshell, the idea is to progressively enrich each annotation using different tools. These tools employ a variety of methodologies, such as controlled annotation based on patterns and lists of names, statistically–based annotators build on annotated corpora, etc. The main goal of the linguistic processing is to explore the unstructured information found in text by detecting, extracting and classifying elements and their relevant correlations.

[4] EUMSSI D2.3 Data Infrastructure and representation definition

4.3.2. *Text Analysis pipelines in EUMSSI*

An annotation pipeline is a chain or sequence of analysis engines, each of which outputs a specific annotation after analysing the input object. The engines proceed in a sequential way, and some of them build on the results of the preceding engines.

Specific analysis pipeline are foreseen for each particular language (EN, ES, DE and FR) and for each type of source text (edited (e.g. news), user-generated (e.g. tweets), OCR and ASR). That is 16 distinct pipelines.

The motivation for having separate pipelines for different text types is because each type of text requires specific components: for example, as explained in section 8.2, ASR input may need to include a trueCaser; or certain tools may require specifically trained models, e.g. for user-generated content; or, finally, because certain engines cannot be applied to all text types, e.g. pos-tagger does not work on context-free OCR input.

Figure 6 models one of such pipelines, with the analysis engines that have been developed so far in EUMSSI.

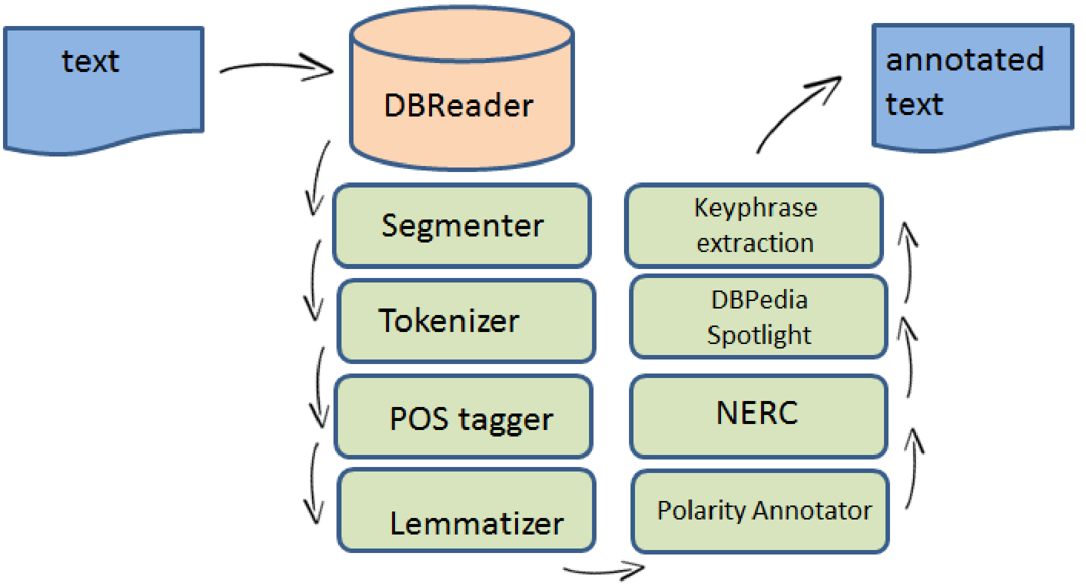


Figure 6: Standard text analysis pipeline

The Analysis Engines that are currently integrated in the Text Analysis pipeline are listed in the sections below.

4.3.2.1. Sentence Segmenter

The Sentence Detector module uses algorithms such as Maximum Entropy in order to decide whether a punctuation mark denotes the end of a sentence or not, thus segmenting the text into sentences.

4.3.2.2. Tokenizer

Tokens are the smallest units of linguistic information, and, generally speaking, can be either words or punctuation. Segmentation of sentences into tokens provides units suitable for lexical processing, such as lemmatizing or part-of-speech tagging. Consequently, proper operation of subsequent modules crucially depends upon the correct identification of words.

4.3.2.3. Part of Speech Tagger

The part-of-speech annotation tool identifies the basic syntactic category of a word or lexical unit (i.e. noun, verb, adjective, etc.) and the morpho-syntactic features encoded in the inflected wordform (i.e. number, gender, tense, etc.). Most of POS taggers are trained on syntactically annotated corpora.

4.3.2.4. Lemmatizer

Being able to assign a lemma to a word allows for clustering of a diversity of lexical forms that share a core meaning but whose morphological shape varies in function of tense, mood, gender and number values. Thus, a Spanish verbal form such as *dijo* (i.e. *he said*) can be straightforwardly clustered with other forms of the same verb *decir* (e.g. *dirá, dicho, decía, dice, ...*) , thus reducing the dimension of the vocabulary and the search space.

Most lemmatizers use dictionaries of lemmas and wordforms. On a first step all possible lemmas a given wordform may have are assigned. Later, a disambiguation module removes lemmas that do not fit the hypotesized POS tag.

4.3.2.5. Polarity Word Annotator

This module, based on the UIMA Concept Mapper, has the function of annotating polar terms that are obtained from word lists and dictionaries, as described in section 7.4.

4.3.2.6. Named Entity Recognizer and Classifier

NER operates on tokenized text, annotating names of person, organisations and locations, as described in section 5.1.

4.3.2.7. DBPediaSpotlight

As explained in section 5.2, DBpediaSpotlight annotates entities and concepts with links to DBpedia pages and with resource types. The types are not restricted to person, location and organisation but encompass all of the 272 classes of the DBpedia Ontology.

4.3.2.8. Keyphrase Extraction

Section 6.2 describes the keyphrase extraction tool that has been integrated into the pipeline. It detects relevant keyphrases within the text by providing their offsets as stand-off metadata.

5. NAMED ENTITY RECOGNITION AND DISAMBIGUATION

Named Entity Recognition (NER) tools determine mentions of names (such as "George Washington" or "Washington") and the type of name (such as *person*, *location*, *organisation*). Named entity disambiguation, also called identity resolution or entity linking, disambiguates named entity mentions by relating them to a uniform resource identifier (URI) such as a Wikipedia or DBpedia entry. Thus, if both mentions of "George Washington" and "Washington" are disambiguated as an entity of type *person*, they are linked to the same URI. Named entity disambiguation and named entity identification is needed in EUMSSI in order to relate information about entities across different text passages and across different media sources.

5.1. Named Entity Recognition (NER)

In EUMSSI, the Stanford NER tool[5] [Finkel et al 2005] is used. Illinois NER[6] has also been experimented with, but since Stanford NER provides very good results and is already integrated in DKPro, it has been chosen over other tools. Stanford NER detects named entities and categorise them into types or classes. E.g. "New York" is a named entity of type *location*.
Stanford NER uses a combination of conditional random field (CFR) sequence taggers trained on various corpora and a few additional features such as gazetteers to recognize named entities. For more details see [Manning et al 2014] and [Finkel et al 2005].

The advantages of Stanford NER are:

1. It is already integrated in DKPro.
2. Models for the four EUMSSI languages are either available or annotated training data is available.
3. Appropriate choice of gazetteers improves the system.

Disadvantages of Stanford NER:

1. The models are case-sensitive. Therefore the regular models do not provide satisfactory results for the lower case output of automatic speech recognition (ASR).
2. For the analysis of ASR separate lower-case models or other solutions need to be used.

5.1.1. **Named entity types**

The goal of NER within EUMSSI is to detect names of *persons, organisations* and *locations*, but also *dates* and *times*. Most available models and annotated training data only provide the first three classes, namely *persons, organisations* and *locations*. Models or annotated training data for *dates* and *times* are not available for Stanford

[5] http://nlp.stanford.edu/software/CRF-NER.shtml
[6] http://cogcomp.cs.illinois.edu/page/software_view/4

NER for FR, ES or DE. Therefore we use Stanford NER only for detecting names of person, organisation and location, and we plan to use a different tool for detection of dates and times. The chosen tool will probably be *Heideltime*[7] [Strötgen et al 2013].

5.1.2. **Available models and annotated training corpora**

More specifically, the following models and annotated training corpora are available for Stanford NER for the different EUMSSI languages:
For English, Stanford provides a 4 class model trained for CoNLL, a 7 class model trained for MUC, and a 3 class model trained on both data sets for the intersection of those class sets. Currently the 3 class model is used which recognizes person, location and organisation. The 7 class model also recognizes names of dates and times.
For German, Stanford recommends a model by Sebastian Pado[8] [Faruqui & Pado 2010] which is currently used. It detects person, location and organisation.
For French, WikiNER data provided by schwa lab[9] [Nothman et al 2012] has been used to train a three class model which recognizes person, location and organisation.
For ES, the Ancora corpus[10], tagged by UPF, has been used to train a 3 class model covering *person, location and organisation*.
The Stanford NER system has been enhanced for all four languages by using DBpedia names as gazetteers and by parameter tuning.
Example output for "*Fracking evokes 'angst' in Germany*":[11]

```
Fracking O
evokes O
' O
angst O
' O
in O
Germany LOC
```

Figure 7: Stanford NER output

StanfordNER annotates the types PER, LOC and ORG. O stands for no NE.

5.1.3. ***Dates and Times***

In the EUMSSI context, dates and times are important for anchoring events on a time line. The publication date of the text narrows down the potential temporal anchor and often coincides with it - especially in the news domain - , however, this cannot be taken for granted. Therefore additional NER of date and time expressions is necessary. It is planned to use Heideltime to recognise dates and times. Heideltime provides a

[7] https://code.google.com/p/heideltime/
[8] http://www.nlpado.de/~sebastian/software/ner_german.shtml
[9] http://schwa.org/projects/resources/wiki/Wikiner
[10] http://clic.ub.edu/corpus/en
[11] DKpro-internally, the entity model that has been developed in Deliverable 2.3 is used.

version that can be integrated into UIMA and it is available for all four EUMSSI languages.

5.1.4. *Impact*

1. A first version of Named Entity Recognition is running for all four EUMSSI languages.
2. A first version of Named Entity Recognition in lower-cased ASR output is running for EN.

5.1.5. *Next steps*

1. Training of lower case models for FR, ES, and DE.
2. Installing an additional tool for recognition of dates and times (Heideltime).

5.2. Named Entity Disambiguation and Linking

For named entity disambiguation and linking DBpediaSpotlight[12] [Daiber et al 2013, Mendes et al. 2011] is used. DBpediaSpotlight provides links to DBpedia pages and annotations of resource types. The types are not restricted to person, location and organisation but encompass all of the 272 classes of the DBpedia Ontology. Thus a name such as "Washington" is disambiguated not only with respect to location and person but more fine-grained types are provided that distinguish whether the state or the city of Washington is referred to. The overlap with Stanford NER is discussed in section 5.2.6 below.

DBpediaSpotlight uses a disambiguation algorithm that is based on cosine similarities and a modification of the TF-IDF weights which are used to compute the relevance of a given word for disambiguating the appropriate DBpedia concept. For a more complete description see [Mendes et al 2011] and [Daiber et al 2013].

The main advantages of DBpediaSpotlight are:

1. DBpediaSpotlight is available for all four EUMSSI languages.
2. The program code is downloadable.
3. DBpediaSpotlight is written in JAVA and a provisional wrapper for DKPro exists.
4. DBpediaSpotlight can be configured with respect to several parameters.
5. DBpediaSpotlight provides as output additional scores and candidate lists.
6. DBpediaSpotlight delivers competitive results.
7. DBpediaSpotlight also processes lower case text stemming from ASR.

A major disadvantage is:

1. DBpediaSpotlight results for lower case text are deprecated especially for multi-word names.
2.

[12] http://dbpedia-spotlight.github.io/demo/

5.2.1. *Comparison with other tools*

There are several studies that provide comparisons of several tools with respect to NER and NEL functionalities. In [Mendes et al 2011] a comparative evaluation of NER tools is conducted with an optimised confidence parameter for DBpediaSpotlight. The highest F-score is reached with a confidence parameter of 0.7 which achieves an F-score of 56.0 whereas the default configuration yields an F-score of 45.2. In their study only the Wiki Machine[13] achieves a higher F-score, whereas Zemanta[14], Alchemy[15], Open Calais[16] and Ontos[17] achieve lower F-scores.

In the comparative evaluation of [Gangemi 2013], DBpediaSpotlight as NER tool achieves an F-score of 0.33 and as NE resolution (linking) tool an F-score of 0.40 for a short English text containing 1491 characters and 58 named entities (with the default confidence setting). Other tools such as AIDA[18] and Wikimeta[19] achieve better F-scores, however, they are not available for all EUMSSI languages and (partly) the source code is not downloadable.

