1st International Workshop on Multimodal Media Data Analytics (MMDA 2016)

The rapid advancements of digital technologies, as well as the penetration of internet and social media usage have resulted in a great increase of heterogeneous multimedia data production worldwide. In many application fields (e.g. journalism, media monitoring) intelligent systems are required for supporting the stakeholders. In the recent years, research has directed towards the development of intelligent systems and tools dealing with media content analysis. However, the magnitude, the diversity and the heterogeneity of the data involved require novel intelligent techniques from the broad area of artificial intelligence to deal and extract meaningful interpretation and provide decision support and summarization services. This need is reflected also by relevant research projects such as MULTISENSOR and EUMSSI, which focus on providing multimodal analytics of heterogeneous data with an aim to support journalism, media monitoring, international investments and second screen applications.

In this context, MMDA 2016 welcomes novel research works focusing that deal with intelligent applications for media management, web content extraction, concept and event based indexing, semantic integration and retrieval, as well as multimodal fusion, content summarization and visual analytics.

The organizers would like to thank all the authors for submitting their papers and the members of the program committee for their valuable review contribution.

Workshop site
http://mklab.iti.gr/mmda2016/

The Organizers
S. Vrochidis, M. Melero, L. Wanner,
J. Grivolla, Yannick Esteve

Program Committee

S. Mille, UPF, Spain
S. Papadopoulos, CERTH-ITI, Greece
G. Gravier, IRISA, CNRS, France
I. Arapakis, EURECAT, Spain
B. Huet, EURECOM, France
R. Busch, LT, Germany
G. Casamayor, UPF, Spain
G. Thurmair, LT, Germany

List of Accepted Papers

- J. Grivolla, M. Melero and T. Badia, *EUMSSI: Multilayered analysis of multimedia content using UIMA, MongoDB and Solr*
- I. Gialampoukidis, D. Liparas, S. Vrochidis and I. Kompatsiaris, *Query-based Topic Detection Using Concepts and Named Entities*
- P.A. Broux, D. Doukhan, S. Petitrenaud, S. Meignier and J. Carrive, *An active learning method for speaker identity annotation in audio recordings*
- Y. Estève, S. Ghannay and N. Camelin, *Recent improvements on error detection for automatic speech recognition*
- S. Mille, M. Ballesteros, A. Burga, G. Casamayor and L. Wanner, *Multilingual Natural Language Generation within Abstractive Summarization*
- J. Codina-Filbà and L. Wanner, *Combining Dictionary- and Corpus-Based Concept Extraction*

Supported by:

EUMSSI: Multilayered analysis of multimedia content using UIMA, MongoDB and Solr

Jens Grivolla and **Maite Melero** and **Toni Badia** [1]

Abstract. Journalists, as well as users at home, face increasing amounts of data from a large variety of sources, both in professionally curated media archives and in the form of user-generated-content or social media. This provides a great opportunity at the same time as a great challenge to use all of this information, which EUMSSI approaches by providing semantically rich analysis of multimedia content, together with intuitive visual interfaces to explore the data and gain new insights.

We present a scalable platform that allows for distributed processing of large quantities of multimedia content. The EUMSSI platform provides support for both synchronous and asynchronous analysis processes and thus allows for *on-demand* services as well as long running batch processes. Analysis services for speech, video and text are integrated in the platform, as well transversal services that combine and enrich the existing outputs from various modalities. The EUMSSI platform builds on established open source projects such as UIMA, MongoDB and Solr and the project outcomes are published under permissive open source licenses.

The EUMSSI system is currently running and accessible through public demonstrators, incorporating hundreds of thousands of videos and news articles, as well as almost 10 million tweets.

1 Introduction

Nowadays, a multimedia journalist has access to a vast amount of data from many types of sources to document a story. In order to put information into context and tell his story from all significant angles, he needs to go through an enormous amount of records with information of very diverse degrees of granularity. Manually searching through a variety of different unconnected sources and relating all the disperse information can be time-consuming, especially when a topic or event is interconnected with multiple entities from different domains.

At a different level, many TV viewers are getting used to navigating with their tablets or iPads while watching TV, the tablet effectively functioning as a second screen, often providing background information on the program or interaction in social networks about what is being watched. However, this again requires an important effort either from the provider of a specific second screen application that includes curated content, or from the viewer at home who needs to identify potentially relevant sources of background knowledge and perform appropriate searches.

In the EUMSSI FP7 project[2] we have developed a system that can help both the journalist and the TV viewer by automatically analyzing and interpreting unstructured multimedia data streams and, with this understanding, contextualizing the data and contributing with new, related information.

The huge amounts of textual content available on the web and in news archives have been "tamed" over the last years and decades to some degree through the development of efficient, scalable search engines and improved ranking algorithms, and more recently by providing the user with more directly usable insights and answers, in addition to the traditional search result lists.

Tackling multimedia data in a similar way is a complicated endeavor, requiring the combination of many different types of analysis to bridge the gap from raw video recordings to semantically meaningful insights. It is now becoming computationally feasible to analyze large amounts of media content, and the EUMSSI project leverages the project partners' expertise in speech recognition, audio and video based person identification, text analysis or semantic inference to provide an integrated platform for large-scale media analysis and exploration.

In EUMSSI we have developed methodologies and techniques for identifying and aggregating data presented as unstructured information in sources of very different nature (video, image, audio, speech, text and social context), including both online (e.g. YouTube, Twitter) and traditional media (e.g. audiovisual repositories, news articles), and for dealing with information of very different degrees of granularity.

This is accomplished thanks to the integration of state-of-the-art information extraction and analysis techniques from the different fields involved (image, audio, text and social media analysis) in a UIMA-based multimodal platform. The multimodal interpretation platform continuously analyzes a vast amount of multimedia content, aggregates all the resulting information and semantically enriches it with additional metadata layers.

In this article we focus on the underlying platform that we developed using UIMA, MongoDB, Solr and other Open Source technologies to manage complex workflows involving online (on-demand) and offline (batch) processing, with mutual dependencies between the different modalities. After a brief introduction to the project objectives and some of the analysis technologies that are use for these aims we present the three main challenges of the core platform:

- Enabling the integration and combination of different annotation layers using UIMA and its CAS format
- Managing the processing workflow using MongoDB and UIMA
- Providing efficient and scalable access to the analyzed content for applications and demonstrators using Solr

[1] Universitat Pompeu Fabra, Spain, email: <firstname>.<lastname>@upf.edu

[2] Event Understanding through Multimodal Social Stream Interpretation: http://eumssi.eu

2 Project objectives

The goal of the EUMSSI project is to provide a complete system for large-scale multimedia analysis, including a wide range of analysis components working on different media modalities (video, audio, text). Additionally, we have developed two example applications (demonstrators) that build upon this platform with the goal to showcase the platform's potential, but also to lead towards a commercial exploitation of the project outcomes.

2.1 Multimodal analytics and Semantic Enrichment

For reasoning with and about the multimedia data, the EUMSSI platform needs to recognize entities, such as actors, places, topics, dates and genres. A core idea is that metadata resulting from analyzing one media helps reinforce the aggregation of information from other media. For example, an important issue in speech recognition is the transcription of previously unknown (out-of-vocabulary) words. This is particularly important when dealing with current news content, where person and organization names, and other named entities that may not appear in older training corpora, are among the most critical parts of the transcription. Existing text, tags and other metadata, as well as information automatically extracted from these sources, are used to improve and adapt the language models. Further, OCR on video data, speech analysis and speaker recognition mutually reinforce one another.

The combined and integrated results of the audio, video and text analysis significantly enhance the existing metadata, which can be used for search, visualization and exploration. In addition, the extracted entities and other annotations are exploited for identifying specific video fragments in which a particular person speaks, a new topic begins, or an entity is mentioned. Figure 1 illustrates some of the different layers of analysis that may exist for a video content item.

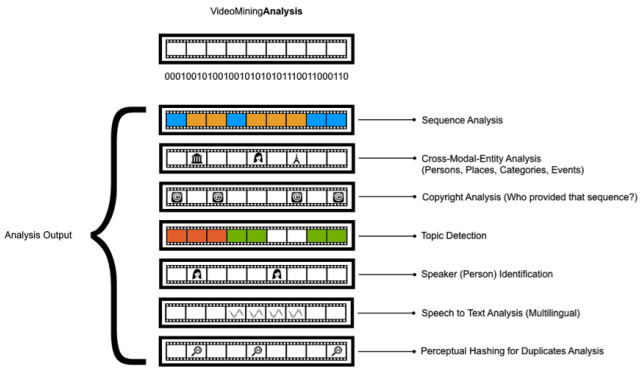

Figure 1. Video Mining Analysis

The EUMSSI system currently includes a wide variety of analysis components (many of which leverage and improve upon existing open source systems), such as automatic speech transcription (ASR), person identification (combining voice and face recognition, OCR on subtitles for naming, and Named Entity Recognition and Linking), and many natural language processing approaches, applied to speech transcripts as well as original written content or social media, e.g. NER (Stanford NLP), Entity Linking (DBpedia Spotlight), keyphrase extraction (KEA), quote extraction, topic segmentation, sentiment analysis, etc.

2.2 Assisted storytelling and second screen

Two application demonstrators have been implemented on top of the EUMSSI platform, each catering to a different use-case: (i) a computer-assisted *storytelling* tool integrated in the workflow of a multimedia news editor, empowering the journalist to monitor and gather up-to-date documents related with his investigation, without the need of reviewing an enormous amount of insufficiently annotated records; and (ii) a *second-screen* application for an end-user, able to provide background information or infotainment content, synchronized to what the user is currently watching. Figure 2 shows how both applications build on a common base of multimedia analysis and content aggregation/recommendation algorithms.

The storytelling tool (figure 3) provides a web interface that allows a journalist to work on an article in a rich editor. The system then analyses the text the journalist is writing in order to provide relevant background information. In particular, the journalist can directly access the Wikipedia pages of entities that appear in the text, or find related content in the archives (including content from outside and social media sources). A variety of graphical widgets then allow to explore the content collection, finding relevant video snippets, quotes, or presenting relevant entities and the relations between them.

The second screen application, on the other hand, is aimed at end users at home who would like to easily access background information, or have their viewing augmented with infotainment activities such as automatically generated quizzes relating to the currently viewed content.

Links to the publicly accessible demonstrators can be found on the EUMSSI web page at http://eumssi.eu/.

3 Architecture overview

The EUMSSI architecture was designed with a few core principles and requirements in mind:

- Simplicity: The platform should not be overly complex, in order to make it maintainable as well as to rapidly have a working system that all involved parties can build on
- Robustness: Failures, even hardware failures, should not have disastrous consequences
- Portability: It should be possible to easily migrate the platform to a different system
- Flexibility: It must be possible to quickly extend the platform, in particular by adding new analysis processes or content sources
- Scalability: The platform must be able to support large-scale content processing, as well as efficiently provide results to end users

As a result, the EUMSSI platform relies on open source technologies with a proven track record of reliability and scalability as its foundation.

The EUMSSI platform functions as a set of loosely coupled components that only interact through a common data back-end (MongoDB) that ensures that the system state is persisted and can be robustly recovered after failures of individual components or even the whole platform (including hardware failures).

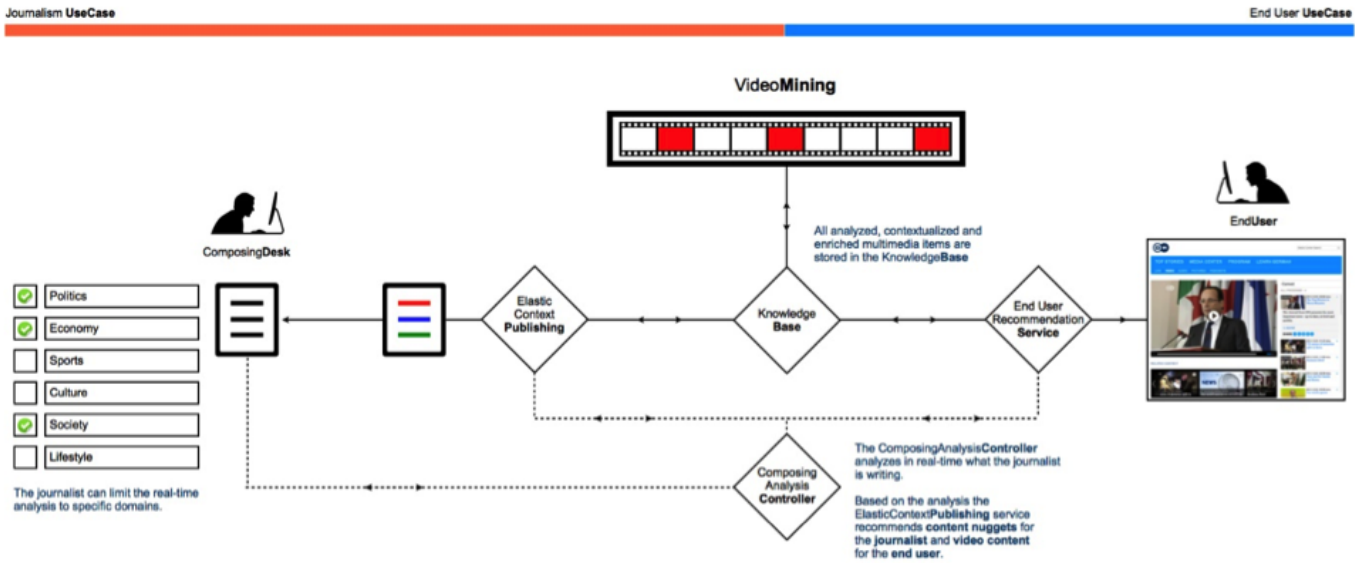

Figure 2. Multimodal platform catering both for the journalist and the end-user's use-cases

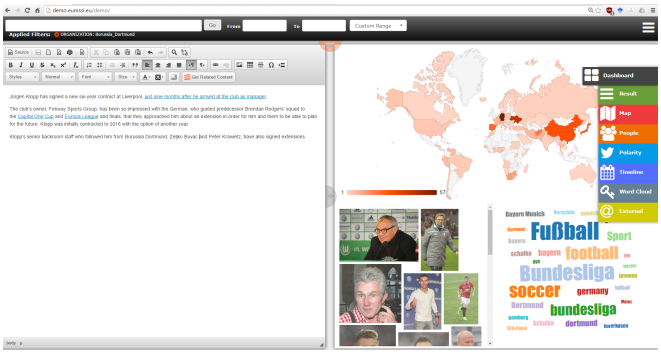

Figure 3. The storytelling web application

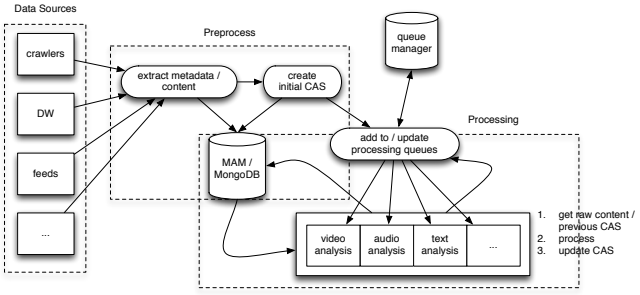

Figure 4. Architecture design

All components run independently and can be seen as basically "stateless" in that they maintain only the information necessary for immediate execution. As such it is possible to restart individual components without affecting the overall system, making it relatively easy to ensure the overall reliability of the platform.