[Daiber et al 2013] also report competitive results for DBpediaSpotlight. They report the following F-scores: German: 46.32, French: 45.02 Spanish: 44.10. (For English, no score is reported.)

5.2.2. *Input parameters*

DBpediaSpotlight can be configured with respect to several parameters:

1. confidence: The confidence configuration applies two checks:
   a. The similarity score of the first ranked entity must be bigger than a threshold. [20]
   b. The gap between the similarity score of the first and second ranked entity must be bigger than a relative threshold.

   The parameter range is 0.0 to 1.0 where 0.0 is liberal and 1.0 is restrictive. The default is 0.1.
2. support: number of in-links
3. type: allowed or forbidden URIs or types, works in combination with policy-setting
4. policy: whitelist (allowed)(default) or blacklist (forbidden)
5. coreferenceResolution: is a heuristic that seeks coreference in all text and infer the surface form. When it is true, no other filter will be applied. (default:true)

[13] http://www.thewikimachine.fbk.eu
[14] http://www.zemanta.com/
[15] http://www.alchemyapi.com/company/
[16] http://www.opencalais.com/
[17] http://www.ontos.com/products/
[18] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/
[19] http://www.wikimeta.com/wapi/semtag.pl
[20] http://wiki.dbpedia.org/spotlight/technicaldocumentation?show_files=0
For a description of similarity score see *output features* in section 5.2.3.

In EUMSSI the confidence parameter is set to 0.0 and the support parameter is also set to 0. Coreference resolution is not used because it blocks the usage of other filters. Whitelists and blacklists are currently not used because the goal is to provide as much information as possible.[21]

5.2.3. *Output features*

A major advantage of DBPediaSpotlight is that it provides rich output information, e.g. resource prominence (support feature) and topic pertinence (similarity score) of the entities specified. These features can be used in the inference mechanisms that build on the text analysis. It is also possible to output the n-best candidate list. This feature can be used to improve the Named Entity Disambiguation by using cross-media information and information stemming from other tools such as StanfordNER to re-rank the n-best candidates.

Technically, DBpediaSpotlight provides two different operations which provide different outputs: The *annotate* operation annotates the input tokens with maximally one (= the best) type annotation and URI (web link). The *candidate* operation annotates tokens with n-best candidate lists that also contain the respective additional scores.

Output features provided by the *annotate* operation:[22]

1. similarity score: the topical relevance of the annotated resource for the given context.
2. support: how prominent is this entity, i.e. number of inlinks in Wikipedia
3. types: types from DBpedia, Schema, Freebase, foaf
4. URI: link to DBpedia web page

Example: "Washington" disambiguated as Washington State:

```
JCasResource
  sofa: _InitialView
  begin: 0
  end: 10
  similarityScore: 0.7550103129997501
  support: 24525
  types: "Schema:Place,
         DBpedia:Place,
         DBpedia:PopulatedPlace,
         DBpedia:Region,
         Schema:AdministrativeArea,
         DBpedia:AdministrativeRegion"
  URI: http://dbpedia.org/resource/Washington_(state)
```

Figure 8: DBpediaSpotlight output (annotate operation)

[21] Blacklists do not work properly, see: https://github.com/dbpedia-spotlight/dbpedia-spotlight/issues/251.

[22] Notice that the confidence value is not an explicit output feature, see discussion in http://sourceforge.net/p/dbp-spotlight/mailman/message/31849395/

Output features provided by the *candidate* operation:

N-best candidate list including the following features for each candidate:

1. similarity score: the topical relevance of the annotated resource for the given context.
2. support: how prominent is this entity, i.e. number of inlinks in Wikipedia
3. types: types from DBpedia and Schema
4. URI: link to DBpedia web page

Additional features are:[23]

5. priorScore: normalized support
6. contextualScore: score from comparing the context representation of an entity with the text (e.g. cosine similarity with if-icf weights)
7. percentageOfSecondRank: measure by how much the winning entity has won by taking contextualScore_2ndRank / contextualScore_1stRank, which means the lower this score, the further the first ranked entity was "in the lead"
8. finalScore: combination of all of them

Example: n-best candidate list for "Washington" consisting of Washington State, Washington D.C. and George Washington:

```
<surfaceForm name="Washington" offset="0">
  <resource label="Washington (state)"
        uri="Washington_(state)"
        contextualScore="0.10367993528279018"
        percentageOfSecondRank="0.3080651605722548"
        support="24525"
        priorScore="2.039994947300588E-4"
        finalScore="0.7550103129997501"
        types="Schema:Place, DBpedia:Place,
                DBpedia:PopulatedPlace, DBpedia:Region,
                schema:AdministrativeArea, DBpedia:AdministrativeRegion"/>
   <resource label="Washington, D.C."
        uri="Washington,_D.C."
        contextualScore="0.08173821375812629"
        percentageOfSecondRank="0.03905833939906741"
        support="45486"
        priorScore="3.783535582993458E-4"
        finalScore="0.23259237330797633"
        types="Schema:Place, DBpedia:Place,
                DBpedia:PopulatedPlace, DBpedia:Settlement"/>
   <resource label="George Washington"
```

[23] see https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service

```
        uri="George_Washington"
        contextualScore="0.09308415672842982"
        percentageOfSecondRank="0.12614828083179355”
        support="7902"
        priorScore="6.572901151302445E-5"
        finalScore="0.009084671858297526"
        types="DBpedia:Agent, Schema:Person,
                http://xmlns.com/foaf/0.1/Person,
                DBpedia:Person, DBpedia:OfficeHolder"/>
</surfaceForm>
```

Figure 9: DBpediaSpotlight output (candidates operation)

The n-best candidate list can be used for cross-media and cross-tool enhancement of the text analysis results, especially if the text type does not provide enough context to properly disambiguate the entities, as it is often the case with OCR output, but possibly also with other text types (e.g. article titles, short tweets, etc.)

5.2.4. *Program availability*

DBpediaSpotlight has been tested as a web service for all EUMSSI languages. In addition, the English module has been locally installed and tested. There are two different versions: [24] a lucene-based version and a version with a statistical backend. The currently downloaded version is the lucene version 0.6.5 and the full version of the latest models (version 0.5).

5.2.5. *DKPro integration*

There is a provisional wrapper for DBPediaSpotlight that allows to call the *annotate* operation but does not work for the *candidates* operation.

5.2.6. *DBpedia Spotlight and Stanford NER*

There is considerable overlap between StanfordNER and DBpediaSpotlight since DBpediaSpotlight provides not only entity linking, but also name recognition and recognition of types of names. The number of types recognised by DBpediaSpotlight is much larger than the ones recognised by Stanford NER. Therefore the question arises whether both tools are necessary or whether DBpediaSpotlight suffices to do both Named Entity Recognition, Classifying and Disambiguation.

Keeping Stanford NER makes sense if there are cases in which Stanford NER performs better than DBpediaSpotlight. Then, the results of both can be combined to provide better output. E.g. [Rizzo et al 2014] provide an evaluation of NER tools that shows that Stanford NER has very high scores for precision and recall, that are higher than the ones obtained by DBpediaSpotlight (they do not provide figures but only graphs).

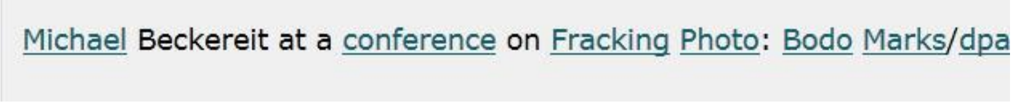
In their experiment, they train a support vector machine (SVM) with the annotations of several NER tools and a gold standard in order to optimize the NER annotations. The

[24] See https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Installation

resulting combined SVM is not much better than StanfordNER alone. Hence - as first approximation – we have decided to take Stanford NER to filter out ambiguous DBPediaSpotlight links. The next step would be to train a classifier.

Manual inspection of the output of both tools has also led to the assessment that Stanford NER is more precise in assigning types and that it also recognises names of lesser known entities such as inhabitants of villages, journalists and the like, which could be relevant as opinion holders and which are often incorrectly linked to some more famous entity by DBpediaSpotlight.

This is illustrated by the following example taken from the document in Figure 16. Here, DBpediaSpotlight does not properly recognize the name "Michael Beckereit":



Michael Beckereit at a conference on Fracking Photo: Bodo Marks/dpa

Figure 10: DBpediaSpotlight annotation

DBpediaSpotlight does not properly recognize the end of the name and incorrectly links "Michael" to *Michael Jackson* with a high similarity score of 0.9:

```
JCasResource
   sofa: _InitialView
   begin: 257
   end: 264
   similarityScore: 0.939562459194219
   support: 6694
   types:
"Schema:MusicGroup,DBpedia:Agent,Schema:Person,Http://xmlns.com/foaf/0.1/Person
,DBpedia:Person,DBpedia:Artist,DBpedia:MusicalArtist"
   URI: "http://dbpedia.org/resource/Michael_Jackson"
```

Figure 11: DBpediaSpotlight annotation for "Michael"

Stanford NER correctly recognizes the beginning and end of the name:

```
Michael       PER
Beckereit     PER
```

Figure 12: StanfordNER output for "Michael Beckereit"

Thus, the Stanford NER name offsets can be taken as an indication that the link provided by DBpediaSpotlight is incorrect.

Another advantage of Stanford NER is that it seems to perform better on ASR output (with lower case models) than DBpediaSpotlight. However, more systematic evaluation is needed.

5.2.7. *Impact*

A first version of Named Entity Disambiguation and Linking is running for all four EUMSSI languages.

5.2.8. *Next steps*

1. Modify the DKPro wrapper to allow calling the *candidates* operation.
2. Use the DBpediaSpotlight n-best candidate list to enhance the Named Entity Linking of e.g. OCR output with cross-media inferencing.
3. Possibly training of a classifier that combines StanfordNER and DBpediaSpotlight NER.
4. Systematic evaluation and comparison of StanfordNER and DBpediaSpotlight annotations for regular text and lower-case ASR output.

6. EXTRACTION OF EVENTS, RELATIONS AND TOPICS

6.1. Introduction

In order to provide information about events, relations and topics, the first step has been to install a keyphrase extraction tool which provides a variable range of keyphrases for all four EUMSSI languages. Longer keyphrases provide information about relations and events if they contain verbal expressions. Shorter, nominal keyphrases provide topic information and can be used to weight sentences for text summarization functionalities. Keyphrases can also be used to compute text similarity which is a prerequisite for text clustering.

Keyphrases, unlike the entities extracted by DBpediaSpotlight, are weighted according to TFxIDF which reflects their relevance for a given document relative to a document collection. This information is important for document retrieval and search functionalities.

For later project phases it is planned to add tools that are more specific for the different purposes (event, relation and topic extraction) if this improves the contextualisation and recommendation tools.

6.2. Keyphrase extraction

For keyphrase extraction, KEA (Keyphrase Extraction Algorithm)[25] [Medelyan et al 2005, 2006] is used. As [Witten et al. 2000] point out, KEA keyphrase extraction does not use a controlled vocabulary but chooses keyphrases from the text itself. Thus, KEA keyphrase extraction is not limited to the topics or domains of the training data. KEA is a Naïve Bayes classifier trained with the following features:

1. *TFxIDF*: Term Frequency times Inverse Document Frequency
2. *First Occurrence*

TFxIDF is a measure of a term's frequency in a document compared to its frequency in a document collection. *First Occurrence* measures the distance into the document of the term's first occurrence relative to the document length. Candidate keyphrases are case-folded bags-of-stems derived from token sequences found in the text. There are special treatments for punctuation marks, numbers and stopwords. As a result, token sequences such as "ban fracking", "ban of fracking" and "fracking ban" are associated with the same bag-of-stems, namely "ban frack". For each bag-of-stem keyphrase, the most frequent surface form per document is retained for presentation to the user. KEA provides as output a list of keyphrases containing for each keyphrase the most frequent surface form, the bag-of-stems and a score. The order of keyphrases in the list represents an additional ranking of the keyphrases. KEA uses the WEKA[26] workbench which provides a range of machine learning (ML) techniques, see [Hall et al 2009].

[25] http://www.nzdl.org/Kea/description.html

[26] http://www.cs.waikato.ac.nz/ml/index.html

The advantages of KEA are:

1. KEA keyphrases are not limited to a controlled vocabulary.
2. KEA can be trained for all four EUMSSI languages.
3. Its Java source code is available under GNU public license.
4. KEA provides scores for key phrases.
5. KEA provides an implicit complete ranking of keyphrases.
6. There are several parameters such as:
    a. Parameter for the number of extracted key phrases
    b. Parameter for the length of key phrases
    c. Parameter for ignoring proper nouns
    d. Stopword list
    e. Parameter for in-domain training corpora
    f. Parameter for using ontology as keyphrase candidates
7. KEA applies case-folding, therefore ASR output can be processed without further ado.

Further advantages of KEA which are reported by [Witten et al 2000] and confirmed by preliminary experiments are:

8. A small training set of 50 documents suffices to produce adequate keyphrase extraction.
9. KEA is fairly robust with regard to key phrase extraction out of different domains even though training with an in-domain corpus improves scores.
10. KEA is also fairly robust with respect to text length.

Disadvantages of KEA:

1. KEA needs annotated training data.
2. The scores are not diverse enough to provide interesting differentiations for search requests and visualisations with tag clouds.
3. The complete ranking of key phrases is only implicitly provided by the order in the output list.
4. The top-ranked keyphrases are predominantly one-word keyphrases. For applications such as tag clouds two-word keyphrases are preferable.
5. KEA lists only the most frequent surface form for each key phrase per document. Thus the offsets which are computed on the surface forms might not be complete.
6. The *first occurrence* feature is inappropriate if texts contain several divergent but equally important topics as is the case in interviews and speeches.

The disadvantages of KEA can be remedied by the following measures:

Ad 1) Keyphrase annotations are often available, e.g. many of the crawled news items contain author-annotated keyphrases. For the case that such author-annotated keyphrases do not exist, a tool for keyphrase annotation is available at GFAI, see section 6.2.5.
Ad 2) Increasing the training data has led to more diverse score distributions.

Ad 3) The implicit ranking is converted into a separate output feature called *rank*.

Ad 4) It is possible to restrict keyphrases to two-stem units, however, then relevant one-word keyphrases such as "*fracking*" are lost. Another option is to interpolate high-ranked one-word and two-word keyphrases in an inference step to determine an appropriate mix of one-word and two-word keyphrases.

Ad 5) Case-folding and lemmatization of keyphrase reduce the problem of listing all surface forms somewhat but do not account for different word orders within keyphrases or deleted stopwords. Therefore, a better solution would be to output all surface forms instead of only the most frequent surface form. This requires going deep into the code since the surface forms are discarded early on. This hasn't been done yet.

Ad 6) Divergent topics and sub-topics are captured either by increasing the number of extracted keyphrases per document or by applying keyphrase extraction to smaller text units such as paragraphs or by training a separate topic model, see section 6.5.

6.2.1. ***Model training***

KEA provides models, stemmers and stopword lists for EN, ES and FR. For DE it provides a stemmer and a stopword list but no model. Therefore, the first measure has been to close the language gap and to train a model for German using annotated DW data generated by the GFAI keyword extraction tool. For unsupervised extraction of keyphrases with the GFAI tool see section 6.2.5.

We are currently exploring the following issues:

1. Impact of large numbers of extracted keyphrases
2. Impact of training corpus size on F-score and score diversity and keyphrase length
3. Comparison of models trained with author-annotated keyphrases and automatically annotated keyphrases (GFAI tool)
4. Impact of applying KEA to smaller text units such as paragraphs and sentences
5. Impact of different training corpora (in-domain versus out of domain)

6.2.2. ***Experiment with large number of extracted keyphrases***

In this experiment, KEA was configured to output 500 keyphrases for each DW article thus pushing the number of extracted keyphrases to an extreme.[27] The idea behind this experiment is to provide a rich pool of data on which different search filters can operate and provide different sets of keyphrases determined by the search and data mining requests, possibly also influenced by the user. Additional filters could be to output only top-ranking multi-word keyphrases or keyphrases with/without verbs or keyphrases containing a certain keyword such as *fracking* and so on.

[27] The number of keyphrases is too high given that the text only contains 936 words. A specific ratio of number of extracted keyphrases relative to text length needs to be established so that the size of the extracted keyphrases is considerably smaller than the text size.

Example1: the 20 highest ranking KEA key phrases for the DW text entitled "Fracking evokes 'angst' in Germany" see Figure 16 for the complete text.[28]

Token:	Stem:	Score:
Fracking	frack	0.0502
Germany	germani	0.034
experts	expert	0.034
risks	risk	0.034
Germans	german	0.034
Europe	europ	0.034
Merkel	merkel	0.034
water	water	0.034
pressure	pressur	0.034
streets	street	0.034
Thousands	thousand	0.034
plans	plan	0.034
parts	part	0.034
gas	gas	0.0339
method	method	0.0339
extract	extract	0.0339
soil	soil	0.0339
chemicals	chemic	0.0339
concerned	concern	0.0339
Photo	photo	0.0339[29]

Figure 13: Highest ranking keyphrases

They contain named entities such as Merkel, and terms such as "risks", "gas", "water" and also attitude verbs such as "concerned" but out of context it is unclear what the risks are or who is concerned. Example2: The 20 highest ranking multiword keyphrases for the same newspaper article are:

Token:	Stem:	Score:
environmental risks	environment risk	0.0338
environmental experts	environment expert	0.0338
Germans are concerned	concern german	0.0338
extract oil	extract oil	0.0338
high pressure	high pressur	0.0338
Chancellor Angela Merkel	angela chancellor merkel	0.0338
Angela Merkel	angela merkel	0.0338
nuclear waste	nuclear wast	0.0338
natural gas	gas natur	0.0338

[28] In comparison, the author-assigned keyphrases for the same article are: "CDU, FDP, fracking, environment protection, law, Bundesrat, Angela Merkel, Oettinger, gas production".

[29] The term 'Photo' is part of a copyright notice under a picture. The html markup is not fine-grained enough yet to detect such descriptions.

large scale	larg scale	0.0338
Martin Faulstich	faulstich martin	0.0338
party CDU	cdu parti	0.0338
fracking here in Germany	frack germani	0.0338
method to extract	extract method	0.0338
risks of fracking	frack risk	0.0338
fracking activities	activ frack	0.0338
stop all fracking	frack stop	0.0338
Recommendation to forego	forego recommend	0.0338
Forego fracking	forego frack	0.0338
need additional gas	addit gas need	0.0338

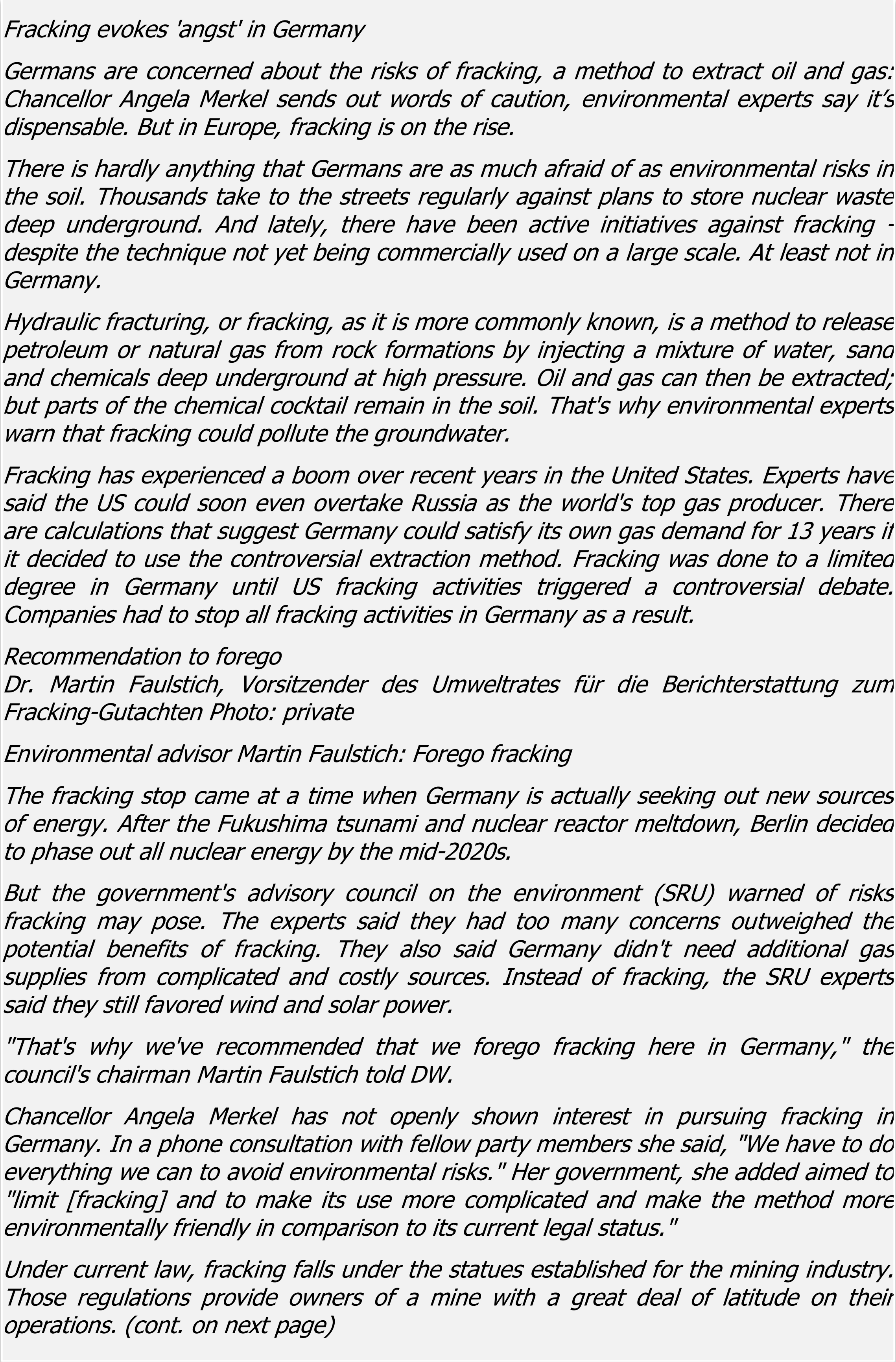
Figure 14: Highest ranking multi-word keyphrases

The multiword keyphrases are more expressive than the single-word keyphrases. Eg. "environmental risks" is more expressive than "risks". More named entities appear: "Martin Faulstich" and "party CDU". Multi-word keyphrases also contain rudimentary relations such as "Germans are concerned", "extract oil", "method to extract", "stop all fracking", "Forego fracking" and "need additional gas". The 20 highest ranking keyphrases containing the string [Ff]racking for the same article are:

Token:	Stem:	Score:
Fracking	frack	0.0502
fracking here in Germany	frack germani	0.0338
risks of fracking	frack risk	0.0338
fracking activities	activ frack	0.0338
stop all fracking	frack stop	0.0338
Forego fracking	forego frack	0.0338
ban fracking	ban frack	0.0338
Fracking evokes	evok frack	0.0338
fracking is on the rise	frack rise	0.0338
conference on Fracking	confer frack	0.0338
conference on Fracking Photo	confer frack photo	0.0338
Fracking Photo	frack photo	0.0338
active initiatives against fracking	activ frack initi	0.0338
initiatives against fracking	frack initi	0.0338
environmental experts warn that fracking	environment expert frack warn	0.0338
experts warn that fracking	expert frack warn	0.0338
warn that fracking	frack warn	0.0338
warn that fracking could pollute	frack pollut warn	0.0338
fracking could pollute	frack pollut	0.0338
fracking could pollute the groundwater	frack groundwat pollut	0.0338
Fracking has experienced a boom	boom experienc frack	0.0338

Figure 15: Highest ranking keyphrases containing "fracking"

The original document can be found in the following figure:



Fracking evokes 'angst' in Germany

Germans are concerned about the risks of fracking, a method to extract oil and gas: Chancellor Angela Merkel sends out words of caution, environmental experts say it’s dispensable. But in Europe, fracking is on the rise.

There is hardly anything that Germans are as much afraid of as environmental risks in the soil. Thousands take to the streets regularly against plans to store nuclear waste deep underground. And lately, there have been active initiatives against fracking - despite the technique not yet being commercially used on a large scale. At least not in Germany.

Hydraulic fracturing, or fracking, as it is more commonly known, is a method to release petroleum or natural gas from rock formations by injecting a mixture of water, sand and chemicals deep underground at high pressure. Oil and gas can then be extracted; but parts of the chemical cocktail remain in the soil. That's why environmental experts warn that fracking could pollute the groundwater.