All new content coming into the system is first normalized to a common metadata schema (based on schema.org) and stored in a MongoDB database to make it available for further processing. Analysis results, as well as the original metadata, are stored in UIMA's CAS format[3] to allow integration of different aligned layers of analysis as well as in a simplified format that is then indexed with Solr. The applications use the Solr indexes for efficient and scalable access to the analyzed content, as well as statistical metrics over the whole document collection or specific subsets that can be used for exploration and visualization.

[3] Unstructured Information Management Architecture: http://uima.apache.org/

The process flow, pictured in Figure 4, can be summarized as follows:

1. new data arrives (or gets imported)
2. preprocessing stage
   (a) make content available through unique internal identifier
   (b) create initial CAS with aligned metadata / text content and content URI
   (c) mark initial processing queue states
3. processing / content analysis
   (a) distributed analysis systems query queue when they have processing capacity
   (b) retrieve CAS with existing data (or get relevant metadata from wrapper API)
   (c) retrieve raw content based on content URI
   (d) process
   (e) update CAS (possibly through wrapper API)
   (f) create simplified output for indexing

(g) update queues
   i. mark item as processed by the given queue
   ii. mark availability of data to be used by other analysis processes
4. updating the Solr indexes whenever updated information is available for a content items

Note that this architecture design mainly depicts the data analysis part of the EUMSSI system. The applications for end users are built upon the Solr indexes that are automatically synchronized with the analysis results.

Crawlers, preprocessors and API layer are maintained as part of the core EUMSSI platform. The **MongoDB database** is installed separately and managed from within the platform components (with little or no specific configuration and setup), and the same goes for some external dependencies such as having a Tomcat server on which to run the API layer.

Analysis components for video and audio are fully external and independent and communicate with the platform through the API layer. Text analysis and cross-modality components are implemented as UIMA components and run as pipelines integrated into the platform using custom input (CollectionReader) and output (CASConsumer) modules that read an existing CAS representation of the document from the MongoDB back-end, and write back a modified CAS with added annotations (and possibly layers/views) as well as extracted or "flattened" metadata that can be used by other components (e.g. a list of all detected entities in the document).

Crawlers make external data sources available to the platform. Some crawler components are run only once to import existing datasets, whereas others feed continuously into the platform. **Preprocessing** takes original metadata from the different sources and transforms it into a unified representation with a common metadata vocabulary.

The **EUMSSI API** abstracts away from the underlying storage (MongoDB and CAS data representation) to facilitate access for external components such as video and audio processing. It acts as a light-weight layer that translates between the internal data structure and REST-like operations tailored to the needs of the components.

Indexing takes care of making the metadata (from the original source as well as automatically extracted) available to demonstrators and applications by mirroring the data on a Solr server that is accessible to those applications. It is performed using mongo-connector[4], leveraging built-in replication features of MongoDB for low-latency real time indexing of new (even partial) content, as well as content updates.

Components that are part of the core platform can be found on GitHub and are organized into directories corresponding to the type of component. More detailed information about those components may be found in their respective README.md files.

3.1 Design decisions and related content analysis platforms

Apart from integrating a wide variety of analysis components working on text, audio, video, social media, etc., at different levels of semantic abstraction, a key aspect of EUMSSI is the integration and combination of those different information layers. This is the main motivation for using UIMA as the main underlying framework, as described in section 4. This also has the advantage of providing a platform for building processing pipelines that has low overhead when running on a single machine (all information is passed in-memory), while still enabling distributed and scaled-out processing when necessary.

On the other hand, it quickly became apparent that not all kinds of analysis are a good fit for such a workflow, leading to the hybrid approach described in section 5. Having the workflow control in the same database as the data itself eliminates some of the potential failures of more complex queue management systems by ensuring consistency between the stored data and its analysis status. It also means that efforts in guaranteeing availability and performance can focus on optimizing and allocating resources for the MongoDB database (for which best practices are well established).

While there are commercial content management systems on the market, some of which allow for the integration of some automatic content analysis, none of them have the flexibility of the EUMSSI platform, and in particular none are aimed at facilitating cross-modality integration.

Some recent research projects approach similar goals. MultiSensor[5] combines analysis services through distributed RESTful services based on NIF as an interchange format, incurring higher communication overheads in exchange for greater independence of services (compared to the UIMA-based parts of EUMSSI). LinkedTV[6] has a similar approach to EUMSSI (also using MongoDB and Solr), integrating the outputs of different analysis processes in a common MPEG-7 representation in the *consolidation* step, however (it appears) with far less mutual integration of outputs from different modalities. MediaMixer[7] focuses on indexing Media Fragments[8] with metadata to improve retrieval in media production, and BRIDGET[9] provides means to link (bridge) from broadcast content to related items, partly based on automatic video analysis.

4 Aligned data representation

Much of the reasoning and cross-modal integration depends on an aligned view of the different annotation layers, e.g., in order to connect person names detected from OCR with corresponding speakers from the speaker recognition component, or faces detected by the face recognition.

The Apache UIMA[10] CAS (common analysis structure) 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, Keyword, 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)

[4] https://github.com/mongodb-labs/mongo-connector

[5] http://multisensorproject.eu/
[6] http://linkedtv.eu
[7] http://mediamixer.eu
[8] https://www.w3.org/2008/WebVideo/Fragments/
[9] http://ict-bridget.eu
[10] http://uima.apache.org/

- While each CAS represents one "document" or "content item", it can have several *Views* that represent different aspects of that item, e.g. the video layer, audio layer, metadata layer, transcribed text layer, etc., with separate source content (SofA or "subject of annotation") and separate sets of annotations
- CASes can be passed efficiently in-memory between UIMA analysis engines
- CASes can be serialized in a standardised OASIS format[11] for storage and interchange

Annotations based directly on multimedia content (video and audio) naturally refer to that content via timestamps, whereas text analysis modules normally work with character offsets relative to the text content. It is therefore fundamental that any textual views created from multimedia content (e.g. via ASR or OCR) refer back to the timestamps in the original content. This is done by creating annotations, e.g. tokens or segments, that include the original timestamps as attributes in addition to the character offsets.

As an example, we may have a CAS with an audio view which contains the results of automatic speech recognition (ASR), providing the transcription as a series of tokens/words with a timestamp for each word as an additional feature.

In this way it is possible to apply standard text analysis modules (that rely on character offsets) on the textual representation, while maintaining the possibility to later map the resulting annotations back onto the temporal scale.

So called *SofA-aware* UIMA components are able to work on multiple views, whereas "normal" analysis engines only see one specific view that is presented to them. This means that e.g. standard text analysis engines don't need to be aware that they are being applied to an ASR view or an OCR view; they just see a regular text document. SofA-aware components, however, can explicitly work on annotations from different views and can therefore be used to integrate and combine the information coming from different sources or layers, and create new, integrated views with the output from that integration and reasoning process.

5 Synchronous and asynchronous workflow management

In EUMSSI we decided to use a dual approach to workflow management, allowing for synchronous (and even on-demand) analysis pipelines as well as the execution of large batch jobs which need to be run asynchronously, possibly scheduled according to the availability of computational resources.

We opted for UIMA as the basis for synchronous workflows, as well as the data representation used for integrating different analysis layers. On the other hand, a web-based API allows other analysis processes, such as audio and video analysis, to retrieve content and upload results independently, giving them complete freedom to schedule their work according to their specific needs.

5.1 Analysis pipelines using UIMA

UIMA provides a platform for the execution of analysis components (*Analysis Engines* or *AEs*), as well as for managing the flow between those components. CPE or uimaFIT[12] [2] can be used to design and execute pipelines made up of a sequence of AEs (and potentially some more complex flows), and UIMA-AS[13] (*Asynchronous Scale-out*) permits the distribution of the process among various machines or even a cluster (with the help of UIMA DUCC[14]).

Within the EUMSSI project we have developed and integrated a number of UIMA analysis components, mostly dealing with text analysis and semantic enrichment. Whenever possible, components from the UIMA-based DKPro project [1] were used, especially for the core analysis components (tokenization, part-of-speech, parsing, etc.). In addition to a large number of ready-to-use components, DKPro Core provides a unified type system to ensure interoperability between components from different sources. Other components developed or integrated in EUMSSI were made compatible with this type system.

5.2 Managing content analysis with MongoDB

There are some components of the EUMSSI platform, however, that do not integrate easily in this fashion. This is the case of computationally expensive processes that are optimized for batch execution. A UIMA AE needs to expose a *process()* method that operates on a single CAS (= document), and is therefore not compatible with batch processing. This is particularly true for processes that need to be run on a cluster, with significant startup overhead, such as many video and audio analysis tasks.

It is therefore necessary to have an alternative flow mechanism for offline or batch processes, which needs to integrate with the processing performed within the UIMA environment.

The main architectural and integration issues revolve around the data flow, rather than the computation. In fact, the computationally complex and expensive aspects are specific to the individual analysis components, and should not have an important impact on the design of the overall platform.

As such, the design of the flow management is presented in terms of transformations between data states, rather than from the procedural point of view. The resulting system should only rely on the robustness of those data states to ensure the reliability and robustness of the overall system, protecting against potential problems from server failures or other causes. At any point, the system should be able to resume its function purely from the state of the persisted data.

To ensure reliability and performance of the data persistence, we use the well-established and widely used database system MongoDB, which provides great flexibility as well as proven scalability and robustness.

Figure 5 shows the general flow of the EUMSSI system, focusing on the data states needed for the system to function.

In order to avoid synchronization issues, the state of the data processing is stored together with the data within each content item, and the list of pending tasks can be extracted at any point through simple database queries. We therefore only depend on the MongoDB database (which can be replicated across several machines or even a large cluster for performance and reliability) to fully establish the processing state of all items. For example, the queues for analysis processes can be constructed directly from the "processing.queues.*queue_name*" field of an item by selecting (for a given queue) all items that have not yet been processed by that queue and that fulfill all prerequisites (dependencies).

The analysis results are stored in CAS format (optionally with compression). In order to avoid potential conflicts or race conditions

[11] http://docs.oasis-open.org/uima/v1.0/uima-v1.0.html
[12] https://uima.apache.org/uimafit.html
[13] http://uima.apache.org/doc-uimaas-what.html
[14] http://uima.apache.org/doc-uimaducc-whatitam.html

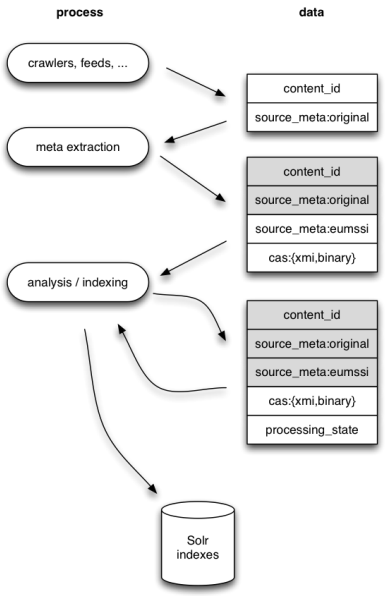

Figure 5. data flow and transformations

between components (most analysis processes run independently of one another), the different layers are stored in separate database fields as independent CASes. Components that work across layers then merge the separate CASes into a single one (as separate Views) in order to combine the information. The "meta.extracted" section of a document is used to store the simplified analysis results that are automatically synchronized with the Solr index, and can also be used as inputs to other annotators (such as detected Named Entities as input to speech recognition), to avoid the overhead of extracting that information from the CAS on demand.

In its simplest form, the processes responsible for the data transitions are fully independent and poll the database periodically to retrieve pending work. Those processes can then be implemented in any language that can communicate comfortably with MongoDB.

5.3 Multimodal multilayer data integration and enrichment

The integration of data from different analysis layers is usually done by loading the CAS representations generated by different prior processes and merging them as individual Views in a single CAS. Layers that work on different representations, e.g. speaker recognition, audio transcript and OCR, are aligned by using timestamps associated with the segments or tokens. As a result, new integrated views can be created, combining the different information layers. Metadata is also enriched by adding information to existing annotations or creating new ones, e.g. with information obtained from SPARQL DBpedia lookups.

5.4 Indexing for scalable data-driven applications

The final applications do not use the information stored in MongoDB directly, but rather access Solr indexes created from that information to respond specifically to the types of queries needed by the applications. Those indexes are updated whenever new analysis results are available for a given item, through the use of *mongo-connector* which keeps the indexes always up-to-date with the content of the "meta.source" and "meta.extracted" sections.

6 Standards and interoperability

EUMSSI uses established protocols and uses freely available and widely used open source software as its underpinnings, in addition to publishing in-project developments under permissive open source licenses through popular platforms such as GitHub.

The API for external analysis components is REST-like and uses JSON for communication, whereas the end applications access the data through Solr's REST-like API (which supports various result formats). Metadata is represented using a vocabulary built upon *schema.org* and the internal representations in UIMA use the DKPro type system as a core. Entity linking is performed against the DBpedia, thus yielding Linked Open Data URIs for entities and allowing for the use of SPARQL and RDF to access additional information.

There is now a starting initiative to establish a "standard" type system for UIMA, with initial conversations pointing towards building upon the DKPro type system for this purpose. Various institutions have expressed their interest in endorsing such a type system, leading to a major step forward in improving interoperability between UIMA components from different sources.