Fracking has experienced a boom over recent years in the United States. Experts have said the US could soon even overtake Russia as the world's top gas producer. There are calculations that suggest Germany could satisfy its own gas demand for 13 years if it decided to use the controversial extraction method. Fracking was done to a limited degree in Germany until US fracking activities triggered a controversial debate. Companies had to stop all fracking activities in Germany as a result.

Recommendation to forego
Dr. Martin Faulstich, Vorsitzender des Umweltrates für die Berichterstattung zum Fracking-Gutachten Photo: private

Environmental advisor Martin Faulstich: Forego fracking

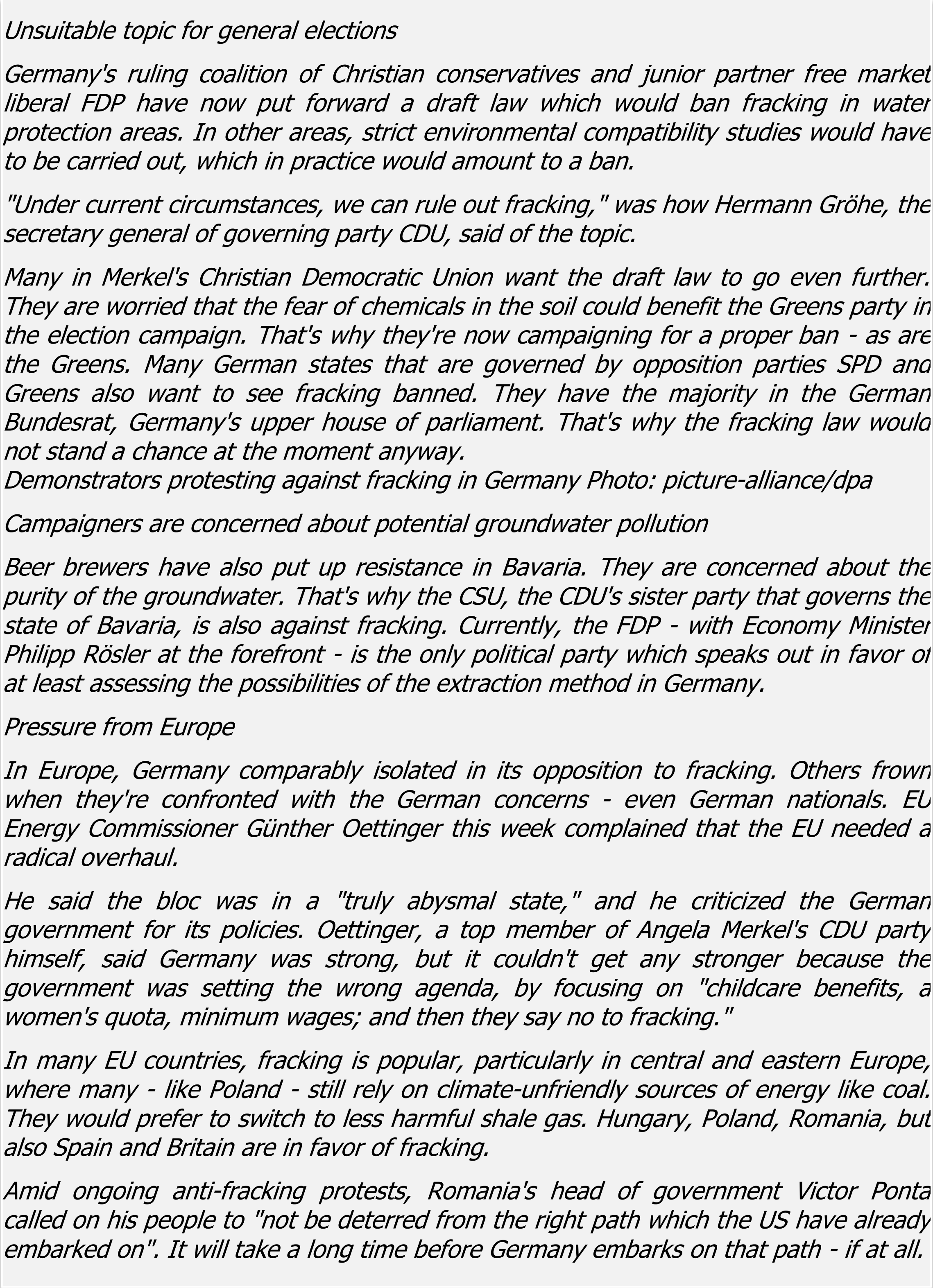
The fracking stop came at a time when Germany is actually seeking out new sources of energy. After the Fukushima tsunami and nuclear reactor meltdown, Berlin decided to phase out all nuclear energy by the mid-2020s.

But the government's advisory council on the environment (SRU) warned of risks fracking may pose. The experts said they had too many concerns outweighed the potential benefits of fracking. They also said Germany didn't need additional gas supplies from complicated and costly sources. Instead of fracking, the SRU experts said they still favored wind and solar power.

"That's why we've recommended that we forego fracking here in Germany," the council's chairman Martin Faulstich told DW.

Chancellor Angela Merkel has not openly shown interest in pursuing fracking in Germany. In a phone consultation with fellow party members she said, "We have to do everything we can to avoid environmental risks." Her government, she added aimed to "limit [fracking] and to make its use more complicated and make the method more environmentally friendly in comparison to its current legal status."

Under current law, fracking falls under the statues established for the mining industry. Those regulations provide owners of a mine with a great deal of latitude on their operations. (cont. on next page)



Unsuitable topic for general elections

Germany's ruling coalition of Christian conservatives and junior partner free market liberal FDP have now put forward a draft law which would ban fracking in water protection areas. In other areas, strict environmental compatibility studies would have to be carried out, which in practice would amount to a ban.

"Under current circumstances, we can rule out fracking," was how Hermann Gröhe, the secretary general of governing party CDU, said of the topic.

Many in Merkel's Christian Democratic Union want the draft law to go even further. They are worried that the fear of chemicals in the soil could benefit the Greens party in the election campaign. That's why they're now campaigning for a proper ban - as are the Greens. Many German states that are governed by opposition parties SPD and Greens also want to see fracking banned. They have the majority in the German Bundesrat, Germany's upper house of parliament. That's why the fracking law would not stand a chance at the moment anyway.
Demonstrators protesting against fracking in Germany Photo: picture-alliance/dpa

Campaigners are concerned about potential groundwater pollution

Beer brewers have also put up resistance in Bavaria. They are concerned about the purity of the groundwater. That's why the CSU, the CDU's sister party that governs the state of Bavaria, is also against fracking. Currently, the FDP - with Economy Minister Philipp Rösler at the forefront - is the only political party which speaks out in favor of at least assessing the possibilities of the extraction method in Germany.

Pressure from Europe

In Europe, Germany comparably isolated in its opposition to fracking. Others frown when they're confronted with the German concerns - even German nationals. EU Energy Commissioner Günther Oettinger this week complained that the EU needed a radical overhaul.

He said the bloc was in a "truly abysmal state," and he criticized the German government for its policies. Oettinger, a top member of Angela Merkel's CDU party himself, said Germany was strong, but it couldn't get any stronger because the government was setting the wrong agenda, by focusing on "childcare benefits, a women's quota, minimum wages; and then they say no to fracking."

In many EU countries, fracking is popular, particularly in central and eastern Europe, where many - like Poland - still rely on climate-unfriendly sources of energy like coal. They would prefer to switch to less harmful shale gas. Hungary, Poland, Romania, but also Spain and Britain are in favor of fracking.

Amid ongoing anti-fracking protests, Romania's head of government Victor Ponta called on his people to "not be deterred from the right path which the US have already embarked on". It will take a long time before Germany embarks on that path - if at all.

Figure 16: Document "Fracking evokes 'angst' in Germany"

The listing of keyphrases in Figure 15 that contain the term *fracking* conveys that fracking is highly controversial. Negative expressions such as “risks of fracking”, “stop all fracking”, “Forego fracking”, “ban fracking”, “active initiatives against fracking”, “environmental experts warn that fracking” and “fracking could pollute the groundwater” can be seen alongside with positive expressions such as “fracking is on the rise” and “Fracking has experienced a boom”, where negative expressions outnumber the positive ones.

6.2.3. *Sample visualization of keyphrases*

Keyphrases of differing length provide information of different granularity. This can be used to provide scalable search filters to the user. Users determine the level of granularity, possibly also by combining several search criteria such as keyword search, named entity spotting and sentiment/attitude analysis. Thus, it is e.g. possible that the user determines keyphrase length, or whether the keyphrases contain verbs, or certain named entities and the user can zoom in to visualize families of keyphrases (e.g. keyphrases containing the term *fracking* or keyphrases containing attitude verbs such as *warn*, *concerned* and so on).

To take the above article about fracking as an example, the one-word keyphrases give an overview of prominent named entities (Germany, Germans, Europe, Merkel), terms (risks, gas, water, soil, chemicals) and attitudes (concerned). They can be visualized as tag cloud:

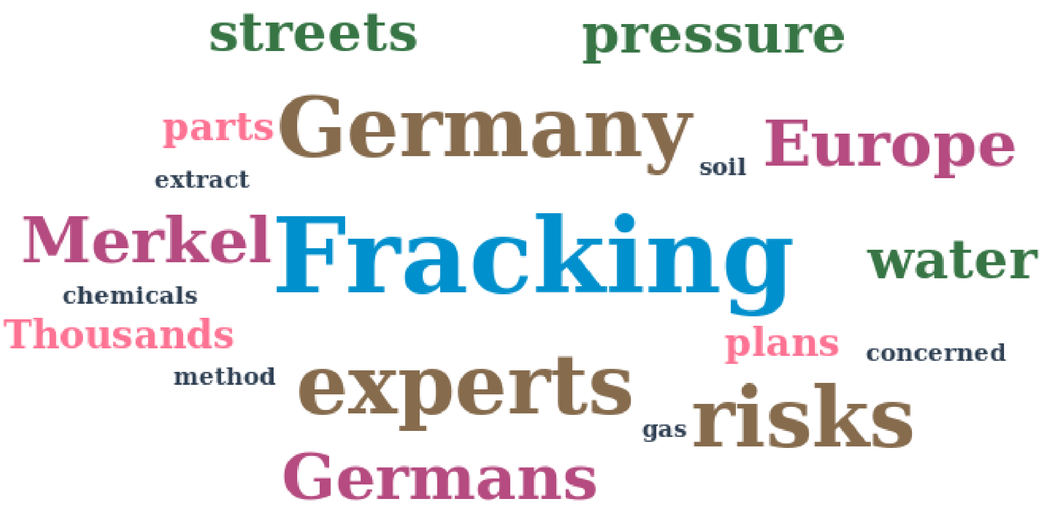


Figure 17: Tag cloud of single-word keyphrases

The multi-word keyphrases convey more specific information, e.g they specify the type of risks (*environmental risks*) and they bring up new relevant named entities and terms such as Martin Faulstich, CDU and “*nuclear waste*”. And they specify rudimentary relations such as “*Germans are concerned” and* “*extract oil*”.

They can be visualized as in the following tag cloud:

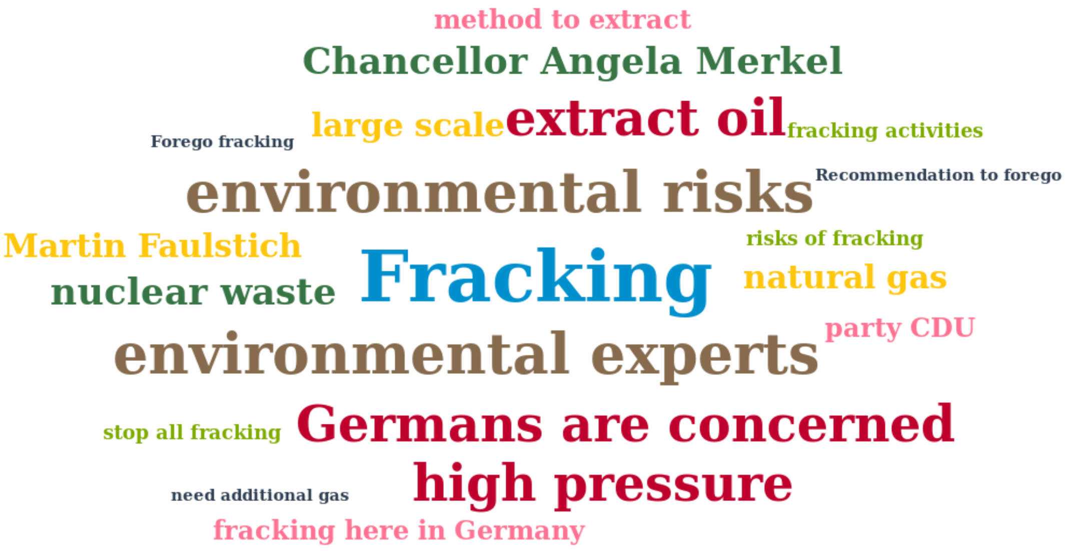


Figure 18: Tag cloud of multi-word keyphrases

Selecting the keyphrases that contain the term *fracking* provides even more detailed information of the risks of fracking (*fracking could pollute the groundwater*) and the attitudes towards fracking (*environmental experts warn that fracking...*) as illustrated in the following tag cloud:

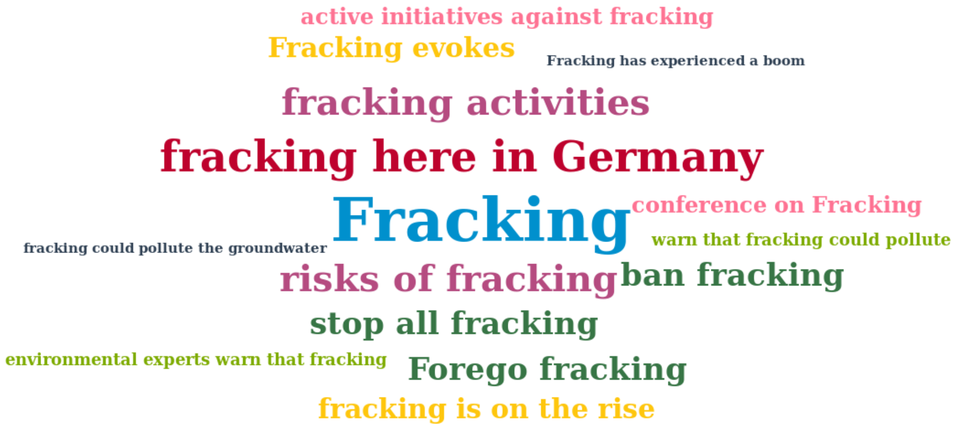


Figure 19: Tag cloud with keyphrases containing the term *fracking*

In order to obtain the word-clouds presented above, we have taken the outputs shown in Figure 13, Figure 14 and Figure 15, and filtered them according to the following criteria:

1. Overlap between keyphrases is reduced by ignoring keyphrases that are a substring of a higher-ranked keyphrase, thus, longer keyphrases that convey more content are preferred. [30]
2. Smaller groups of keyphrases: Since the scores are not diverse enough to define small groups of keyphrases, rank information which is encoded in the order of the keyphrase list is used to define smaller groups of two or three keyphrases.

These filters are exemplified in Figure 20. Grey letters indicate keyphrases that are ignored because they are substrings of higher ranked keyphrases. The keyphrases with score 0.0338 are grouped into sets of two or three keyphrases, indicated by substituting the last digit in the score with a lower digit:

Token:	Stem:	Score:
Fracking	frack	0.0502
fracking here in Germany	frack germani	0.0338
risks of fracking	frack risk	0.0338/6
fracking activities	activ frack	0.0338/6
stop all fracking	frack stop	0.0338/5
Forego fracking	forego frack	0.0338/5
ban fracking	ban frack	0.0338/5
Fracking evokes	evok frack	0.0338/4
fracking is on the rise	frack rise	0.0338/4
conference on Fracking	confer frack	0.0338/3
active initiatives against fracking	activ frack initi	0.0338/3
initiatives against fracking	frack initi	0.0338
environmental experts warn that fracking	environment expert frack warn	0.0338/2
experts warn that fracking	expert frack warn	0.0338
warn that fracking	frack warn	0.0338
warn that fracking could pollute	frack pollut warn	0.0338/2
fracking could pollute	frack pollut	0.0338
fracking could pollute the groundwater	frack groundwat pollut	0.0338/1
Fracking has experienced a boom	boom experienc frack	0.0338/1

Figure 20: Filtering keyphrases

These sample visualisations are only a preliminary illustration of the possibilities generated by the keyphrases extracted from one document. Additional ways of scaling keyphrases that stem from a search query over a document set need to be explored further. Also, whether keyphrases provide interesting text snippets that can be used for visualisations of text summaries and so on.

[30] According to [Witten et al 2000] the substring deletion should be provided by KEA, this might be a programming bug in KEA.

6.2.1. *Granularity of scores*

The article on fracking contains 936 words, the score distribution of the 500 key phrases is the following:

Score	number of key phrases
0.0502	1
0.034	12
0.0339	40
0.0338	447

Table 5: Score distribution for sample text

The granularity of the scores has been examined further since LUH has pointed out that too coarsely grained scores do not provide adequate differentiation for tag clouds. Therefore experiments have been conducted that measure the score diversity and precision/recall/f-score based on increased training corpus size and different training corpora.

6.2.2. *Experiments with different training corpus size*

In the first set of experiments, we have used the DW corpus EN and keyphrase annotations stemming from the GFAI tool for keyphrase extraction. The training corpus size is increased from 10 documents to 1418 documents. The test corpus contains 500 documents, the maximum number of extracted keyphrases per document is 50. Since the number of keyphrases per document greatly varies in the gold standard, the number of keyphrases extracted by KEA has been reduced to the number annotated in the gold standard for the calculation of precision, recall and F-score. Thus each document in the evaluation has the same number of keyphrases in the test set and in the gold standard annotation. The reduction yields a more balanced precision/recall. Score frequency has been computed without keyphrase reduction.

Table 6 lists average precision, recall and F-score as well as the scores of the keyphrases and their average frequency per document. It can be seen that the training corpus size does not influence much precision, recall and F-score, the highest F-score is achieved with training corpus size 40. With larger training corpora the F-score decreases slightly. Even a training corpus of 10 documents achieves a comparable F-score. The granularity of the scores is poor. It increases with training corpus size but only the training corpus with 1418 documents yields a score distribution with three different scores. Since this is still a rather poor score distribution, it has been decided to include the rank of the keyphrases as expressed in the order in the keyphrase list as additional sorting criterion in the UIMA annotations.

Train corp	Precision	Recall	f-score	Scores	Av score freq per doc
10	0.39898324	0.3343996	0.36384773	0.3588	36.392
40	0.4025773	0.3378866	0.36740607	0.3376	36.462
50	0.39717245	0.33246785	0.36195114	0.3303	36.452
100	0.39592987	0.3313358	0.36076427	0.3013 0.2834	18.044 18.444
500	0.38470188	0.32007608	0.34942600	0.2759 0.2576	5.99 30.534
1000	0.38342398	0.31875992	0.34811450	0.2798 0.2662	8.86 27.664
1418	0.38355327	0.31890193	0.34825248	0.2864 0.2745 0.2714	8.384 16.326 11.798

Table 6: Precision, recall, F-score and score distribution relative to training corpus size using DW corpus

In the second experiment, we have used the Guardian EN corpus. Here the author-assigned keywords are used as annotations, without checking if they occur in the text body. Thus the annotated keyphrases can be considered very noisy. Otherwise the same settings are used as in the DW experiment. The test corpus contains 500 documents, the maximum number of extracted keyphrase per document is 50. The training corpus size is increased from 40 documents to 1000 documents.

Again, the F-score is best with a smaller training set (50 documents). The score diversity seems to increase again with increasing training corpus (even though the 50 document training corpus is an outlier). Interestingly, the score diversity achieved with the Guardian training corpus is higher than the one achieved with the DW corpus. The reasons for this are not fully investigated yet.

Train corp	Precision	Recall	f-score	Scores	Av score freq per doc
50	0.11718188	0.13437416	0.12519053	0.1759 0.0330 0.0289 0.0239	8.47 1.126 33.68 1.284
100	0.110152654	0.12645356	0.11774159	0.1193 0.0287	9.732 34.828
1000	0.099848000	0.11449652	0.10667171	0.1025 0.0527 0.0480 0.0449 0.0332 0.0320 0.0309 0.0271 0.0265 0.0255	6.132 5.176 1.756 0.598 1.326 0.414 0.006 0.002 26.834 2.226

Table 7: Precision, recall, F-score and score distribution relative to training corpus size using Guardian corpus

In the third experiment the models trained with the Guardian corpus are used to perform keyphrase extraction in the DW test set and evaluate it against the gold standard used in the first experiment. Interestingly, the F-scores are comparable to the ones achieved with DW training data while maintaining the higher score diversity. Thus, KEA seems quite robust regarding the quality of the keyphrase annotations in the training data.

Train corp	Precision	Recall	f-score	Scores	Av score freq per doc
50	0.3823838	0.31759396	0.34699040	0.1759 0.0330 0.0289 0.0239	4.942 0.302 30.044 1.126
100	0.37888196	0.31406780	0.34344375	0.1193 0.0287	5.304 31.108
1000	Data deleted by mistake				

Table 8: Precision, recall, F-score and score distribution relative to training corpus size using Guardian as training corpus and DW as test corpus

6.2.3. *Author-annotated keyphrases versus tool-based keyphrase annotations*

The impact of author-annotated keyphrases versus GFAI-tool based annotations for training has been measured with the help of the English DW corpus. In the first experiment the author-annotated keyphrases have been used for training and as gold standard and in the second experiment the same DW-corpus split into test and training set has been used with keyphrases annotated by the GFAI tool. Table 9 lists precision, recall and F-score for the two experiments:

	Precision	Recall	F-Score
author annotation	0.26121905	0.26162530	0.2614220
GFAI tool annotation	0.33884310	0.33680326	0.3378201

Table 9: Precision, recall and F-score for author-annotated and GFAI-tool annotated training data

The experiment using the GFAI tool for key phrase annotation yields higher precision, recall and f-score. The reasons for the difference are not fully investigated yet. The comparison is taken as further indication that KEA is fairly robust regarding the choice of keyphrase annotations. Thus, if no author-annotated training data is available, the GFAI tool can be used to annotate the training data.

6.2.4. *Topic diversity*

A precursory look at the English DW corpus and the Guardian corpus has shown that some text types such as speeches and interviews contain highly divergent topics. To put a number to it, out of the 120 Guardian documents that contain the term *fracking*, only 51 documents are assigned the keyphrase *fracking* when the number of extracted keyphrases is set to 40. Here more studies are needed whether the documents that contain the term *fracking* but are not assigned the keyphrase *fracking* are interesting for search on *fracking* or not. Even if *fracking* is not the major topic in these texts, it might still be interesting for journalists to explore *all* contexts in which *fracking* occurs. In this context, a search-request based text summarization tool could be built, see e.g. [Jones et al. 2002] for a discussion of interactive text summarization tools.

6.2.5. *GFAI tools for linguistic analysis and keyword extraction*

The DW corpus DE and EN has been analysed with linguistic tools available at GFAI. The goal was to automatically assign keyphrases to the documents in the DW corpus.

The GFAI keyphrase extraction tool is based on terminology extraction routines which, in turn, are based on a linguistic analysis of the input text using a morphological analyser to determine word forms[31] and shallow parsing methods to achieve disambiguation[32]. As the result of terminology extraction, a set of terms is obtained with each term being assigned its basic statistic values: term frequency, number of documents in which a term occurs.

[31] Morphological analysis is performed by the MPRO tool developed at IAI, see [Maas et al 2009]

32 Shallow parsing is performed using the FRED (formerly KURD) formalism, see [Carl et al. 1998]

The following table shows the overall number of documents and of terms extracted per language:

	EN	DE
Documents	225.816	457.377
Terms	1.050.000	1.265.000

Table 10: Terms extracted per language

Based on its frequency in a given document (***tf***), each term is assigned a local weight normalised on a scale of 0 to 100. In addition, the inverse document frequency is computed (***idf***) per term, thus putting the number of documents in which a term occurs in relation to the overall number of documents in the document collection[33]. Based on ***tf*** and ***idf***, each term in a given document is assigned a global weight, again normalised on a scale of 0 to 100. The global weight is used to define a threshold for the display of the most prominent terms of a given document. With respect to the overall document collection, the global weight is used to separate relevant from irrelevant terms by specifying a respective threshold. Terms that have a global weight that is below the defined threshold in any document of the document collection are considered irrelevant. These are typically terms that either occur very rarely (just once) in the document collection or otherwise occur in very many documents.