7 Conclusions and future work

In the EUMSSI project we have developed a platform capable of handling large amounts of multimedia, with support for online and offline processing as well as the alignment and combination of different information layers. The system includes many interactions between different modalities, such as doing text analysis on speech recognition output, or adding Named Entities from surrounding text to the vocabulary known to the ASR system, among others.

The platform has proven capable of handling millions of content items on modest hardware, and is designed to allow for easily adding capacity through horizontal scaling.

The source code of the platform and many of the analysis components is publicly available at https://github.com/EUMSSI/. Additional documentation can be found in the corresponding wiki at https://github.com/EUMSSI/EUMSSI-platform/wiki.

ACKNOWLEDGEMENTS

The work presented in this article is being carried out within the FP7-ICT-2013-10 STREP project EUMSSI under grant agreement n° 611057, receiving funding from the European Union's Seventh Framework Programme managed by the REA-Research Executive Agency http://ec.europa.eu/research/rea.

REFERENCES

[1] Richard Eckart de Castilho and Iryna Gurevych, '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*, pp. 1–11, Dublin, Ireland, (August 2014). Association for Computational Linguistics and Dublin City University.

[2] Philip V. Ogren and Steven J. Bethard, 'Building test suites for UIMA components', *SETQA-NLP '09 Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 1–4, (June 2009).

Query-based Topic Detection Using Concepts and Named Entities

Ilias Gialampoukidis[1], Dimitris Liparas[1], Stefanos Vrochidis[1], and Ioannis Kompatsiaris[1]

Abstract. In this paper, we present a framework for topic detection in news articles. The framework receives as input the results retrieved from a query-based search and clusters them by topic. To this end, the recently introduced "DBSCAN-Martingale" method for automatically estimating the number of topics and the well-established Latent Dirichlet Allocation topic modelling approach for the assignment of news articles into topics of interest, are utilized. Furthermore, the proposed query-based topic detection framework works on high-level textual features (such as concepts and named entities) that are extracted from news articles. Our topic detection approach is tackled as a text clustering task, without knowing the number of clusters and compares favorably to several text clustering approaches, in a public dataset of retrieved results, with respect to four representative queries.

1 INTRODUCTION

The need by both journalists and media monitoring companies to master large amounts of news articles produced on a daily basis, in order to identify and detect interesting topics and events, has highlighted the importance of the topic detection task. In general, topic detection aims at grouping together stories-documents that discuss about the same topic-event. Formally, a topic is defined in [1] as "*a specific thing that happens at a specific time and place along with all necessary preconditions and unavoidable consequences*". It is clarified [1] that the notion of "topic" is not general like "accidents" but is limited to a specific collection of related events of the type accident, such as "cable car crash". We shall refer to topics as news clusters, or simply clusters.

The two main challenges involved in the topic detection problem are the following: one needs to (1) estimate the correct number of topics/news clusters and (2) assign the most similar news articles into clusters. In addition, the following assumptions must be made: Firstly, real data is highly noisy and the number of clusters is not known a priori. Secondly, there is a lower bound for the minimum number of documents per news cluster.

In this context, we present and describe the hybrid clustering framework for topic detection, which has been developed within the FP7 MULTISENSOR project[2]. For a given query-based search, the main idea is to efficiently cluster the retrieved results, without the need for a pre-specified number of topics. To this end, the framework, recently introduced in [2], combines automatic estimation of the number of clusters and assignment of news articles into topics of interest, on the results of a text query. The estimation of the number of clusters is done by the novel "DBSCAN-Martingale" method [2], which can deal with the aforementioned assumptions. All clusters are progressively extracted (by a density-based algorithm) by applying Doob's martingale and then Latent Dirichlet Allocation is applied for the assignment of news articles to topics. Contrary to [2], the contribution of this paper is based on the fact that the overall framework relies on high-level textual features (concepts and named entities) that are extracted from the retrieved results of a textual query, and can assist any search engine.

The rest of the paper is organized as follows: Section 2 provides related work with respect to topic detection, news clustering and density-based clustering. In Section 3, our framework for topic detection is presented and described. Section 4 discusses the experimental results from the application of our framework and several other clustering methods to four collections of text documents, related to four given queries, respectively. Finally, some concluding remarks are provided in Section 5.

2 RELATED WORK

Topic detection is traditionally considered as a clustering problem [3], due to the absence of training sets. The clustering task usually involves feature selection [4], spectral clustering [5] and k-means oriented [3] techniques, assuming mainly that the number of topics to be discovered is known a priori and there is no noise, i.e. news items that do not belong to any of the news clusters. Latent Dirichlet Allocation (LDA) is a popular approach for topic modelling for a given number of topics k [6]. LDA has been generalized to nonparametric Bayesian approaches, such as the hierarchical Dirichlet process [7] and DP-means [8], which predict the number of topics k. The extraction of the correct number of topics is equivalent to the estimation of the correct number of clusters in a dataset. The majority vote among 30 clustering indices has been proposed in [9] as an indicator for the number of clusters in a dataset. In contrast, we propose an alternative majority vote among 10 realizations of the "DBSCAN-Martingale", which is a modification of the DBSCAN algorithm [10] with parameters the density level ε and a lower bound for the minimum number of points per cluster. However, the DBSCAN-Martingale [2] regards the density level ε as a random variable and the clusters are progressively extracted. We consider the general case, where the number of topics to be discovered is unknown and it is possible to have news articles which are not assigned to any topic.

Graph-based methods for event detection and multimodal clustering in social media streams have appeared in [11], where a graph clustering algorithm is applied on the graph of items. The decision, whether to link two items or not, is based on the output of a classifier, which assigns or not, the candidate items in the same

[1] Information Technologies Institute, CERTH, Thessaloniki, Greece, email: {heliasgj, dliparas, stefanos, ikom}@iti.gr

[2] **http://www.multisensorproject.eu/**

cluster. Contrary to this graph-based approach, we cluster news items in an unsupervised way.

Density-based clustering does not require as input the number of topics. OPTICS [12] is very useful for the visualization of the cluster structure and for the optimal selection of the density level ε. The OPTICS-ξ algorithm [12] requires an extra parameter ξ, which has to be manually set in order to find "dents" in the OPTICS reachability plot. The automatic extraction of clusters from the OPTICS reachability plot, as an extension of the OPTICS-ξ algorithm, has been presented in [13] and has been outperformed by HDBSCAN [14] in several datasets of any nature. In the context of news clustering, however, we shall examine whether some of these density-based algorithms perform well on the topic detection problem and by comparing them with our DBSCAN-Martingale, in terms of the number of estimated topics. All the aforementioned methods, which do not require the number of topics to be known a priori, are combined with LDA in order to examine whether the use of DBSCAN-Martingale (combined with LDA) provides the most efficient assignment of news articles to topics.

3 TOPIC DETECTION USING CONCEPTS AND NAMED ENTITIES

The MULTISENSOR framework for topic detection, which is presented in Figure 1, is approached as a news clustering problem, where the number of topics needs to be estimated. The overall framework is based on textual features, namely concepts and named entities. The number of topics k is estimated by DBSCAN-Martingale and the assignment of news articles to topics is done using Latent Dirichlet Allocation (LDA).

LDA has shown great performance in text clustering, given the number of topics. However, in realistic applications, the number of topics is unknown to the system. On the other hand, DBSCAN does not require as input the number of clusters, but its performance in text clustering is very weak, due to the fact that it assigns too much noise to the news article collection and this results in very limited performance [2]. Moreover, it is difficult to find a unique density level that can output all clusters. Thus, we keep only the number of clusters using density-based clustering and the assignment of documents to topics is done by the well-performing LDA.

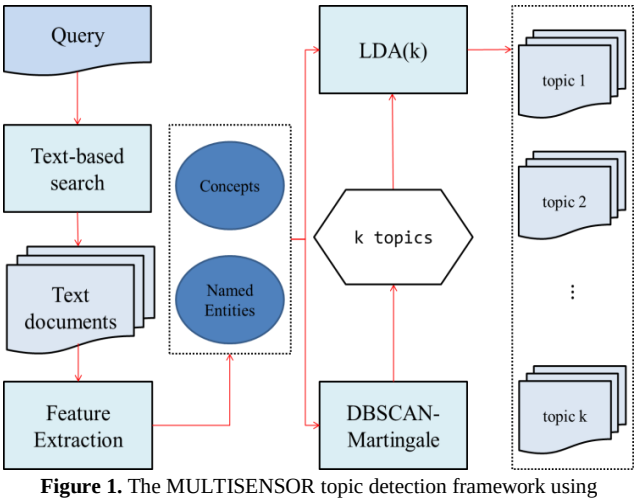

Figure 1. The MULTISENSOR topic detection framework using DBSCAN-Martingale and LDA

In our approach, the constructed DBSCAN-Martingale combines several density levels and is applied on high-level concepts and named entities. In the following, the construction of DBSCAN-Martingale is briefly reported.

3.1 The DBSCAN-Martingale

Given a collection of n news articles, density-based clustering algorithms output clustering vector C with values the cluster IDs $C[j]$ for each news item $j = 1,2,\dots,n$, where we denote by $C[j]$ the j-th element of a vector C. In case the j-th document is not assigned to any of the clusters, the j-th cluster ID is zero. Assuming that $C_{DBSCAN(\varepsilon)}$ is the clustering vector provided by the DBSCAN [10] algorithm for the density level ε, the problem is to combine the results for several values of ε, into one unique clustering result. To that end, a martingale construction has been presented in [2], where the density level ε is a random variable, uniformly sampled in a pre-defined interval.

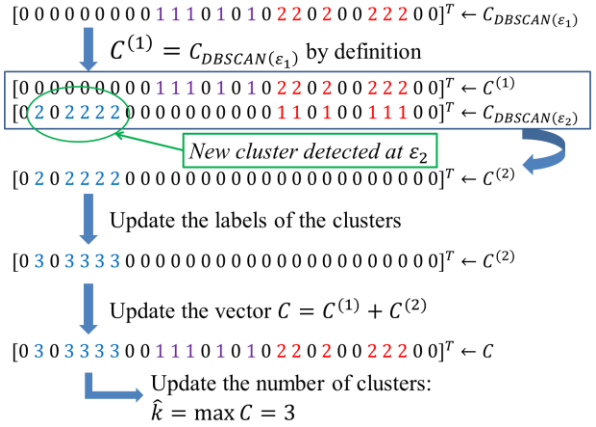

Figure 2. One realization of the DBSCAN-Martingale with T = 2 iterations and 3 topics detected [2]

The DBSCAN-Martingale progressively updates the estimation of the number of clusters (topics), as shown in Figure 2, where 3 topics are detected in 2 iterations of the process. Due to the randomness in the selection of the density levels ε, it is likely that each realization of the DBSCAN-Martingale will output a random variable $\hat{k}$ as an estimation of the number of clusters. Hence, we allow 10 realizations $\widehat{k_1}, \widehat{k_2}, \dots, \widehat{k_{10}}$ and the final estimation of the number of clusters is the majority vote over them. An illustrative example of 5 clusters in the 2-dimensional plane is demonstrated in Figure 3.

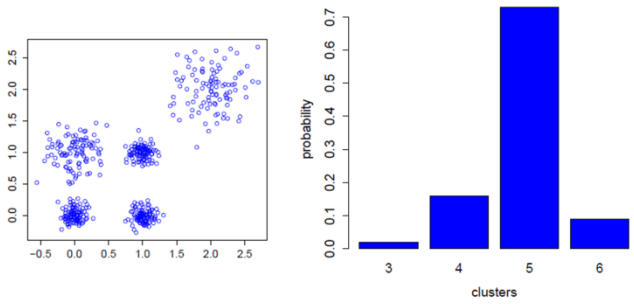

Figure 3. Example in the 2-dimensional plane and the histogram of results after 100 realizations of the DBSCAN-Martingale

In brief, the DBSCAN-Martingale is mathematically formulated as follows. Firstly, a sample of size T $\varepsilon_t,\ t = 1,2,\dots,T$ is randomly generated in $[0, \varepsilon_{max}]$, where ε_{max} is an upper bound for the density levels. The sample of $\varepsilon_t,\ t = 1,2,\dots,T$ is then sorted in increasing order. For each density level ε_t we find the corresponding clustering vectors $C_{DBSCAN(\varepsilon_t)}$ for all stages $t = 1,2,\dots,T$. In the first stage, all clusters detected by $C_{DBSCAN(\varepsilon_1)}$ are kept, corresponding to the lowest density level ε_1. In the second stage ($t = 2$), some of the detected clusters by $C_{DBSCAN(\varepsilon_2)}$ are new and some of them have also been detected by $C_{DBSCAN(\varepsilon_1)}$. In order to keep only the newly detected clusters, we keep only groups of numbers of the same cluster ID with size greater than $minPt$s. Finally, the cluster IDs are relabelled and the maximum value of a clustering vector provides the number of clusters.

Complexity: The DBSCAN-Martingale requires T iterations of the DBSCAN algorithm, which runs in $\mathcal{O}(n \log n)$ if a tree-based spatial index can be used and in $\mathcal{O}(n^2)$ without tree-based spatial indexing [12]. Therefore, the DBSCAN-Martingale runs in $\mathcal{O}(Tn \log n)$ for tree-based indexed datasets and in $\mathcal{O}(Tn^2)$ without tree-based indexing. Our code[3] is written in R[4], using the dbscan[5] package, which runs DBSCAN in $O(n\ log\ n)$ with kd-tree data structures for fast nearest neighbor search.

3.2 Latent Dirichlet Allocation (LDA)

LDA assumes a Bag-of-Words (BoW) representation of the collection of documents and each topic is a distribution over terms in a fixed vocabulary. LDA assigns probabilities to words and assumes that documents exhibit multiple topics, in order to assign a probability distribution on the set of documents. Finally, LDA assumes that the order of words does not matter and, therefore, LDA is not applicable to word n-grams for $n \geq 2$, but can be applied to named entities and concepts. This input allows topic detection even in multilingual corpora, where n-grams are not available in a common language.

4 EXPERIMENTS

In this Section, we describe our dataset and evaluate our method.