Table 11 shows the split between relevant and irrelevant terms that is achieved by requiring a relevant term to occur at least twice in the document collection with a global weight of at least 30 and at least 50:

	dfi≥2 & w_{glob}≥30		dfi≥2 & w_{glob}≥50	
	EN	**DE**	**EN**	**DE**
relevant	279.026 (ca 26,6%)	459.658 (ca. 36,3%) (11.110 Verbs)	259.259 (ca. 24,7%)	411.000 658 (ca. 32,5%) (10.867 Verbs)
irrelevant	770974 (ca 73,4%)	805342 (ca 63,7%)	790741 (ca 75,3%)	854000 (ca 66,5%)

Table 11: Relevant and irrelevant terms extracted per language

Discarding all irrelevant terms, each relevant term is finally assigned a relevance weight that is, once more, normalised on a scale of 0 to 100. The relevance weight takes into account the average global weight of a term as well as the ratio between relevant occurrences of a term and its overall number of occurrences. Terms with a

[33] See [Salton et al.1988]

w_{glob}≥30 have been used as keyphrase annotations for KEA model training. For tag clouds the higher threshold of w_{glob}≥50 is better.

6.3. Relation extraction

Currently KEA is used for rudimentary relation extraction. The multi-word keyphrases that contain verbs specify rudimentary relations (without role labels). The advantage of using KEA keyphrase extraction is that it can be trained for all four EUMSSI languages on unannotated data, such as the one that we are already collecting within EUMSSI. No additional data is necessary.

In later project stages, we plan to complement the keyphrase-strategy with more sophisticated tools. A potential candidate tool for relation extraction is the semanticRoleLabeler[34] from ClearNLP, see [Choi 2012], which has a UIMA integration[35]. This tool needs a dictionary, a part-of-speech tagging model, a dependency parsing model, and a semantic role labeling model. Such resources may not be ready available for all four languages.

6.4. Event extraction

In event semantics, events take individuals as arguments and the argument relations can be labelled with semantic roles. Hence, this concept of event extraction is close to relation extraction.

The term event detection is used in the context of information streams. Here the challenge is to detect when new events such as a disease outbreak starts. This aspect of event detection is closer to trend detection, which is part of the Social Media analysis performed in tasks 4.4 and 4.5.

6.5. Topic detection

Currently the KEA keyphrases are used to determine document-specific topics. More specific topic models are useful for analysing the contents of large document collections. In addition, advanced topic models such as Latent Dirichlet Allocation (LDA) [Blei et al 2003] can be used to determine multiple topics per document, and Pachinko Model Allocation (PAM) has the added benefit of providing relations between topics, see [Wei et al 2006]. These topic models are also available in a public domain toolkit. The MALLET toolkit[36] contains implementations of Latent Dirichlet Allocation, Pachinko Allocation, and Hierarchical LDA.

[34] http://clearnlp.wikispaces.com/semLabeler. For a Question/Answering application using the semanticRoleLabeller see [Maqsud et al 2014].

[35] UIMA integration: http://mvnrepository.com/artifact/de.tudarmstadt.ukp.dkpro.core/de.tudarmstadt.ukp.dkpro.core.clearnlp-asl/1.6.2

[36] McCallum, Andrew Kachites (2002). MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu

The disadvantage of topic models such as LDA and PAM is that they are computed relative to a document collection and do not adapt to new topics. An exception is Relevance Modelling (RM), see [Lavrenko et al 2001]. RM does not need any training data but builds on the query alone. See also [Yi et al 2009] for a comparison of the different topic models when used in an information retrieval task. Their finding is that in document retrieval Relevance Modelling performs best. Within the EUMSSI context, LDA and PAM are promising as well, since they provide fine-grained topic information for individual documents which supports text mining tasks.

6.5.1. ***Impact***

A first version of key phrase extraction is running for all four EUMSSI languages. Keyphrases provide the basis for relation extraction, event extraction, topic modelling, text summarizsation and text similarity calculations.

6.5.2. ***Next steps***

Next steps are to improve, extend and evaluate the usage of KEA and to explore other, more specific tools for topic modelling and relation extraction:

1. Improving, extending and evaluating KEA:
   a. Output all surface forms instead of only the most frequent surface form
   b. Use GFAI keyphrase extraction tool to annotate more corpora.
   c. Create in-domain corpora.
   d. Train models on different corpora.
   e. Examine the optimal ratio of number of assigned keyphrases per document in relation to document length.
   f. Examine how many paragraphs are assigned how many keyphrases per document under which parameter settings. This is interesting with regard to interviews and speeches that contain divergent topics. Paragraphs that are not assigned any keyphrases (or other labels) are invisible to the search algorithm.
   g. Possibly: the usage of ontologies as KEA parameter.
2. Explore whether an interactive, search-request based text summarization tool is of interest.
3. Explore topic models that are provided by the MALLET tool kit such as LDA.
4. Explore semantic role labelling as provided by ClearNLP for relation extraction.

7. SENTIMENT ANALYSIS COMPONENT

7.1. Introduction

Sentiment or opinion corresponds to the mental or emotional state of the writer or speaker or some other entity referenced in the discourse. News articles, for example, often report emotional responses to a story in addition to the facts. Editorials, reviews, weblogs, and political speeches, as well as comments to news, posts to on-line forums, product reviews or tweets, convey the opinions, beliefs, or intentions of the writer or speaker. Accordingly, Sentiment Analysis, or Opinion Mining, as it is also known, are a set of computational techniques for extracting, classifying, understanding, and assessing the opinions expressed in news sources, social media comments, and other user-generated contents. One of the most common sub-tasks of opinion mining is polarity classification. That is, given an opinionated piece of text about one single topic, classify the opinion as falling under one of two opposing sentiment polarities, or locate its position on the continuum between these two polarities.

Opinion Mining (OM) is generally regarded as being a very complex task even at its more basic level of polarity classification. The reason is that sentiment and subjectivity are expressed through subtle linguistic resources, including irony. Moreover, they tend to be very domain and context dependent, getting to the point that even the exact same word or phrasing can indicate different sentiments in different contexts. An example that has become popular in the literature on Opinion Mining is: "go read the book", which most likely indicates positive sentiment for book reviews, but negative sentiment for movie reviews.

As the scenarios described in D2.1 and D7.1 indicate, detection of opinion may be a useful feature of the demonstrators that EUMSSI intends to build. In one of these scenarios, the journalist needs to consider all perspectives and opinions around a controversial topic such as fracking.

7.2. Approaches to OM: Rule-based and Corpus-based

The literature agrees on two main approaches for classifying opinion expressions: (i) use of supervised learning methods and (ii) application of lexical or rule based knowledge (see [Liu, 2012] for an overview). In principle, the latter yields better precision while the former is able to discover unseen examples and thus enhances recall. [Rodriguez-Penagos et al, 2013] have experimented with a combination of both on noisy data, to arrive to the conclusion that used in isolation, lexical based methods outperform machine learning, and used in combination, contribute the most.

7.2.1. ***Rule-based approaches***

7.2.1.1. Keyword spotting

Keyword spotting is the most naïve approach and probably also the most popular because of its accessibility and cost efficiency. In this technique, text is classified into *affect* categories, based on the presence of fairly unambiguous affect words like 'happy', 'sad', 'afraid', and 'bored'.

The simplest keyword-spotting approach consists of annotating text by using a language specific polarity lexicon. Given this lexicon, the polarity of a document is annotated by combining the polarity of its constituent sentences, where in turn the polarity of a sentence is determined as a summation of the polarity of the words found in the sentence. Alternative to summation is counting: a number of negative words above a particular threshold renders the document negative, whereas a majority of positive words triggers a positive classification.

One of the main weaknesses of this approach is related to the use of negation, which can actually reverse the polarity of the opinion. A second weakness is its reliance on the presence of obvious affect words in the surface textual form.

7.2.1.2. Heuristic rules for Polarity detection

In order to overcome the limitations of simple keyword spotting, most rule-based systems use some kind of heuristics to predict polarity. Handling negation is an important concern in opinion and sentiment related analysis, as it can reverse the meaning of a statement. Such task, however, is not trivial as not all appearances of explicit negation terms reverse the polarity of the enclosing sentence and that negation can often be expressed in rather subtle ways, e.g., sarcasm and irony, which are quite difficult to detect.

To supplement the lexical information provided by a polarity lexicon, the following clues are commonly used to detect sentiment in a text: special punctuation, complete upper-case words, cross-linguistic onomatopoeias, exclamation words, degree adverbs, and emoticons.

Recent studies have underlined that position is particularly relevant in the context of sentiment summarisation. In particular, in contrast to topic-based text summarisation, where the incipits of articles usually serve as a strong baseline, the last n sentences of a review have been shown to serve as a much better summary of the overall sentiment of the document, and to be almost as good as the n (automatically-computed) most subjective sentences.

Below is an example of heuristic rule taken form [Rodriguez-Penagos et al, 2013] that takes into account not only polarity value of individual words, but also effect of quantifiers and negation.

```
if has polar word(CUE) then
     polarity = lex(P)-0.5*lex(QP)-lex(N)+0.5*lex(QN)
     if polarity>0 then
          if has negation(CUE) then negative
          else positive
          end if
     else if polarity<0 then
          if has negation(CUE) then positive
          else negative
          end if
     else
          if has negation(CUE) then positive
          else negative
          end if
     end if
```

```
else if has negation(CUE) then negative
else
     polarity= tlex(P)-0.5*tlex(QP)-tlex(N)+0.5*tlex(QN)
     if polarity<0 then negative
     else if tlex(NEU)>0 then neutral
     else if polarity>0 then positive
     else if has negemo(CUE) then negative
     else if has posemo(CUE) then positive
     else unknown
     end if
end if
```

Figure 21: Heuristic polarity rule

7.2.2. *Corpus-based approach*

The most straight-forward approach for corpus-based document annotation is to train a machine learning classifier, assuming that a set of annotated data already exists. Experiments demonstrate that if enough training data are available, it is relatively easy to build more or less accurate sentiment classifiers. The main drawback of this method is the scarcity of such a resource, or total lack thereof. One usual way to overcome this difficulty is to use rule-based methods to annotate a corpus in an unsupervised manner and then use the annotated corpus to train the model.

7.3. Scope of polarity and type of text in EUMSSI

Opinions and sentiments do not occur only at document level, nor are they limited to a single valence or target. Contrary or complementary attitudes toward the same topic or multiple topics can be present across the span of a document. However, a certain degree of continuity in subjectivity labels of adjacent sentences may be assumed, as an author usually does not switch too frequently between being subjective and being objective, particularly in texts with limited length such as the ones we plan to tackle within EUMSSI.

More specifically, within EUMSSI, we aim at extracting sentiment from statements expressed by an entity or person with respect to a given topic, both in news text and in user-generated content from Social Media.

7.3.1. *News text*

We want to detect the author's position in news text. For this, we are going to classify for polarity **quoted statements**, i.e. quotations that are attributable to a given person or organisation. E.g.:

In his State of the Union address tonight President Obama said: "We produce more natural gas than ever before – and nearly everyone's energy bill is lower because of it.".

7.3.2. ***User-generated content from Social Media***

We plan to analyse opinion from two main Social Media sources: Twitter and You Tube. That is, the following text types:

a. tweets, i.e. short messages, not longer than 140 characters, sent by the micro-blogging platform. E.g.:

> #Fracking? Not in my #backyard!

b. comments on YouTube

For the sake of simplicity, we are going to assume that both single tweets and single quotations encompass just **one opinionated unit**, with one single value of polarity that refers to the pre-defined topic of the tweet or quotation.

7.4. Preliminary work: Benchmarking of Polar Lexicons

Much of the research work to date on sentiment and subjectivity analysis has been applied to English, with much lesser efforts being invested in other languages. However migrating resources from one language to another has been proven feasible as explained by [Banea et al, 2011].

In EUMSSI, we plan to start developing a system for English and then migrating strategies -and if necessary resources- to the other three languages. Our first step will be to perform a comparative evaluation of three of the most popular existing Polar Lexicons in English using a simple Keyword Spotting approach, against a Gold Standard corpus.

The results of this exercise will be threefold:

1. It will allow us to build an analysis pipeline or processing chain and integrate it into the EUMSSI platform.
2. It will also help us build an initial baseline, as well as an evaluation framework able to monitor progress of the EUMSSI Sentiment Analysis component.
3. It may potentially set the basis for an improved lexical resource resulting from a weighed merging of the existing resources.