4.1 Dataset description

A part of the present MULTISENSOR database (in which articles crawled from international news websites are stored) was used for the evaluation of our query-based topic detection framework. We use the retrieved results for a given query in order to cluster them into labelled clusters (topics) without knowing the number of clusters. The concepts and named entities are extracted using the DBpedia spotlight[6] online tool and the final concepts and named entities replaced the raw text of each news article. The final collection of text documents is available online[7].

The queries that were used for the experiments are the following:

- energy crisis
- energy policy
- home appliances
- solar energy

It should be noted that the aforementioned queries are considered representative, with respect to the use cases addressed by the MULTISENSOR project. The output of our topic detection framework can be visualized in Figure 4 for the query "home appliances", where the retrieved results are clustered by 9 topics. The font size of the clusters' labels depends on the particular word probability within each cluster.

4.2 Evaluation results

In order to evaluate the clustering of the retrieved news articles, we use the average precision (AP), broadly used in the context of information retrieval, clustering and classification. A document d of a cluster C is considered relevant to C (true positive), if at least one concept associated with document d appears also in the label of cluster C. It should be noted that the labels of the clusters (topics) are provided by the concepts or named entities that have the highest probability (provided by LDA) within each topic. Precision is considered the fraction of relevant documents in a cluster and average precision is the average for all clusters of a query. Finally, we average the AP scores for all considered queries to obtain the Mean Average Precision (MAP).

We compared the clustering performance of the proposed topic detection framework, in which the DBSCAN-Martingale algorithm (for estimating the number of topics) and LDA (for assigning news articles to topics) are employed, against a variety of well-known clustering approaches, which were also combined with LDA for a fair comparison. DP-means is a Dirichlet process and we used its implementation in R[8]. HDBSCAN is a hierarchical DBSCAN approach, which uses the "excess-of-mass" (EOM) approach to find the optimal cut. Nbclust is a majority vote of the first 16 indices, which are all described in detail in [9].

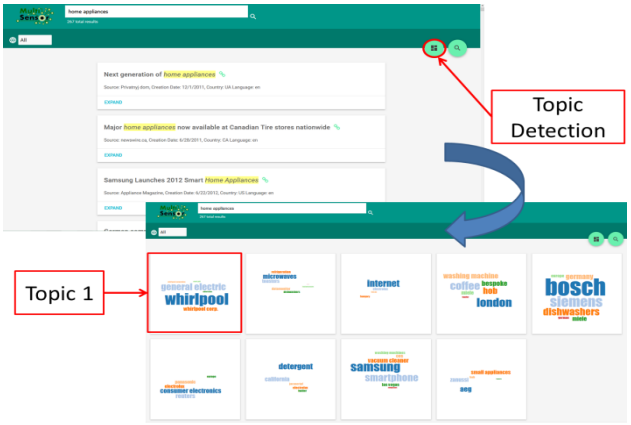

Figure 4. Demonstration of the MULTISENSOR topic detection framework

[3] https://github.com/MKLab-ITI/topic-detection
[4] https://www.r-project.org/
[5] https://cran.r-project.org/web/packages/dbscan/index.html
[6] https://dbpedia-spotlight.github.io/demo/
[7] http://mklab2.iti.gr/project/query-based-topic-detection-dataset
[8] https://github.com/johnmyleswhite/bayesian_nonparametrics

Table 1. Average Precision (± standard deviation) and Mean Average Precision over 10 runs of LDA using the estimated number of topics

Index + LDA	energy crisis	energy policy	home appliances	solar energy	MAP
CH	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
Duda	0.4498±0.0671	**0.5534±0.0457**	0.4299±0.0237	0.4484±0.0067	0.4703
Pseudo t^2	0.4498±0.0671	**0.5534±0.0457**	0.4299±0.0237	0.4484±0.0067	0.4703
C-index	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
Ptbiserial	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
DB	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
Frey	0.3541±0.0181	0.3911±0.0033	0.3745±0.064	0.4484±0.0067	0.3920
Hartigan	0.5938±0.0502	0.5336±0.0375	0.5942±0.0282	0.5961±0.0347	0.5794
Ratkowsky	0.5357±0.0151	0.5371±0.0357	0.4962±0.0721	0.5375±0.0446	0.5266
Ball	0.4207±0.0093	0.4501±0.0021	0.4975±0.016	0.4464±0.0614	0.4536
McClain	0.5786±0.0425	0.5371±0.0357	0.3745±0.064	0.5961±0.0347	0.5215
KL	0.5786±0.0425	0.5371±0.0357	0.5701±0.0145	0.5961±0.0347	0.5704
Silhouette	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
Dunn	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
SDindex	0.3541±0.0181	0.3911±0.0033	0.5942±0.0282	0.4484±0.0067	0.4469
SDbw	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
NbClust	0.5786±0.0425	0.5371±0.0357	0.5942±0.0282	0.5961±0.0347	0.5765
DP-means	0.3541±0.0181	0.3911±0.0033	0.3745±0.064	0.4484±0.0067	0.3920
HDBSCAN-EOM	0.4498±0.0671	0.3911±0.0033	0.5951±0.0184	0.5375±0.0446	0.4933
DBSCAN-Martingale	**0.7691±0.0328**	**0.5534±0.0457**	**0.6115±0.0225**	**0.6073±0.0303**	**0.6353**

Table 2. Estimation of the number of topics in the MULTISENSOR queries

Index	energy crisis	energy policy	home appliances	solar energy
CH	12	8	15	15
Duda	**4**	**4**	**3**	**2**
Pseudo t^2	4	4	3	2
C-index	12	8	15	15
Ptbiserial	12	8	15	15
DB	12	8	15	15
Frey	2	2	2	2
Hartigan	11	7	15	15
Ratkowsky	7	8	5	5
Ball	3	3	3	3
McClain	12	8	2	15
KL	12	8	11	15
Silhouette	12	8	15	15
Dunn	12	8	15	15
SDindex	2	2	15	2
SDbw	12	8	15	15
NbClust	12	8	15	15
DP-means	2	2	2	2
HDBSCAN-EOM	4	2	10	5
DBSCAN-Martingale	6	4	9	10

The AP scores per query and the MAP scores per method over 10 runs of LDA are displayed in Table 1, for each estimation of the number of topics combined with LDA. In addition, the numbers of news clusters estimated by the considered clustering indices for each query are presented in Table 2. Looking at Table 1, we observe a relative increase of 9.65% in MAP, when our topic detection framework is compared to the second highest MAP score (by Hartigan+LDA) and a relative increase of 10.20%, when compared to the most recent approach (NbClust+LDA).

In general, the proposed topic detection framework outperforms all the considered clustering approaches both in terms of AP (within each query) and in terms of MAP (overall performance for all queries), with the exception of the "energy policy" query, where the performance of our framework is matched by that of the Duda and Pseudo t^2 clustering indices.

Finally, we evaluated the time performance of the DBSCAN-Martingale method and we selected several baseline approaches in order to compare their processing time with that of our approach. In Figure 5, the number of news clusters is estimated for T = 5 iterations for the DBSCAN-Martingale and for maximum number of clusters set to 15 for the indices Duda, Pseudo t^2, Silhouette, Dunn and SDindex. We observe that DBSCAN-Martingale is faster than all other methods. Even when it is applied to 500 documents, it is able to reach a decision about the number of clusters in approximately 0.4 seconds.

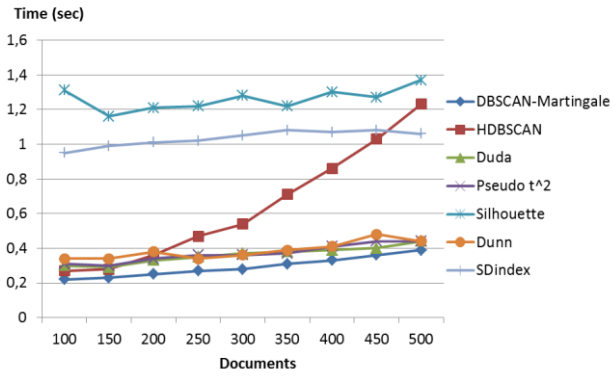

Figure 5. Time performance of DBSCAN-Martingale and several baseline approaches to estimate the number of news clusters

5 CONCLUSIONS

In this paper, we have presented a hybrid topic detection framework, developed for the purposes of the MULTISENSOR project. Given a query-based search, the framework clusters the retrieved results by topic, without the need to know the number of topics a priori. The framework employs the recently introduced DBSCAN-Martingale method for efficiently estimating the number of news clusters, coupled with Latent Dirichlet Allocation for assigning the news articles to topics. Our topic detection framework relies on high-level textual features that are extracted from the news articles, namely textual concepts and named entities. In addition, it is multimodal, since it fuses more than one sources of information from the same multimedia object. The query-based topic detection experiments have shown that our framework outperforms several well-known clustering methods, both in terms of Average Precision and Mean Average Precision. A direct comparison, by means of time performance, has shown that our approach is faster than several well-performing methods in the estimation of the number of clusters, given as input the same number of query-based retrieved news articles.

As future work, we plan to investigate the behavior of our framework by introducing additional modalities/features, examine the application of alternative (other than LDA) text clustering approaches, as well as investigate the extraction of language-agnostic concepts and named entities, something that could provide multilingual capabilities to our topic detection framework.

ACKNOWLEDGEMENTS

This work was supported by the projects MULTISENSOR (FP7-610411) and KRISTINA (H2020-645012), funded by the European Commission.

REFERENCES

[1] J. Allan (Ed.), 'Topic detection and tracking: event-based information organization', vol. 12, Springer Science & Business Media, (2012).

[2] I. Gialampoukidis, S. Vrochidis and I. Kompatsiaris, 'A hybrid framework for news clustering based on the DBSCAN-Martingale and LDA', In: Perner, P. (Ed.) Machine Learning and Data Mining in Pattern Recognition, LNAI 9729, pp. 170-184, (2016).

[3] C. C. Aggarwal and C. Zhai, 'A survey of text clustering algorithms', In Mining Text Data, pp. 77-128, Springer US, (2012).

[4] M. Qian and C. Zhai, 'Unsupervised feature selection for multi-view clustering on text-image web news data', In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1963-1966, ACM, (2014).

[5] A. Kumar and H. Daumé, 'A co-training approach for multi-view spectral clustering', In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 393-400, (2011).

[6] D. M. Blei, A. Y. Ng and M. I. Jordan, 'Latent dirichlet allocation', the Journal of machine Learning research, vol. 3, pp. 993-1022, (2003).

[7] Y. W. Teh, M. I. Jordan, M. J. Beal and D. M. Blei, 'Hierarchical dirichlet processes', Journal of the american statistical association, 101(476), (2006).

[8] B. Kulis and M. I. Jordan, 'Revisiting k-means: New algorithms via Bayesian nonparametrics', arXiv preprint arXiv:1111.0352, (2012).

[9] M. Charrad, N. Ghazzali, V. Boiteau and A. Niknafs, 'NbClust: an R package for determining the relevant number of clusters in a data set', Journal of Statistical Software, 61(6), pp. 1-36, (2014).

[10] M. Ester, H. P. Kriegel, J. Sander and X. Xu, 'A density-based algorithm for discovering clusters in large spatial databases with noise', In Kdd, 96(34), pp. 226-231, (1996).

[11] G. Petkos, M. Schinas, S. Papadopoulos and Y. Kompatsiaris, 'Graph-based multimodal clustering for social multimedia', Multimedia Tools and Applications, 1-23, (2016).

[12] M. Ankerst, M. M. Breunig, H. P. Kriegel and J. Sander, 'OPTICS: ordering points to identify the clustering structure', In ACM Sigmod Record, 28(2), pp. 49-60, ACM, (1999).

[13] J. Sander, X. Qin, Z. Lu, N. Niu and A. Kovarsky, 'Automatic extraction of clusters from hierarchical clustering representations', In Advances in knowledge discovery and data mining, pp. 75-87, Springer Berlin Heidelberg, (2003).

[14] R. J. Campello, D. Moulavi and J. Sander, 'Density-based clustering based on hierarchical density estimates', In Advances in Knowledge Discovery and Data Mining, pp. 160-172, Springer Berlin Heidelberg, (2013).

An active learning method for speaker identity annotation in audio recordings

Broux Pierre-Alexandre[1,2] and **Doukhan David**[1] and **Petitrenaud Simon**[2] and **Meignier Sylvain**[1] and **Carrive Jean**[2]

Abstract. Given that manual annotation of speech is an expensive and long process, we attempt in this paper to assist an annotator to perform a speaker diarization. This assistance takes place in an annotation background for a large amount of archives. We propose a method which decreases the intervention number of a human. This method corrects a diarization by taking into account the human interventions. The experiment is done using French broadcast TV shows drawn from ANR-REPERE evaluation campaign. Our method is mainly evaluated in terms of KSR (Keystroke Saving Rate), and we reduce the number of actions needed to correct a speaker diarization output by 6.8% in absolute value.

1 Introduction

The work presented in this paper has been realized to meet the needs of the French national audiovisual institute[3] (INA). INA is a public institution in charge of the digitalization, preservation, distribution and dissemination of the French audiovisual heritage. Annotations related to speaker identity, together with speech transcription, meet several use-cases. Temporal localization of speaker interventions can be used to enhance the navigation within a media [12, 22]. It may also be used to perform complex queries within media databases [5, 11, 19].

This article focuses on the realization of human-assisted *speaker diarization* systems. Speaker diarization methods consist in estimating "who spoke when" in an audio stream [2]. This media structuring process is an efficient pre-processing step, for instance to help segmenting a broadcast news into anchors and reports before manual documentation processes. Speaker diarization algorithms are generally based on unsupervised machine learning methods [21], in charge of estimating the number of speakers, and splitting the audio stream into labelled speech segments assigned to hypothesized speakers. Speaker identity and temporal localization is known to be a pertinent information for the access and exploitation of speech recordings [5, 20]. However, the accuracy of automatic state-of-the-art speaker recognition methods is still inadequate to be embedded into INA's archiving or media enhancement applications, and a human intervention is required to obtain an optimal description of a speech archive.

Manual annotation of speech is a very expensive process. Nine hours are required to perform the manual annotation corresponding to one hour of spontaneous speech (speech transcription and speaker identity). Previous studies have shown that the speech annotation process may be sped-up using the output of automatic speech recognition systems (ASR) together with speech turn annotations [3]. The resulting annotation task consists in correcting the output of automatic systems, instead of doing the whole annotation manually.