We have used three different lexicons: OpinionFinder, SenticNet and LIWC. Each lexicon has its own way to express polarity. We have made an effort to map the different encodings into a common framework: a three value system (POSITIVE, NEGATIVE, NEUTRAL), such as the one used by RepLab (see section 7.4.5).

7.4.1. ***OpinionFinder***

Opinion Finder [Wiebe and Riloff, 2005] is a lexicon compiled from a manually annotated corpus. It contains 8222 entries. Verbs are lemmatized but nouns are given in full-form. The entries in the lexicon have been labelled for part of speech, as well as for reliability – those that appear most often in subjective contexts are strong clues of subjectivity, while those that appear less often, but still more often than expected by chance, are labelled weak. Each entry is also associated with a polarity label, indicating whether the corresponding word or phrase is positive, negative, or neutral. As illustration, consider the following entry from the OpinionFinder lexicon:

```
type=strongsubj    len=1    word1=agree    pos1=verb    stemmed1=y
priorpolarity=positive
```

Figure 22: OpinionFinder lexical entry

This lexical entry indicates that the word “agree”, when used as a verb, is a strong clue of subjectivity and has a positive polarity.

These values have been mapped, as a first step towards convergence to a common evaluation framework, into numeric values, following this rationale:

```
Positive + strongsubj=1
Positive + weaksubj=0.5
Neutral=0
Both=0
Negative + strongsubj=-1
Negative+weaksubj=-0.5
```

Figure 23: Mapping of OpinionFinder polarity information onto numeric values

7.4.2. ***SenticNet***

SenticNet [Cambria et al, 2010] is a resource for opinion mining that aims to create a collection of commonly used ‘polarity concepts’, that is, concepts extracted from ConceptNet [Havasi et al, 2007], with relatively strong positive or negative polarity. It is encoded in RDF/XML format on the base of HEO (Human Emotions Ontology), a high level ontology for human emotions that supplies the most significant concepts and properties which constitute the centrepiece for the description of every human emotion.

SenticNet was inspired by SentiWordNet[37], a lexical resource in which each WordNet synset is associated to three numerical scores describing how objective, positive and negative the terms contained in the synset are. However, differently from SentiWordNet (which also includes null polarity terms), SenticNet does not contain concepts with neutral or almost neutral polarity, i.e., concepts with polarity magnitude close to zero. SenticNet is freely available online and currently contains more than 5,700 polarity concepts (nearly 40% of the Open Mind corpus [Stork, 1999]), many of which are multiword, such as ‘accomplish goal’, ‘bad feeling’, etc.

This is the format of the entry “agree” in SenticNet:

```
<Description about="http://sentic.net/api/en/concept/agree">
          <type resource="http://sentic.net/api/concept"/>
          <text xmlns="http://sentic.net/api">agree</text>
          <semantics xmlns="http://sentic.net/api"
resource="http://sentic.net/api/en/concept/reach_tentative_agree
ment"/>
```

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

```
          <semantics xmlns="http://sentic.net/api"
resource="http://sentic.net/api/en/concept/thank"/>
          <semantics xmlns="http://sentic.net/api"
resource="http://sentic.net/api/en/concept/comfort_friend"/>
          <semantics xmlns="http://sentic.net/api"
resource="http://sentic.net/api/en/concept/forgive"/>
          <semantics xmlns="http://sentic.net/api"
resource="http://sentic.net/api/en/concept/take_oath"/>
          <pleasantness xmlns="http://sentic.net/api"
datatype="http://www.w3.org/2001/XMLSchema#float">0.723</pleasan
tness>
          <attention xmlns="http://sentic.net/api"
datatype="http://www.w3.org/2001/XMLSchema#float">0</attention>
          <sensitivity xmlns="http://sentic.net/api"
datatype="http://www.w3.org/2001/XMLSchema#float">0</sensitivity
>
          <aptitude xmlns="http://sentic.net/api"
datatype="http://www.w3.org/2001/XMLSchema#float">0.98</aptitude
>
          <polarity xmlns="http://sentic.net/api"
datatype="http://www.w3.org/2001/XMLSchema#float">0.568</polarit
y>
     </Description>
```

Figure 24: SenticNet lexical entry

For computation of polarity, we consider only the floating number value of the ‘polarity’ feature.

7.4.3. ***LIWC***

The LIWC2007 Dictionary [Pennebaker et al., 2001] is composed of 2,290 words and word stems. Each word or word-stem defines one or more word categories or subdictionaries. For example, the word 'cried' is part of four word categories: *sadness, negative emotion, overall affect*, and a *past tense verb*. Hence, if it is found in the target text, each of these four subdictionary scale scores will be incremented. As in this example, many of the LIWC2007 categories are arranged hierarchically. All anger words, by definition, will be categorized as ‘negative emotion’ and overall ‘emotion’ words. Each of the 74 preset LIWC2007 categories is composed of a list of dictionary words that define that scale. Find below the entry “agree” for illustration, together with the labels for the relevant category values:

```
agree      125   126   462
[ 125: affec; 126: posemo; 462: assent]
```

Figure 25: LIWC lexical entry

The categories from the table that have been considered for polarity are the following:

- *Posemo, Posfeel* and *Optim* are mapped into numerical value 1
- *Negemo, Anx, Anger, Sad, Death* and *Swear* are mapped into -1.
- The rest are considered neutral and therefore mapped into 0.

The LIWC dictionary is also available Spanish and German. French, Italian and Dutch versions are currently under development.

7.4.4. *Computation of polarity*

For this experiment, we try two different methods of computing polarity of a given textual unit (i.e. a tweet or a quotation):

- summation of the values of all the polar words and mapping to the three-value system: e.g. NEGATIVE<0; NEUTRAL=0; POSITIVE>0. However, given that SenticNet has continous values, the following mapping seems more accurate:

  - NEGATIVE < -0.5
  - POSITIVE > 0.5
  - -0.5 < NEUTRAL < 0.5

Figure 26: Mapping of SenticNet polarity information onto numeric values

- computation of the ratio of positive words vs negative words.

The two methods give identical results for LIWC and the second performs slightly worse for the other two.

7.4.5. *Evaluation framework*

We have compared the performance of the three resources on Twitter text, using the RepLab 2013 dataset as Gold Standard [Amigó et al., 2013]. Filtering for language and number of opinionated units per tweet (we keep only those with a unique polarity), as well as accounting for those that cannot be crawled because they have disappear from the Network, we end up with a total of 74888 tweets in English, annotated for polarity.
For this experiment we have used a simple strategy: keyword spotting and sum of polarities, with no use of heuristic rules. For each one of the results we compute its confusion matrix and based on these values, the corresponding metrics.
The confusion matrix reflects the number of times that the predicted class and the known class (according to a groundtruth or gold standard) are confused or mistaken by the system.

tpPS: true positives for POSITIVE
tpNG: true positives for NEGATIVE
tpNT: true positives for NEUTRAL
ePSNG: errors where POSITIVES have been classified as NEGATIVES
ePSNT: errors where POSITIVES have been classified as NEUTRALS
eNGPS: errors where NEGATIVES have been classified as POSITIVES
eNGNT: errors where NEGATIVES have been classified as NEUTRALS
eNTPS: errors where NEUTRALS have been classified as POSITIVES
eNTNG: errors where NEUTRALS have been classified as NEGATIVES

Figure 27: Confusion matrix

This is a case of a three-class classification: *positive*, *negative* and *neutral*. However, in the real task, we are actually more interested in two of these classes: *positive* and *negative*, which, together, constitute a class by themselves, namely *opinionated*.
With the information that the confusion matrix provides us, we are going to assess the three systems in two aspects:

1. How well does the system detect **opinionated tweets**, in terms of precision, recall and f-measure. With this purpose, we are going to split our search space into two classes NEUTRAL and OPINIONATED, which contains both the POSITIVE and NEGATIVE class.

    Precision in Detection = tpOP/(tpOP + fpOP)
    Recall in Detection = tpOP/(tpOP + fnOP)
    F-measure in Detection = 2 x ((Precision x Recall) / (Precision + Recall))
    where:
    True Positives of OPINIONATED: tpOP=tpPS+tpNG+ePSNG+eNGPS
    False Positives of OPINIONATED: fpOP=eNTPS+eNTNG
    False Negatives of OPINIONATED: fnOP= ePSNT+eNGNT

    Figure 28: Precision / recall / f-measure for detection of Opinionated

2. How accurate is the system in deciding whether a certain opinionated tweet has **negative** or **positive polarity**.

7.4.6. ***Analysis of results***

The following tables summarize the results obtained. Table 12 shows the results obtained by the three systems for precision, recall and f-measure in the task of detecting opinionated tweets. As we can see, precision is quite similar for the three systems but recall is much lower in two of them, being Opinion Finder the system that obtains the best value for the combined metrics F-measure.

	Opinion Finder	**SenticNet**	**LIWC 2007**
Precision in detection %	73.33	**75.29**	75.18
Recall in detection %	**68.50**	37.02	33.66
F-measure in detection %	**70.83**	49.64	46.50

Table 12: Detection of opinionated tweets

Table 13 shows precision and recall in classifying positive and negative tweets within the group of opinionated tweets. Here the results are less clearly cut. What can be observed is that all systems have more trouble in classifying NEGATIVE tweets than POSITIVE ones, as deduced from the low F-measures obtained for this class. Interestingly, LIWC2007's precision for negative beats its precision for positive, although recall is still very low.

		Opinion Finder	**SenticNet**	**LIWC 2007**
Precision %	POS	76.56	**91.36**	56.93
	NEG	50.34	25.71	**65.50**
Recall %	POS	**86.92**	85.67	86.39
	NEG	33.25	**37.99**	28.38
F-measure %	POS	81.41	**88.42**	68.67
	NEG	**40.05**	30.67	39.60

Table 13: Precision and Recall for polarity classification (positive/negative) for opinionated tweets

7.5. Next steps

These are the next issues that we plan to tackle:

- Implement an extractor of quotations (direct and certain cases of indirect speech).
- Improve our baseline Sentiment Analysis system for English, by:
  - Merging existing lexicons.
  - Including heuristic rules to deal with negation and quantifiers.
  - Use our enhanced rule-based system to automatically annotate training corpus for a corpus-based approach.
- Test enhanced Sentiment Analysis on other text types: You Tube comments, quotations extracted from news.
- Build Sentiment Pipelines for ES, DE and FR, based on research results on EN, using native resources[38] and if necessary expanded with resources transferred from English, following methodologies presented in by [Banea et al, 2011].

[38] Existing lexical resources in languages other than English: SentiMerge (DE) and LIWC (FR, ES). Existing Gold Standard corpora: Semeval (EN), RepLab (EN, ES) and SenTube (EN, ES).

8. ANALYSIS OF OTHER TEXT TYPES: OCR, ASR

Preliminary experiments with other text types such as the output of optical character recognition (OCR) and automatic speech recognition (ASR) have been conducted.

8.1. Optical Character Recognition (OCR)

OCR provides an n-best candidate list with attached scores. Hence, cross-media inference mechanisms can use information from other text types to enhance the OCR output by re-ranking the candidate list.

Example: Four hypotheses for the input "*Fracking Angst*". The forth, lowest ranking hypothesis is correct. Here cross-media inferences can improve text analysis.

```
<VideoText id="vd1" ...>
 <Location line="246" column="128" height="22" width="135" />
        <Hyp1 score="45.081" count="41">Fucking Angst</Hyp1>
        <Hyp2 score="8.417" count="6">I- racking Angst</Hyp2>
        <Hyp3 score="7.013" count="5">F- racking Angst</Hyp3>
        <Hyp4 score="7.013" count="2">Fracking Angst</Hyp4>
 <img src="frame_0000000036"/>
```

Figure 29: Candidate hypotheses of OCR

A major feature of OCR output is that it does not contain full sentences and that names occur with minimal or no context. These properties do not pose any major problems for the statistical tools used in EUMSSI.
A precursory analysis shows that Stanford NER performs reasonably well on OCR output if the capitalization is correct. Applying KEA leads to key phrases that are possibly too small, e.g. the string "New York" is assigned three key phrases: "New", "York" and "New York".

8.1.1. *Impact*

N-best lists of OCR output and DBpediaSpotlight analysis allow for cross-media enhancement of OCR output and Named Entity Linking.