The model proposed in this paper is an active-learning extension of this paradigm, applied to the *speaker diarization* task. Annotator corrections are used in real-time to update the estimations of the speaker diarization system. The aim of this update strategy is to lower the amount of manual corrections to be done, which impact the time spent in the interaction with the system. The quality of the annotations obtained through this process should be maximal, with respect to human abilities on speaker recognition tasks [13].

The paper is organized as follows: Section 2 presents the Human-assisted speaker diarization system. Section 3 presents the corpus, the metrics, whereas section 4 analyzes the results. Section 5 concludes with a discussion of possible directions for future works.

2 Human-assisted speaker diarization system

The proposed speaker diarization prototype is aimed at interacting in real-time with a human user, in charge of correcting the predictions of the system. This system is aimed at producing high quality diarization annotations with a minimal human cost. Such system could be used to speed-up the annotation process of any speech corpus requiring temporal speaker information.

2.1 System overview

In the following description, we assume that an easy-to-use interface is provided to the user, and that the speech segments are presented together with the speech transcription. We also assume that the feedback of the user is limited to three actions:

1. The validation, when the speech segment has a correct speaker label;
2. The speaker label modification, when the speech segment has an incorrect speaker label;
3. The speaker label creation: for speakers encountered for the first time in the recording.

Actions such as speech segment split, or speech segment boundaries modifications are not taken into account in the scope of this paper.

Annotated speech segments corresponding to the whole recording are presented to the annotator. The segment presentation order follows the temporal occurrence of the segments. This choice has been made in order to ease the manual speaker recognition task, with the assumption that the media chronology provides the annotator with a

[1] Computer science laboratory of the university of Maine (LIUM - EA 4023), Le Mans, France
[2] French National audiovisual institute (Ina), Paris, France
[3] `http://www.ina.fr`

better understanding of the speech material. The annotator has to correct, or validate the predictions of the diarization system. Our working paradigm is that a correction requires more time for the annotator than a validation.

Figure 1 describes the proposed active-learning system. The system consists in associating each annotator correction to a real-time re-estimation of the labels of the remaining speech segments to be presented. This method is aimed at improving the quality of the next diarization predictions, resulting in a lower amount of corrections to be done by the annotator, thus lowering the time required for the manual correction. The system is composed of three main steps, which will be detailed in the next sections. The two last steps are repeated until all the segments are checked. Let us give a brief description of these stages:

Initialization: an initial diarization is performed with a fully-automatic speaker diarization system. This step can be time consuming and is performed offline.

User input: the annotator checks each segment, and validates or corrects the speaker label before inspecting the next segment.

Real-time reassignment: the annotator modifications are associated to a re-evaluation of the speaker labels corresponding to the next speech segments to be presented. The computations realized during this step should be fast enough to allow real-time interaction with a human user.

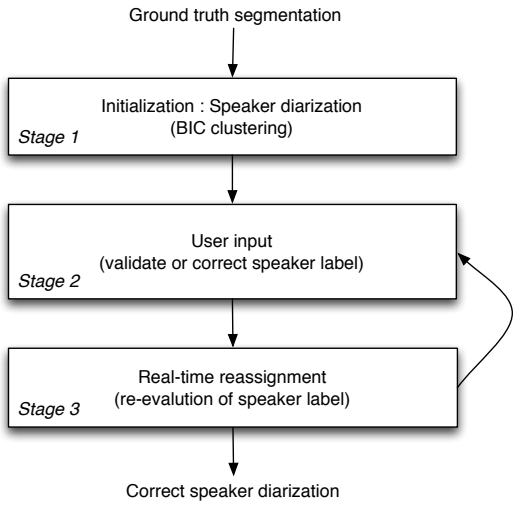

Figure 1: Active-learning system

2.2 Initialization: speaker diarization

The speaker diarization system is inspired by the system described in [2]. It was developed for the transcription and diarization tasks, with the goal of minimizing both word error rate and speaker error rate. It rests upon a segmentation and a hierarchical agglomerative clustering. Furthermore, this system uses MFCC features as audio descriptors [2, 7, 17].

The system is composed of a segmentation step followed with a clustering step. Speaker diarization needs to produce homogeneous speech segments. Errors such as having two distinct clusters (i.e., detected speakers) corresponding to the same real speaker could be easily corrected by merging both clusters. In this article, we focus the study on the clustering step and the segmentation step is based on a perfect manual segmentation (ground truth).

The clustering algorithm is based upon a hierarchical agglomerative clustering. The initial set of clusters is composed of one segment per cluster. Each cluster is modeled by a Gaussian with a full covariance matrix. The ΔBIC measure (cf equation 1) is employed to select the candidate clusters to group as well as to stop the merging process. The two closest clusters i and j are merged at each iteration until $\Delta BIC(i,j) > 0$.

Let $|\Sigma_i|$, $|\Sigma_j|$ and $|\Sigma|$ be the determinants of gaussians associated to the clusters i, j and $i+j$ and λ be a parameter to set up. The penalty factor P (eq. 2) depends on d, the dimension of the features, as well as on n_i and n_j, referring to the total length of cluster i and cluster j respectively. The $\Delta BIC(i,j)$ measure between the clusters i and j is then defined as follows:

$$\Delta BIC(i,j) = \frac{n_i+n_j}{2}\log|\Sigma| - \frac{n_i}{2}\log|\Sigma_i| - \frac{n_j}{2}\log|\Sigma_j| - \lambda P, \quad (1)$$

$$\text{with } P = \frac{1}{2}\left(d + \frac{d(d+1)}{2}\right) + log(n_i+n_j). \quad (2)$$

This speaker diarization system is the first stage of most state-of-the-art systems for TV or radio recording as the one based on GMM or i-vectors[1, 8]. GMM and i-vectors are both statistical models which represent audio data. The generated clusters have a high purity (i.e. each cluster contains mostly only one speaker) and the system is fast.

2.3 User input and Real-time reassignment

User input consists in validating, or correcting, the speaker labels estimated by the diarization system. The proposed active-learning strategy consists in associating each correction, defined as a mismatch between the speakers C_i and C_j, to the computation of new speaker models, trained on the validated speech segments. The resulting models are based on a single gaussian, which is fast to compute, and assumed to be more accurate than the models inferred during the initialization. These simple speaker models are then used to re-estimate the ΔBIC distance with the remaining speech segments involved to the last mismatch (segments attributed to C_i and C_j only).

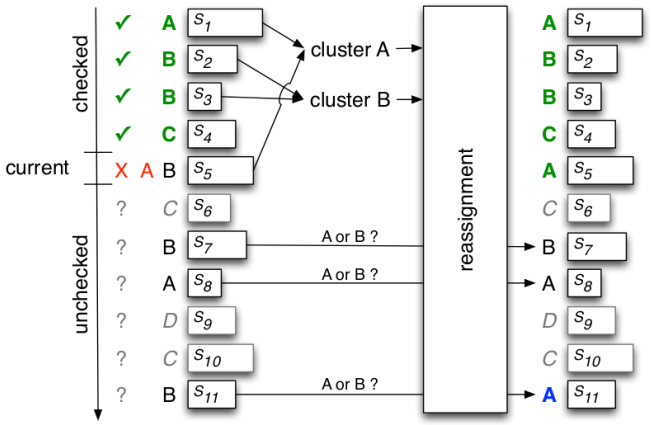

Figure 2: Example of user-input and reassignment

An illustration of these interactions is provided in figure 2. In this example, four speakers (A, B, C, D) have been inferred through the automatic initialization step. The user has manually validated the four first speech segments ($S_1 ... S_4$) before reporting a speaker label modification for segment S_5, tagged as speaker B instead of speaker A. The resulting action of the active-learning system, consists to create speaker models for the mismatching speakers only (A and B). These models are used to re-estimate the labels of the remaining segments tagged with A or B (segments S_7, S_8 and S_{11}), and may lead to a speaker label modification (segment S_{11}). Remaining speech segments tagged with other labels (C and D) are not re-estimated. The modified diarization is updated before the annotator moves to the next segment S_6. The process iterates until the last segments are reached.

3 Evaluation

3.1 Corpus

Experiments were performed on TV recordings drawn from the corpora of ANR-REPERE challenge[4] . The ANR-REPERE is a challenge organized by the LNE (French national laboratory of metrology and testing) and ELDA (Evaluations and Language resources Distribution Agency) in 2010-2014. This challenge is a project in the area of the multimedia recognition of people in television documents. The aim is to find the identities of people who speak along with the quoted and written names at each instant in a television show. The data comes from two French channels (BFM and LCP). Shows were recorded from two French digital terrestrial television channels.

The ANR-REPERE project has started since 2010 and evaluations are set up in 2013 and 2014. In this paper, we merge the 2013 evaluation corpus and the 2014 evaluation corpus to build the corpus called REPERE in the below sections. The table 1 give us some statistics about this corpus. The duration reported in table 1 shows that only a part of the data is annotated and evaluated.

Statistics	REPERE
Show number	15
Recording number	90
Recording time	34h30
Annotation time	13h11
Speaker number	571

Table 1: 2013-2014 news and debate TV recordings from REPERE corpus.

The current diarization systems are less efficient with spontaneous speech mainly present in debates than with prepared speech from news [6]. We have chosen this corpus because of the variety of the shows. The corpus is balanced between prepared and spontaneous speech and composed of street interviews, debates and news shows.

It is common to accept a ± 250 millisecond tolerance on segment boundaries for the recordings with prepared speech and far less for the recordings with spontaneous speech. Having and using a reference segmentation for the segmentation step, we do not normally have segmentation errors. Therefore, we do not use any tolerances on segment boundaries.

Most of the diarization systems are not able to detected overlap speech zones [4, 16, 24]. In the following described experiments, we remove overlap speech from the evaluation and consider it as a non-speech area. Figure 3 shows the segment duration after the superposed speech deletion.

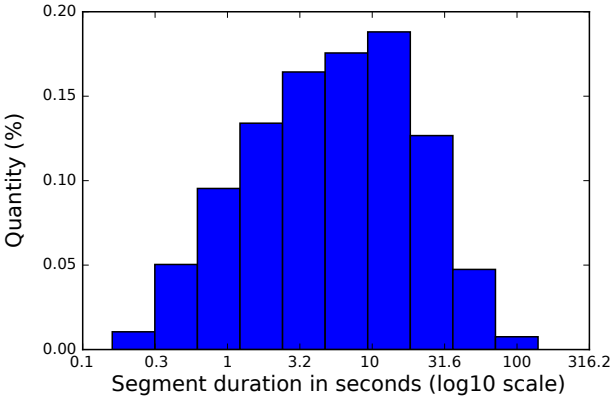

Figure 3: Segment duration of REPERE corpus

3.2 Metrics

3.2.1 Diarization

The metric used to measure performance in the speaker diarization task is the Diarization Error Rate (DER) [18]. DER was introduced by NIST as the fraction of speaking time which is not attributed to the correct speaker, using the best matching between references and hypothesis speaker labels. The scoring tool is available in the $sidekit/s4d$ toolkit[14].

In order to evaluate the impact of a reassignment, we use the percentage of pure clusters with respect to the total number of clusters. We also use the well-known purity as defined in [9] which is the ratio between the number of frames by the dominating speaker in a cluster and the total number of frames in this cluster. This measure is used in order to evaluate the purity of hypothesis clusters according to the assignment provided by reference clusters. To evaluate the action applied by a human, we simply use some counters. These counters will be in the form of percentages in this paper.

3.2.2 Keystroke Saving Rate

The DER and the purity measure the quality of a diarization. The evaluation of the user input is difficult, as the proposed metric needs to be as much as possible reproducible and objective [10]. In our case, the human interactions are simulated.

The proposed method is inspired from a previous work on computer assisted transcription [15]. In this paper the authors proposed to evaluate the human interactions with the Keystroke Saving Rate (KSR) [23].

The KSR method has been developed for AAC (Augmentative and Alternative Communication) systems, so that handicapped persons can use it. It is computed according to the number of keyboard strokes made by the user to write a message. In our case, the strokes corresponds to the number of actions made by the annotator to correct the diarization. To compute the KSR, we assume that the annotator will always choose the best strategy to minimize the number of actions.

[4] `http://www.defi-repere.fr/`

We suppose here that the annotator can make two kinds of actions for a current segment: the reassignment to another cluster (reassignment) or the assignment to a new cluster (creation). The annotator can create a new cluster when the first segment of a given speaker is checked. The number of creations in the whole document, denoted by n_c, is constant for any reassignment even if the threshold λ in equation 1 differs. Similarly the total number of segments reassigned by the user is denoted by n_r and the number of segments is n_s. We define the KSR as the ratio of the sum of the numbers of created clusters and the reassigned segments n_r given the number of segments in the initial diarization (equation 3):

$$KSR = \frac{n_c + n_r}{n_s} \times 100. \quad (3)$$

A KSR equal to 0% corresponds to a perfect speaker diarization in which each segment is assigned to the true corresponding speaker. In this case, the annotator does not reassign any segments. Conversely, a KSR equal to 100% corresponds to the worse speaker diarization in which each segment is assigned to the wrong speaker. Therefore, the annotator needs to change the assignment of all the segments, if the corrections are not gradually propagated in the rest of the document.

4 Results

4.1 Speaker diarization

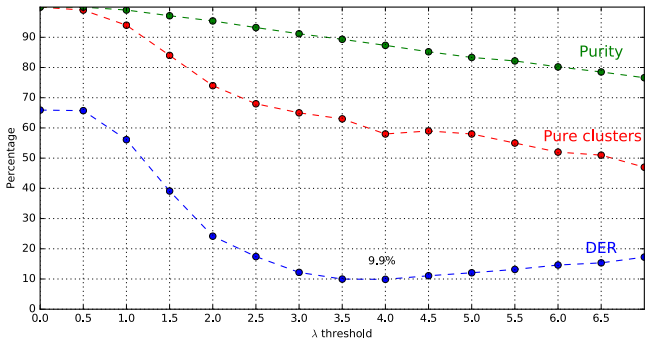

Figure 4: Initial diarization: DER, % of pure clusters and average cluster purity

The speaker diarization is based on hierarchical clustering where each speaker is modeled by a gaussian with a full covariance computed over acoustic features. The acoustic features are composed of 12 MFCCs with energy, and are not normalized (the background channel helps to segment and cluster the speaker) [2, 7, 17].

As mentioned previously, we use the ground truth segmentation as input of the clustering algorithm (corresponding to the stage 1 in figure 1) and the overlapping speaker segments are removed in the ground truth.

Figure 4 shows the DER of the speaker diarization for different λ thresholds (cf. equation 1). The lower DER is 9.9% for a λ threshold of 4.0. Compared to literature [8], this DER is rather low, which is mainly due to ground truth segmentation: the segments contain the voice of a single speaker, overlap segments are removed, as well as there are no missed speech and no false alarm speech segments.

4.2 Active-learning system

In our experiments, the human annotator is simulated with the ground truth speaker annotations. The main objective is to decrease the number of actions performed by an annotator to obtain a perfect diarization. To reach this goal, we compare the KSR obtained with or without the human corrections taken into consideration (i.e. with or without an active-learning reassignment) using various λ thresholds for the speaker diarization.

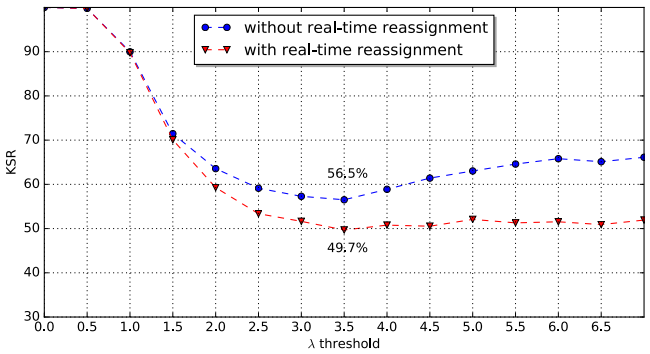

Figure 5: KSR

The real-time segment reassignment stage (stage 3 in figure 1) uses the same parameters as the initial diarization: 12MFCC+energy, full covariance gaussian and BIC metric to label the unchecked segments
.

Figure 5 gives the KSR of the system with real-time reassignment (including stages 2 & 3) and the system without real-time reassignment (including stage 2 only). The KSR decreases until $\lambda = 3.5$ in both systems and increases when λ is upper. The KSR is 56.5% and 49.7% respectively without reassignment and with reassignment when λ is equal to 3.5. About half segments are manually corrected (49.7%) and the 6.8% in absolute value are reassigned to the correct speaker automatically after a user correction.

In the most favorable case when λ is at 3.5, the DER is low, about 10% and the average cluster purity is equal to 90%. In the same time, only 60% of the clusters are 100% pure (cf. figure 4). The difference between these indicators can be explained by the fact that, unlike the DER, the KSR does not take into account the duration of the segments . Most of the errors come from the small segments, and these ones are numerous (cf. figure 3).

The KSR remains almost static when λ is greater than 3.5 in the system with reassignment, whereas the choice of the parameter λ is more critical to minimize the number of actions in the system without reassignment. Finally, one can notice that the system with reassignment always obtains a lower KSR whatever the λ value, except for $\lambda = 0$ where the KSR is equal to 100% in both cases.

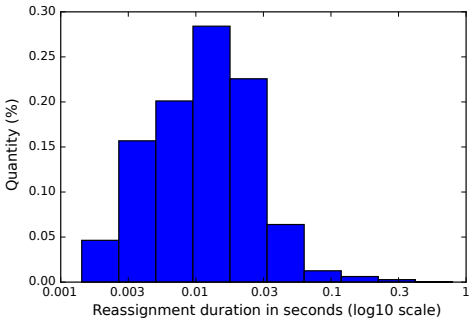

Figure 6: Time of reassignment after each user correction.

After each user correction, the unchecked segments are clustered again in the reassignment stage. The process is generally fast, since the duration takes less than 0.03 second in 95% of cases, so it is interesting to notice that this stage could be done in real time without any impact on the user interface (figure 6).

5 Conclusion & prospects

In this paper, we attempt to find a way to help a human to segment and cluster the speakers in an audio or audio-visual document. We propose a method that takes into consideration the annotator corrections by modifying the allocation of the unchecked segments. The proposed computer assisted method allows us to obtain a noticeable reduction in the number of required corrections. Not only is our method effective, but the corrections are also made quickly. Thanks to its fast treatment, this could be applied in a real application without impacting the reactivity of the interface and without increasing the work intensity of the annotator.

Some future improvements should be done on the base of this preliminary work. Firstly, we plan to minimize the number of user actions by applying a constrained clustering to reassign all unchecked segments and to create or delete clusters. Another improvement would be to integrate the automatic segmentation in the correction process.

6 Acknowledgments

This research was partially supported by the European Commission, as part of the Event Understanding through Multimodal Social Stream Interpretation (EUMSSI) project (contract number FP7-ICT-2013-10) in which the LIUM is involved.

References

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, 'Speaker diarization: A review of recent research', **20**(2), 356–370, (Feb 2012).

[2] C. Barras, X. Zhu, S. Meignier, and J.L. Gauvain, 'Multi-stage speaker diarization of broadcast news', *IEEE Transactions on Audio, Speech and Language Processing*, **14**(5), 1505–1512, (2006).

[3] Thierry Bazillon, Yannick Estève, and Daniel Luzzati, 'Transcription manuelle vs assistée de la parole préparé et spontanée', *Revue TAL*, (2008).

[4] Kofi Boakye, Beatriz Trueba-Hornero, Oriol Vinyals, and Gerald Friedland, 'Overlapped speech detection for improved speaker diarization in multiparty meetings', in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4353–4356. IEEE, (2008).

[5] Mbarek Charhad, Daniel Moraru, Stéphane Ayache, and Georges Quénot, 'Speaker identity indexing in audio-visual documents', in *Content-Based Multimedia Indexing (CBMI2005)*, (2005).

[6] Ruchard Dufour, Vincent Jousse, Yannick Estève, Fréderic Béchet, and Georges Linarès, 'Spontaneous speech characterization and detection in large audio database', *SPECOM, St. Petersburg*, (2009).

[7] Grégor Dupuy, *Les collections volumineuses de documents audiovisuels: segmentation et regroupement en locuteurs*, Ph.D. dissertation, Université du Maine, 2015.

[8] Grégor Dupuy, Sylvain Meignier, Paul Deléglise, and Yannick Esteve, 'Recent improvements on ilp-based clustering for broadcast news speaker diarization', in *Proc. Odyssey Workshop*, (2014).

[9] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda, 'Partitioning and transcription of broadcast news data.', in *ICSLP*, volume 98, pp. 1335–1338, (1998).

[10] Edouard Geoffrois, 'Evaluating interactive system adaptation', in *The International Conference on Language Resources and Evaluation*, (2016).

[11] Jerry Goldman, Steve Renals, Steven Bird, Franciska De Jong, Marcello Federico, Carl Fleischhauer, Mark Kornbluh, Lori Lamel, Douglas W Oard, Claire Stewart, et al., 'Accessing the spoken word', *International Journal on Digital Libraries*, **5**(4), 287–298, (2005).

[12] Nicolas Hervé, Pierre Letessier, Mathieu Derval, and Hakim Nabi, 'Amalia.js: An open-source metadata driven html5 multimedia player', in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, MM '15, pp. 709–712, New York, NY, USA, (2015). ACM.

[13] Juliette Kahn, *Parole de locuteur: performance et confiance en identification biométrique vocale*, Ph.D. dissertation, Avignon, 2011.

[14] Anthony Larcher, Kong Aik Lee, and Sylvain Meignier, 'An extensible speaker identification sidekit in python', in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5095–5099. IEEE, (2016).

[15] Antoine Laurent, Sylvain Meignier, Teva Merlin, and Paul Deléglise, 'Computer-assisted transcription of speech based on confusion network reordering', in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4884–4887. IEEE, (2011).

[16] Xavier Anguera Miro, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals, 'Speaker diarization: A review of recent research', *Audio, Speech, and Language Processing, IEEE Transactions on*, **20**(2), 356–370, (2012).

[17] Lindasalwa Muda, Mumtaj Begam, and I Elamvazuthi, 'Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques', *arXiv preprint arXiv:1003.4083*, (2010).

[18] NIST. The rich transcription spring 2003 (RT-03S) evaluation plan. `http://www.itl.nist.gov/iad/mig/tests/rt/2003-spring/docs/rt03-spring-eval-plan-v4.pdf`, February 2003.

[19] Roeland Ordelman, Franciska De Jong, and Martha Larson, 'Enhanced multimedia content access and exploitation using semantic speech retrieval', in *Semantic Computing, 2009. ICSC'09. IEEE International Conference on*, pp. 521–528. IEEE, (2009).

[20] Julien Pinquier and Régine André-Obrecht, 'Audio indexing: primary components retrieval', *Multimedia tools and applications*, **30**(3), 313–330, (2006).

[21] Sue E Tranter and Douglas A Reynolds, 'An overview of automatic speaker diarization systems', *IEEE Transactions on Audio, Speech, and Language Processing*, **14**(5), 1557–1565, (2006).

[22] Félicien Vallet, Jim Uro, Jérémy Andriamakaoly, Hakim Nabi, Mathieu Derval, and Jean Carrive, 'Speech trax: A bottom to the top approach for speaker tracking and indexing in an archiving context', in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), (2016).

[23] Matthew EJ Wood and Eric Lewis, 'Windmill-the use of a parsing algorithm to produce predictions for disabled persons', *PROCEEDINGS-INSTITUTE OF ACOUSTICS*, **18**, 315–322, (1996).

[24] Martin Zelenák and Javier Hernando, 'The detection of overlapping speech with prosodic features for speaker diarization.', in *INTERSPEECH*, pp. 1041–1044, (2011).

Recent improvements on error detection for automatic speech recognition

Yannick Estève and **Sahar Ghannay** and **Nathalie Camelin**[1]

Abstract.

Automatic speech recognition(ASR) offers the ability to access the semantic content present in spoken language within audio and video documents. While acoustic models based on deep neural networks have recently significantly improved the performances of ASR systems, automatic transcriptions still contain errors. Errors perturb the exploitation of these ASR outputs by introducing noise to the text. To reduce this noise, it is possible to apply an ASR error detection in order to remove recognized words labelled as errors.

This paper presents an approach that reaches very good results, better than previous state-of-the-art approaches. This work is based on a neural approach, and more especially on a study targeted to acoustic and linguistic word embeddings, that are representations of words in a continuous space.

In comparison to the previous state-of-the-art approach which were based on Conditional Random Fields, our approach reduces the classification error rate by 7.2%.

1 Introduction

The advancement in the speech processing field and the availability of powerful computing devices have led to better performance in the speech recognition domain. However, recognition errors are still unavoidable, whatever the quality of the ASR systems. This reflects their sensitivity to the variability: the acoustic environment, speaker, language styles and the theme of the speech. These errors can have a considerable impact on the application of certain automatic processes such as information retrieval, speech to speech translation, etc.

The encountered errors can be due to a misinterpretation of the signal. For example, the noise associated with the sound of the environment or a problem with the quality of recording channel is interpreted as speech by the system. One of the source of errors may also come from a mispronunciation of a word, a non respect speech turn when two speakers are involved at the same time also creates a disturbance of the sound signal.

The efficient generation of speech transcriptions in any condition (*e.g.* noise free environment, etc.) remains the ultimate goal, which is not already solved. Error detection can help to improve the exploitation of ASR outputs by downstream applications, but is a difficult task given the fact that there are several types of errors, which can range from the simple substitution of a word with a homophone to the insertion of an irrelevant word for the overall understanding of the sequence of words. They can also affect neighboring words and create a whole area of erroneous words.

[1] LIUM - University of Le Mans, France, email: name.surname@univ-lemans.fr

Error detection can be performed in three steps: first, generating a set of features that are based on ASR system or gathered from other source of knowledge. Then, based on these features, estimating correctness probabilities (confidence measures). Finally, a decision is made by applying a threshold on these probabilities.

Many studies focus on the ASR error detection. In [14], authors have applied the detection capability for filtering data for unsupervised learning of an acoustic model. Their approach was based on applying two thresholds on the linear combination of two confidence measures. The first one, was derived from language model and takes into account backoff behavior during the ASR decoding. This measure is different from the language model score, because it provides information about the word context. The second is the posterior probability extracted from the confusion network. In [5], authors addressed the issue of error region detection and characterization in Large Vocabulary Continuous Speech Recognition (LVCSR) transcriptions. They proposed to classify error regions in four classes, in particular, they are interested in a person noun error which is a critical information in many information retrieval applications. They proposed several sequential detection, classification approaches and an integrated sequence labeling approach. The ASR error detection problem is related to the Out Of Vocabulary (OOV) detection task, considering that OOV errors behavior and impact differ from other errors, assuming that OOV words contribute to recognition errors on surrounding words. Many studies focused on detecting OOV errors. More recently, in [17], authors have also focused on detecting error regions generated by OOV words. They proposed an approach based on CRF tagger, which takes into account contextual information from neighboring regions instead of considering only the local region of OOV words. This approach leads to significant improvement compared to state of the art. The generalization of this approach for other ASR errors was presented in [1], which proposes an error detection system based on CRF tagger using various ASR, lexical and syntactic features. Their experiments that are performed on two corpora in English for the DARPA BOLT project showed the validity of this approach for the detection of important errors. In [18], new features gathered from other knowledge sources than the decoder itself were explored for ASR error detection, which are a binary feature that compares the outputs from two different ASR systems (word by word), a feature based on the number of hits of the hypothesized bigrams, obtained by queries entered into a very popular Web search engine, and finally a feature related to automatically infered topics at sentence and word levels. Two out of three new features, a binary word match feature and a bigram hit feature, led to significant improvements, with a maximum entropy model and CRF with linear-chain conditional random fields, comparing to a baseline using only decoder-based features. A neural network classifier

trained to locate errors in an utterance using a variety of features is presented in [20]. Two approaches are proposed to extract confidence measures : the first one, is based on Recurrent Neural network Language Model (RNNLM) features to capture long-distance context within and across previous utterances. The second one, consist of combining complementary state-of-the-art DNN and GMM ASR for effective error detection, by leveraging DNN and GMM confusion networks that store word confusion information from multiple systems for feature extraction.

The ASR error detection method presented in this paper is based on incorporating a set of features in the confidence classifier built on neural network architectures, including MLP and DNN, which is in charge to attribute a label (error or correct) for each word of an ASR hypothesis.