8.1.2. *Next steps*

1. Setting up the process chain.
2. Closer examination of OCR output.

8.2. Automatic Speech Recognition (ASR)

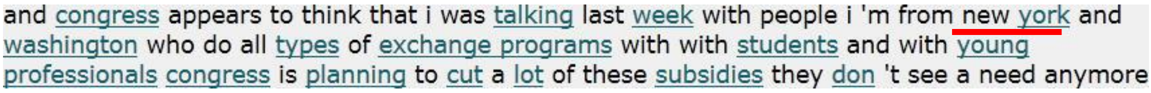
The ASR output has three properties that are potential problems for text processing tools:

1. ASR output lacks punctuation marks.
2. ASR provides noisy, possibly ungrammatical text.
3. ASR output lacks upper case letters, i.e. all letters are in lower case.

Some preliminary text analysis experiments indicate that the lack of punctuation marks does not cause any serious problems, probably because the text processing tools chosen in EUMSSI are predominantly based on statistical methods which do not depend on sentence boundaries. ASR provides chunks the continuous text separated by line breaks. These chunks can be processed by the text analysis tools. Noisy, possibly ungrammatical text does not pose serious problems either.

The lack of upper case letters is more of a problem. E.g. the regular Stanford NER does not provide any acceptable results for lower case text because the models are case-sensitive. Here, training separate lower case models is necessary. For English, Stanford already provides a lower case model. For the other EUMSSI languages DE, FR and ES such models need to be trained with lower case training data that can be generated by lower casing the existing annotated training data. DBpediaSpotlight provides reasonable results for lower case text, however, a precursory look at the output suggests that multi-word names that contain regular words such as the "new" in "new york" seem to be problematic in that they are not recognized as being part of the name. Here using Standford NER as a TrueCaser might help in improving the results. KEA keyphrase extraction handles text in lower case since it uses case folding.

Example of DBpediaSpotlight demo analysis with *new york* in lower case letters:



and congress appears to think that i was talking last week with people i 'm from new york and washington who do all types of exchange programs with with students and with young professionals congress is planning to cut a lot of these subsidies they don 't see a need anymore

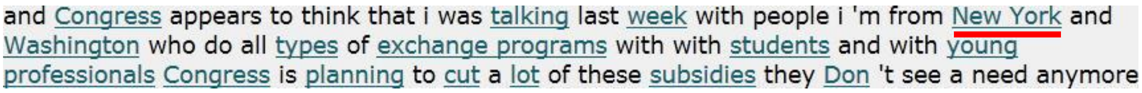
Figure 30: DBpediaSpotlight links for lower case input

Internal output for "york":

```
JCasResource
  sofa: _InitialView
  begin: 85
  end: 89
  similarityScore: 0.8622683393136671
  support: 10565
  types: "Schema:Place,DBpedia:Place,DBpedia:PopulatedPlace,DBpedia:Settlement"
  URI: http://dbpedia.org/resource/York
```

Figure 31: DBpediaSpotlight output for "york"

Example of DBpediaSpotlight demo analysis with New York in trueCased upper case letters: (All names recognized by StanfordNER are capitalized.)



and Congress appears to think that i was talking last week with people i 'm from New York and Washington who do all types of exchange programs with with students and with young professionals Congress is planning to cut a lot of these subsidies they Don 't see a need anymore

Figure 32: DBpediaSpotlight links for trueCased output

Internal output for “New York”:

```
JCasResource
   sofa: _InitialView
   begin: 81
   end: 89
   similarityScore: 0.8882072410057962
   support: 135979
   types:
"Schema:Place,DBpedia:Place,DBpedia:PopulatedPlace,DBpedia:Region,Schema:AdministrativeArea,DBpedia:AdministrativeRegion"
   URI: http://dbpedia.org/resource/New_York
```

Figure 33: DBpediaSpotlight output for “New York”

In summary, the strategies for processing lower case input text are:

1. Standford NER: use lower case models
2. DBpediaSpotlight: precede it with a TrueCaser to improve results
3. KEA: no measures needed

The experiments conducted so far have been done with English. It is expected that ES and FR can be treated in a similar way, since here capitalization is used in a similar way, i.e. it is mostly used for names. DE is different in that all nouns are capitalized. Here it has to be seen whether using Stanford NER as a TrueCaser is successful or whether other strategies need to be developed.

8.2.1. Impact

Using Stanford NER with lower case models as TrueCaser improves text analysis output for ASR for EN.

8.2.2. Next steps

1. Training lower case models for DE, FR, ES;
2. Developing special trueCasing strategies for DE if needed.

9. CONCLUSIONS AND NEXT STEPS

So far, a first prototype for text analysis with basic components for Named Entity Recognition, Named Entity Disambiguation, Named Entity Linking, Keyphrase Extraction and Sentiment Analysis has been set up for some or all four EUMSSI languages. Furthermore the applicability of the tools to other text types such as the output of OCR and ASR has been explored and remedies for potential problems such as lower case text have been outlined.

The workplan for the upcoming period includes:

- enhancement of the current modules following the lines expressed in the previous sections;
- cross-media and cross-tool information enhancement, (WP5) i.e.:
    - what useful information can we provide to audio, video or social media analysis?
    - In which way can the analysis results of different tools enhance each other?
- as more detailed application scenarios and visualisation devices (timelines, tag-clouds, search filters...) get developed in WP2 and WP6, be ready to provide whatever text annotations are necessary for these scenarios (e.g. text snippets, more fine-grained topic information, sematic role labelling, ...).

10. REFERENCES

Amigó, E., J. Carrillo-de-Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, et al. (2013). Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. CLEF '13, pp 333–352.

Banea, C., R. Mihalcea, and J. Wiebe (2011). Multilingual sentiment and subjective analysis. In: Multilingual Natural Language Processing.

Blei, D., A. Ng, M. Jordan (2003). Latent Dirichlet Allocation. Journal of machine Learning Research 3, pp 993–1022.

Cambria, E., Speer, R., Havasi, C., Hussain, A.: SenticNet: A publicly available semantic resource for opinion mining. In: AAAI CSK, pp. 14–18. Arlington (2010)

Carl, Michael, Antje Schmidt-Wigger (1998), Shallow Post Morphological Processing with KURD. In: 3rd International Conference on New Methods in Language Processing (NeMLaP'98), Sydney, Australia.

Choi, Jinho D. (2012). Optimization of Natural Language Processing Components for Robustness and Scalability. Ph.D. Thesis, University of Colorado Boulder, Computer Science and Cognitive Science.

Daiber, Joachim, Max Jakob, Chris Hokamp, Pablo N. Mendes (2013). Improving Efficiency and Accuracy in Multilingual Entity Extraction. In: Proceedings of the 9th International Conference on Semantic Systems (I-Semantics). Graz, Austria, pp 4–6.

Eckart de Castilho, R., I. Gurevych (2014). A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In: Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014, to be published, Dublin, Ireland.

Faruqui, Manaal, Sebastian Pado (2010). Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In: Proceedings of KONVENS.

Finkel, Jenny Rose, Trond Grenager, Christopher Manning (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In: Proceedings of the 43rd Annual Meeting of the ACL, pp 363–370, Ann Arbor.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, C. G. Nevill-Manning (1999). Domain-specific keyphrase extraction. In: Proceeding of 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, pp 668-673, San Francisco, USA: Morgan Kaufmann Publishers.

Gangemi, Aldo (2013). A Comparison of Knowledge Extraction Tools for the Semantic Web. In: The semantic web: semantics and big data, Proceedings of the 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, Springer, pp 351-366.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009). The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Havasi, Catherine, Rob Speer, Jason Alonso (2007). ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In: Proceedings of Recent Advances in Natural Language Processing.

Jones, Steve, Stephen Lundy, Gordon W. Paynter (2002), Interactive document summarisation using automatically extracted keyphrases. In: Proceedings of the 35th Hawaii International Conference on System Sciences.

Lavrenko, V., W. B. Croft (2001). Relevance-based language models. In: Proceedings of ACM SIGIR, pp 120–127.

Li, W., A. McCallum (2006). Pachinko Allocation: DAG-structured mixture models of topic correlations. Proceedings of ICML, Pittsburgh, PA, pp 577–584.

Liu, B., L. Zhang (2012). A Survey of Opinion Mining and Sentiment Analysis. In: C. C. Aggarwal, C. Zhai (eds.). Mining Text Data. Springer US, pp 415-463.

Maas, Heinz-Dieter, Christoph Rösener, Axel Theofilidis (2009). Morphosyntactic and Semantic Analysis of Text, the MPRO Tagging Procedure. In: Mahlow, C., M. Piotrowski (eds.). State of the Art in Computational Morphology. Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, Proceedings. Springer, Berlin, Heidelberg, New York.

Manning, Christopher D., Mihai Surdeanu, John Bauer (2014). The Stanford CoreNLP Natural Language Processing Toolkit, In: ACL 2014.

Maqsud, Umar, Sebastian Arnold, Michael Hülfenhaus, Alan Akbik (2014). Nerdle: Topic-Specific Question Answering Using Wikia Seeds. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, pp 81–85.

Medelyan, O., I. H. Witten (2005). Thesaurus-based index term extraction for agricultural documents. In: Proc. of the 6th Agricultural Ontology Service (AOS) workshop at EFITA/WCCA 2005, Vila Real, Portugal.

Medelyan, O., I. H. Witten (2006). Thesaurus Based Automatic Keyphrase Indexing. In: Proc. of the Joint Conference on Digital Libraries 2006, Chapel Hill, NC, USA, pp 296-297.

Mendes, Pablo N., Max Jakob, Andrés García-Silva, Christian Bizer (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proceedings of the 7th International Conference on Semantic Systems (I-Semantics). Graz, Austria.

Nothman, Joel, Nicky Ringland, Will Radford, Tara Murphy, James R. Curran (2012). Learning multilingual named entity recognition from Wikipedia. In: Artificial Intelligence 194, Elsevier, pp 151-175.

Pennebaker, James W., Martha E. Francis and Roger J. Booth. (2001). Linguistic inquiry and word count: LIWC 2001. Mahway: Lawrence Erlbaum Associates.

Rizzo, Giuseppe, Marieke van Erp, Raphael Troncy (2014). Benchmarking the Extraction and Disambiguation of Named Entities on the SemanticWeb. In: LREC 2014.

Rodríguez-Penagos, Carlos, Jordi Atserias, Joan Codina-Filbà, David García-Narbona, Jens Grivolla, Patrik Lambert, Roser Saurí (2013). FBM: Combining lexicon-based ML and heuristics for Social Media Polarities. In: Second Joint Conference on Lexical and Computational Semantics, Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp 483–489.

Salton, Gerard, Christopher Buckley (1988). Term-weighting approaches in automatic text retrieval. In: Information Processing and Management: an International Journal, Volume 24 Issue 5.

Stork, D. G. (1999). The Open Mind Initiative. In: IEEE Intelligent Systems & their applications 14(3), pp 19-20.

Strötgen, Jannik, Michael Gertz (2013). Multilingual and Cross-domain Temporal Tagging. In: Language Resources and Evaluation.

J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In Proceedings of the 6th International Conference on

Intelligent Text Processing and Computational Linguistics (CICLing-2005) (invited paper), Mexico City, Mexico, 2005.

Witten , Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, Craig G. Nevill-Manning (2000). KEA: Practical Automatic Keyphrase Extraction. Working paper series, The University of Waikato, ISSN 1170-487X.

Yi, Xing, James Allan (2009). A Comparative Study of Utilizing Topic Models for Information Retrieval. In: M. Boughanem et al. (Eds.): Advances in Information Retrieval, 31st ECIR, LNCS 5478, Springer-Verlag Berlin Heidelberg, pp 29–41.

Web references have been listed as footnotes throughout the document.

11. GLOSSARY

ASR: Automatic Speech Recognition
DKPro: Darmstadt Knowledge Processing Repository
DOW: Description of work document
DW: Deutsche Welle
EUMSSI: Event Understanding through Multimodal Social Stream Interpretation
GFAI: Gesellschaft zur Förderung Angewandter Informatik (Society for the Promotion of Applied Computer Science)
IDIAP: The Idiap Research Institute is an independent, non-profit, research foundation affiliated with Ecole Polytechnique Fédérale de Lausanne
KEA: Keyphrase Extraction Algorithm
LDA: Latent Dirichlet Allocation
LIUM: Laboratoire d'Informatique de l'Université du Maine (LE MANS)
LUH: Leibniz Universität Hannover
NED: Named Entity Linking
NEL: Named Entity Disambiguation
NER: Named Entity Recognition
OCR: Optical Character Recognition
UIMA: Unstructured Information Management Architecture
UPF: Universitat Pompeu Fabra
VSN: Video Stream Networks