A combination approach based on the use of an auto encoder is applied to combine well-known word embeddings: this combination helps to take benefit from the complementarities of these different word embeddings, as recently shown in one of our previous studies [10].

2 ASR error detection based on word embeddings

The error detection system has to attribute the label *correct* or *error* to each word in the ASR transcript. Each decision is based on a set of heterogeneous features. In our approach, this classification is performed by analyzing each recognized word within its context.

The proposed ASR error detection system is based on a feed forward neural network and is designed to be fed by different kinds of features, including word embeddings.

2.1 Architecture

This ASR error detection system is based on a multi-stream strategy to train the network, named multilayer perceptron multi stream (MLP-MS). The MLP-MS architecture is used in order to better integrate the contextual information from neighboring words. This architecture is inspired by [7] where word and semantic features are integrated for topic identification in telephone conversations. The training of the MLP-MS is based on pre-training the hidden layers separately and then fine tuning the whole network. The proposed architecture, depicted in Figure 1, is detailed as follows: three feature vectors are used as input to the network – feature vectors are described in the next section. These vectors are respectively the feature vector representing the two left words (L), the feature vector representing the current word (W) and the feature vector for the two right words (R). Each feature vector is used separately in order to train a multilayer perceptron (MLP) with a single hidden layer. Formally, the architecture is described by the following equations:

$$H_{1,X} = f(P_{1,X} \times X + b_{1,X}) \quad (1)$$

where X represents respectively the three feature vectors (L,W and R), P_i is the weight matrix and b_i is the bias vector.
The resulting vectors $H_{1,L}$, $H_{1,W}$ and $H_{1,R}$ are concatenated to form the first hidden layer $H1$. The H_1 vector is presented as the input of the second *MLP-MS* hidden layer H_2 computed according to the equation:

$$H_2 = g(P_2 \times H_1 + b_2) \quad (2)$$

Finally, the output layer is a vector O_k of k=2 nodes corresponding to the 2 labels $correct$ and $error$:

$$O_k = q(P_O \times H_2 + b_O) \quad (3)$$

Note that in our experiments f and g are respectively rectified linear units ($ReLU$) and hyperbolic tangent ($tanh$) activation functions, and q is the $softmax$ function.

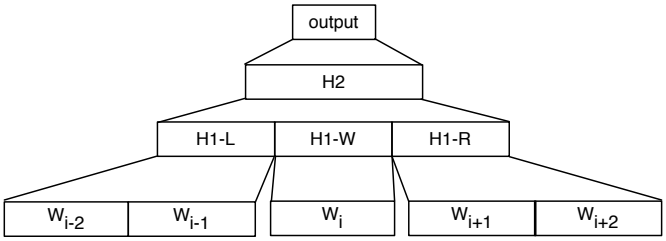

Figure 1. MLP-MS architecture for ASR error detection task.

2.2 Feature vectors

In this section, we describe the features collected for each word and how they are extracted. Some of these features are nearly the same as the ones presented in [1]. The word feature vector is the concatenation of the following features:

- **ASR confidence scores**: confidence scores are the posterior probabilities generated from the ASR system (PAP). The word posterior probability is computed over confusion networks, which is approximated by the sum of the posterior probabilities of all transitions through the word that are in competition with it.
- **Lexical features**: lexical features are derived from the word hypothesis output from the ASR system. They include the word length that represents the number of letters in the word, and three binary features indicating if the three 3-grams containing the current word have been seen in the training corpus of the ASR language model.
- **Syntactic features**: we obtain syntactic features by automatically assigning part-of-speech tags (POS tags), dependency labels – such label is a grammatical relation held between a governor (head) and a dependent –, and word governors, which are extracted from the word hypothesis output by using the MACAON NLP Tool chain[2] [16] to process the ASR outputs.
- **Linguistic word representation (embedding or symbol)**: The orthographic representation of a **word** is used in CRF approaches as for instance in [2]. Using our neural approach we can handle different word embeddings, which permits us to take advantage of the generalizations extracted during the construction of the continuous vectors.
- **Acoustic word embeddings**: these vectors represents the pronunciation of a word as a projection in a space with high dimension. Words projected into a close area are words acoustically similar [3].

[2] http://macaon.lif.univ-mrs.fr

2.3 Linguistic word embeddings: a combination-based approach

Different approaches have been proposed to create word embeddings through neural networks. These approaches can differ in the type of the architecture and the data used to train the model. In this study, we distinguish two categories of word embeddings: the ones estimated on unlabeled data, and others estimated on labeled data (dependency-based word embeddings). These representations are detailed respectively in the next subsections.

2.3.1 Word embeddings based on unlabeled data

This section presents three types of word embeddings coming from two available implementations (word2vec [15] and GloVe [19]):

- **Skip-gram**: This architecture from [15] takes as input the target word w_i and outputs the preceding and the following words.
  The target word W_i is at the input layer, and the context words C are at the output layer. It consists on predicting the contextual words C given the current word w_i.
  The skip-gram model with negative sampling seeks to represent each word W_i and each context C as d-dimensional vectors ($V_{W_{wi}}$,V_C) in order to have similar vector representations for similar words. This is done by maximizing the dot product $V_{W_{wi}}.V_C$ associated with the good word-context pairs that occur in the document D and minimize it for negative examples, that do not necessarily exist in D. These negative examples are created by stochastically corrupting the pairs (W_i, C), thus the name *negative sampling*.
  Also, the context is not limited to the immediate context, and training instances can be created by skipping a constant number of words in its context, for instance, $w_{i_{-3}}$, $w_{i_{-4}}$, $w_{i_{+3}}$, $w_{i_{+4}}$, hence the name *skip-gram*.
- **GloVe**: This approach is introduced by [19], and relies on constructing a global co-occurrence matrix X of words, by processing the corpus using a sliding context window. Here, each element X_{ij} represents the number of times the word j appears in the context of word i.
  The model is based on the global co-occurrence matrix X instead of the actual corpus, thus the name GloVe, for Global Vectors.
  This model seeks to build vectors V_i and V_j that retain some useful information about how every pair of words i and j co-occur, such as:

$$V_i^T V_j + b_i + b_j = log X_{ij} \quad (4)$$

  where b_i and b_j are the bias terms associated with words i and j, respectively.
  This is accomplished by minimizing a cost function J, which evaluates the sum of all squared errors:

$$J = \sum \sum f(X_{ij})(V_i^T V_j + b_i + b_j - log X_{ij})^2 \quad (5)$$

  where f is weighting function which is used to prevent learning only from very common word pairs. The authors define the f as follows [19]:

$$f(X_{ij}) = \begin{cases} \frac{X_{ij}}{X_{max}} & if\ X_{ij} < X_{max} \\ 1 & otherwise \end{cases}$$

2.3.2 Dependency-based word embeddings

Levy *et al.* [13] proposed an extension of word2vec, called word2vecf and denoted **w2vf-deps**, which allows to replace linear bag-of-words contexts with arbitrary features.

This model is a generalization of the skip-gram model with negative sampling introduced by [15], and it requires labeled data for training. As in [13], we derive contexts from dependency trees: a word is used to predict its governor and dependents, jointly with their dependency labels. This effectively allows for variable window size.

2.3.3 Word embedding combination

In the framework of this work, we have experimented different ways to combine the word embeddings presented above. Like described in a previous paper [10], the use of an auto encoder is very effective.

3 Acoustic word embeddings

3.1 Building acoustic word embeddings

The approach we used to build acoustic word embeddings is inspired from the one proposed in [3]. Word embeddings are trained through a deep neural architecture, depicted in figure 2, which relies on a convolutional neural network (CNN) classifier over words and on a deep neural network (DNN) trained by using a triplet ranking loss [3, 21, 22]. This architecture was proposed in [3] with the purpose to use the scores derived from the word classifier for lattice rescoring. The two architectures are trained using different inputs: speech signal and orthographic representation of the word.

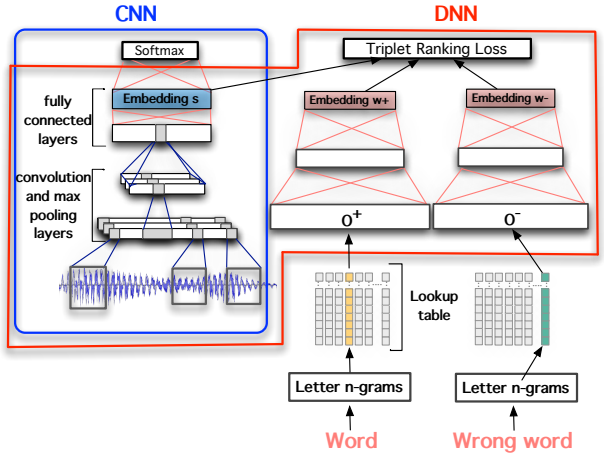

Figure 2. Deep architecture used to train acoustic word embeddings.

The CNN is trained to predict a word given an acoustic sequence of T frames as input. It is composed of a number of convolution and pooling layers, followed by a number of fully connected layers which feeds into the final softmax layer. The final fully connected layer just below the softmax one is called embedding layer **s** (it was called **e** in [3]). It contains a compact representation of the acoustic signal. This representation tends to preserve acoustic similarity between words, such that words are close in this space if they sound alike.

The idea behind using the second architecture is to be able to build an acoustic word embedding from orthographic word representation, especially in order to get an acoustic word embeddings for words not already observed in an audio speech signal. More, a such acoustic

word embedding derived from an orthographic representation can be perceived as a canonical acoustic representation for a word, since different prononciations imply different embeddings **s**.

Like in [3], orthographic word representation consists on a bag of n-grams ($n \leq 3$) of letters, composed of 10222 trigrams, bigrams, and unigrams of letters, including special symbols *I* and *J* to specify the start and the end of a word. Then, we use an auto-encoder to reduce the size of this bag of n-grams vector to d-dimension. To check the performance of the resulting orthographic representation, a neural network is trained to predict a word given this orthographic representation. It reaches 99.99% of accuracy on the training set composed of $52k$ words of the vocabulary, showing the richness of this representation.

Similar to [3], a DNN was trained by using the triplet ranking loss [3, 21, 22] in order to project the orthographic word representation to the acoustic embeddings **s** obtained from the CNN architecture, which is trained independently. It takes as input a word orthographic representation and outputs an embedding vector of the same size as **s**. During the training process, this model takes as inputs the acoustic embedding **s** selected randomly from the training set, the orthographic representation of the matching word $\mathbf{o}^+$, and the orthographic representation of a randomly selected word different to the first word $\mathbf{o}^-$. These two orthographic representations supply shared parameters in the DNN.

We call $t = (\mathbf{s}, \mathbf{w}^+, \mathbf{w}^-)$ a triplet, where **s** is the acoustic signal embedding, $\mathbf{w}^+$ is the embedding obtained through the DNN for the matching word, while $\mathbf{w}^-$ is the embedding obtained for the wrong word. The triplet ranking loss is defined as:

$$Loss = \max(0, m - Sim_{dot}(s, w^+) + Sim_{dot}(s, w^-)) \quad (6)$$

where $Sim_{dot}(x, y)$ is the dot product function used to compute the similarity between two vectors x and y, and m is a margin parameter that regularizes the margin between the two pairs of similarity $Sim_{dot}(\mathbf{s}, \mathbf{w}^+)$ and $Sim_{dot}(\mathbf{s}, \mathbf{w}^-)$. This loss is weighted according to the rank in the CNN output of the word matching the audio signal.

The resulting trained model can then be used to build an acoustic embedding ($\mathbf{w}^+$) from any word, as long as one can extract an orthographic representation from it.

3.2 Experiments

3.2.1 Experimental data

Experimental data for ASR error detection is based on the entire official ETAPE corpus [11], composed by audio recordings of French broadcast news shows, with manual transcriptions (reference). This corpus is enriched with automatic transcriptions generated by the LIUM ASR system, which is a multi-pass system based on the CMU Sphinx decoder, using GMM/HMM acoustic models. This ASR system won the ETAPE evaluation campaign in 2012. A detailed description is presented in [4].

The automatic transcriptions have been aligned with reference transcriptions using the *sclite*[3] tool. From this alignment, each word in the corpora has been labeled as correct (C) or error (E). The description of the experimental data, in terms of size, word error rate (WER) as well as percentage of substitution (Sub), deletion (Del) and insertion (Ins), is reported in Table 1.

The performance of the proposed approach is compared with a state-of-the-art system based on CRFs [2] provided by the *Wapiti* tagger[4] [12] and applied to the set of features presented in Section 2.2. The ASR error detection systems (MLP-MS and CRF) are trained on the training corpus (Train) and are applied on the test (Test) set. The development set (Dev) was used to tune all the parameters: the learning rate, the batch size and the hidden layers size of MLP-MS, and the features template of CRF, that describes which features are used in training and testing.

Name	#words ref	#words hyp	WER	Sub	Del	Ins
Train	349K	316K	25.3	10.3	12.0	3.1
Dev	54K	50K	24.6	10.3	11.0	3.3
Test	58K	53K	21.9	8.3	10.9	2.7

Table 1. Description of the experimental corpus.

The performance is evaluated by using recall (R), precision (P) and F-measure (F) for the misrecognized word prediction and global Classification Error Rate (CER). CER is defined as the ratio of the number of misclassifications over the number of recognized words.

The linguistic word embedding described in Section 2.3 are made of 200 dimensions. They were computed from a large textual corpus, composed of about 2 billions of words. This corpus was built from articles of the French newspaper "Le Monde", the French Gigaword corpus, articles provided by Google News, and manual transcriptions of about 400 hours of French broadcast news.

The training set for the convolutional neural network used to compute acoustic word embedding consists of 488 hours of French Broadcast News with manual transcriptions. This dataset is composed of data coming from the ESTER1 [8], ESTER2 [9] and EPAC [6] corpora.

It contains $52k$ unique words that are seen at least twice each in the corpus. All of them corresponds to a total of 5.75 millions occurrences. In French language, many words have the same pronunciation without sharing the same spelling, and they can have different meanings; *e.g.* the sound [so] corresponds to four homophones: *sot* (fool), *saut* (jump), *sceau* (seal) and *seau* (bucket), and twice more by taking into account their plural forms that have the same pronunciation: *sots*, *sauts*, *sceaux*, and *seaux*. When a CNN is trained to predict a word given an acoustic sequence, these frequent homophones can introduce a bias to evaluate the recognition error. To avoid this, we merged all the homophones existing among the $52k$ unique words of the training corpus. As a result, we obtained a new reduced dictionary containing $45k$ words and classes of homophones.

Acoustic features provided to the CNN are log-filterbanks, computed every 10ms over a 25ms window yielding a 23-dimension vector for each frame. A forced alignment between manual transcriptions and speech signal was performed on the training set in order to detect word boundaries. The statistics computed from this alignment reveal that 99% of words are shorter than 1 second. Hence we decided to represent each word by 100 frames, thus, by a vector of 2300 dimensions. When words are shorter they are padded with zero equally on both ends, while longer words are cut equally on both ends.

3.2.2 Experimental results

Experimental results are summarized in Table 2. In terms of global classification error rate, the proposed neural approach outperforms

[3] http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

[4] http://wapiti.limsi.fr

the CRF, especially by using a combination of embeddings. It yields 5.8% of CER reduction compared to CRF by using only linguistic word embedding. By using also acoustic word embeddings, the CER reduction reached 7.2%. One can also notice that the use of an auto encoder to combine word embeddings is really useful to capture complementarities of different single linguistic word embeddings.

Approach	Word Represent.	Label error			Global
		P	R	F	CER
CRF *(baseline)*	*discrete*	*67.69*	*54.74*	*60.53*	*8.56*
Neural with single ling. word embed.	w2vf-deps	71.90	50.98	59.66	8.26
	Skip-gram	**74.45**	46.75	57.44	8.30
	GloVe	72.16	46.97	56.90	8.53
Neural with ling. word embed. combination	w2vf-deps $\oplus$ Skip-gram $\oplus$ GloVe	69.66	57.89	63.23	8.07
Neural with ling. word embed. combination and acoustic word embed.	w2vf-deps $\oplus$ Skip-gram $\oplus$ GloVe $+ \mathbf{s} + \mathbf{w}^{+}$	70.09	**58.92**	**64.02**	**7.94**

Table 2. Comparison of the use of different types of word embeddings in MLP-MS error detection system on Test corpus.

4 Conclusion

In this paper, we have investigated the use of a neural network to detect ASR error. Specifically, we proposed to effectively represent words through linguistic and acoustic word embeddings.

Experiments were made on automatic transcriptions generated by LIUM ASR system applied on the ETAPE corpus (French broadcast news). They show that the proposed neural architecture, using the acoustic word embeddings as additional features, outperforms state-of-the-art approach based on the use of Conditional Random Fields, with a reduction of the classification error rate of 7.2%.

5 Acknowledgements

This work was partially funded by the European Commission through the EUMSSI project, under the contract number 611057, in the framework of the FP7-ICT-2013-10 call and by the Région Pays de la Loire.

REFERENCES

[1] Frédric Béchet and Benoit Favre, 'Asr error segment localisation for spoken recovery strategy', *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference*, (2013).

[2] Frédric Béchet and Benoit Favre, 'Asr error segment localization for spoken recovery strategy', in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6837–6841, (May 2013).

[3] Samy Bengio and Georg Heigold, 'Word embeddings for speech recognition.', in *INTERSPEECH*, pp. 1053–1057, (2014).

[4] Paul Deléglise, Yannick Estève, Sylvain Meignier, and Teva Merlin, 'Improvements to the LIUM French ASR system based on CMU Sphinx: what helps to significantly reduce the word error rate?', in *Interspeech*, Brighton, UK, (September 2009).

[5] Richard Dufour, Géraldine Damnati, and Delphine Charlet. Automatic error region detection and characterization in lvcsr transcriptions of tv news shows, 2012. Acoustics, Speech and Signal Processing (ICASSP).

[6] Yannick Estève, Thierry Bazillon, Jean-Yves Antoine, Frédéric Béchet, and Jérôme Farinas, 'The EPAC Corpus: Manual and Automatic Annotations of Conversational Speech in French Broadcast News.', in *LREC*. Citeseer, (2010).

[7] Yannick Estève, Mohamed Bouallegue, Carole Lailler, Mohamed Morchid, Richard Dufour, Georges Linarès, Driss Matrouf, and Renato De Mori, 'Integration of word and semantic features for theme identification in telephone conversations', in *6th International Workshop on Spoken Dialog Systems (IWSDS 2015)*, (2015).

[8] Sylvain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre, and Guillaume Gravier, 'The ESTER phase II evaluation campaign for the rich transcription of French Broadcast News.', in *Interspeech*, pp. 1149–1152, (2005).

[9] Sylvain Galliano, Guillaume Gravier, and Laura Chaubard, 'The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts.', in *Interspeech*, volume 9, pp. 2583–2586, (2009).

[10] Sahar Ghannay, Yannick Estève, Nathalie Camelin, Benoit Favre, Camille Dutrey, Fabian Santiago, and Martine Adda-Decker, 'Word embeddings for ASR error detection: combinations and evaluation.', in *LREC*, (2016).

[11] Guillaume Gravier, Gilles Adda, Niklas Paulsson, Matthieu Carr, Aude Giraudel, and Olivier Galibert, 'The ETAPE corpus for the evaluation of speech-based TV content processing in the French language', in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, (2012).

[12] Thomas Lavergne, Olivier Cappé, and François Yvon, 'Practical very large scale CRFs', in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 504–513. Association for Computational Linguistics, (July 2010).

[13] Omer Levy and Yoav Goldberg, 'Dependencybased word embeddings', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pp. 302–308, (2014).

[14] Julie Mauclair, Yannick Estève, Simon Petit-Renaud, and Paul Deléglise, 'Automatic detection of well recognized words in automatic speech transcription', in *Proceedings of the International Conference on Language Resources and Evaluation : LREC*, (2006).

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, 'Efficient estimation of word representations in vector space', (2013).

[16] Alexis Nasr, Frédéric Béchet, Jean-François Rey, Benoît Favre, and Joseph Le Roux, 'Macaon: An nlp tool suite for processing word lattices', in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pp. 86–91. Association for Computational Linguistics, (2011).

[17] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, 'Contextual information improves oov detection in speech', *in North American chapter of thes Association for Computational Linguistics (NAACL)*, (2010).

[18] Thomas Pellegrini and Isabel Trancoso, 'Improving asr error detection with non-decoder based features.', *INTERSPEECH*, 1950–1953, (2010).

[19] Jeffrey Pennington, Richard Socher, and Christopher D Manning, 'Glove: Global vectors for word representation.', in *EMNLP*, volume 14, pp. 1532–1543, (2014).

[20] Yik-Cheung Tam, Yun Lei, Jing Zheng, and Wen Wang, 'Asr error detection using recurrent neural network language model and complementary asr', *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2312–2316, (2014).

[21] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu, 'Learning fine-grained image similarity with deep ranking', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393, (2014).

[22] Jason Weston, Samy Bengio, and Nicolas Usunier, 'Wsabie: Scaling up to large vocabulary image annotation', in *IJCAI*, volume 11, pp. 2764–2770, (2011).

Multilingual Natural Language Generation within Abstractive Summarization

Simon Mille[1], **Miguel Ballesteros**[1], **Alicia Burga**[1], **Gerard Casamayor**[1], **and Leo Wanner**[1,2]

Abstract. With the tremendous amount of textual data available in the Internet, techniques for abstractive text summarization become increasingly appreciated. In this paper, we present work in progress that tackles the problem of multilingual text summarization using semantic representations. Our system is based on abstract linguistic structures obtained from an analysis pipeline of disambiguation, syntactic and semantic parsing tools. The resulting structures are stored in a semantic repository, from which a text planning component produces content plans that go through a multilingual generation pipeline that produces texts in English, Spanish, French, or German. In this paper we focus on the lingusitic components of the summarizer, both analysis and generation.

1 Introduction

With the tremendous amount of multilingual textual data available in the Internet, techniques for intelligent abstractive text summarization in the language of the preference of the user enjoy a steadily increasing demand for different applications, among them journalism and media monitoring. Thus, journalists and media monitors have to review a large number of press articles on a daily basis, a considerable number of which may not be available in their native language. We present work in progress that tackles the problem of multilingual text summarization using semantic representations.

The most popular summarization strategy is still "extraction"-oriented. Text fragments (in general, entire sentences, but in some cases also phrases), are selected from one or more source documents, based on some relevance metric, and the most relevant fragments are put together in a summary (see, e.g., [19] for an overview). Although extractive summarization can be addressed with little linguistic analysis and, in the case of sentence-based selection metrics, the resulting summaries are always grammatically correct, it is known to have some significant shortcomings. For instance, the selection of the content to be included into the summary is rather coarse-grained and surface-(instead of knowledge-)oriented, and the summaries tend to lack internal coherence between the selected text fragments. Furthermore, in general, the summaries are monolingual, i.e., the original text and the summary are in the same language.

Opposed to extractive summarization is "abstractive summarization". Abstractive (i.e., concept-based) summarization analyzes the original textual material using language parsing and/or Information Extraction into intermediate linguistic or conceptual representations. Content selection relevance-driven techniques are then applied to these representations to choose the content elements that are to be communicated in the summary. From the chosen content elements, a summary is generated using deep Natural Language Generation (NLG) techniques. A number of approaches to abstractive summarization have been proposed. Some attempt to adapt extractive techniques to abstractive summarization [23]. Others do not use abstract representations and remain at a superficial level [15, 26], or use partially abstract structures, be it because not all the content of the input text is represented [20, 17], or because some idiosyncratic features are maintained [37]. It is also not always the case that deep generation is used. For instance, Genest and Lapalme [17] and Saggion and Lapalme [35] start from templates, Ganesan et al. [15] from word lattices, and Cheung and Penn [8] and Genest and Lapalme [16] from syntactic structures. Liu et al. [22] do not have a proper generation component at all. Liu et al. [22] and Cheung and Penn [8] apply sentence fusion, rather than content selection.

We developed techniques for abstractive summarization that are capable of generating multilingual summaries in response to a user query on a specific content element using full state-of-the-art (deep) language analysis and language generation mechanisms, combining statistic and rule-based techniques. In this paper, we focus on the general architecture of the summarizer and its generation module.

2 An architecture for abstractive summarization

2.1 Theoretical framework

The theoretical framework that underlies our system is the Meaning-Text Theory [27]. MTT is based on the notion of dependency, which establishes a relation of "governance" between two elements.

The MTT model supports high expressiveness at the three main levels of the linguistic description of written language: semantics, syntax and morphology, while facilitating a coherent transition between them via intermediate levels of deep syntax and deep morphology. In total, the model foresees five strata; at each stratum, a clearly defined type of linguistic phenomena is described in terms of distinct dependency structures.

- **Semantic Structures** (SemSs) are predicate-argument structures in which the relations between predicates and their arguments are numbered in accordance with the order of the arguments.
- **Deep-syntactic structures** (DSyntSs) are dependency trees, with the nodes labeled by meaningful ("deep") lexical units (LUs) and the edges by actant relations *I*, *II*, *III*, ..., *VI* (in accordance with the syntactic valency pattern of the governing LU) or one of the following three non-argumental relations: *ATTR*(ibute), *COORD*(ination), *APPEND*(itive).
- **Surface-Syntactic Structures** (SSyntSs) are dependency trees in which the nodes are labeled by open or closed class lexemes and the edges by grammatical function relations of the type *subject*, *oblique_object*, *adverbial*, *modifier*, etc.

[1] TALN, Pompeu Fabra University, C/ Roc Boronat, 138, 08018 Barcelona, email: simon.mille| miguel.ballesteros|alicia.burga|gerard.casamayor| leo.wanner@upf.edu

[2] ICREA, Passeig Lluís Companys, 23, 08010 Barcelona

- **Deep-Morphological Structures** (DMorphSs) are chains of lexemes in their base form (with inflectional and PoS features being associated to them in terms of attribute-feature pairs) between which a precedence relation is defined and which are grouped in terms of constituents.
- **Surface-Morphological Structures** (SMorphSs) are chains of inflected word forms, i.e., sentences as they appear in the corpus, except that orthographic contractions still did not take place.

The analysis and generation modules in our abstractive pipeline draw upon these strata. In particular, the tasks of language analysis and language generation can be seen as a sequence of mappings between adjacent strata; for analysis, starting from text and arriving at a semantic (or conceptual) representation , and for generation, starting from a semantic representation up to the text surface.

2.2 A pipeline for abstractive summarization

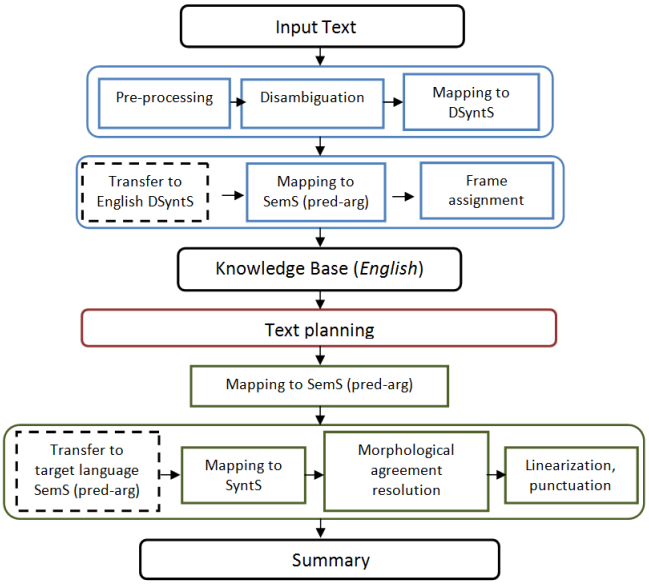

Figure 1. General architecture of the summarizer: Analysis, text planning, and generation

The implementation of our abstractive summarizer is based on a sequence of modules that realize the sequence of transitions between the different strata of the MTT model. The pipeline shown in Figure 1 can be divided into three main parts:

1. **Language analysis**: Language analysis is carried out by a text analysis pipeline that takes as input the textual content of a document in a given language. This content is first analyzed and represented as a forest of DSyntSs. In the case that the input language is different from English, every lexeme in the DSyntSs is mapped onto an English lexeme using bilingual dictionaries in order to arrive at a kind of *interlingua* structure that facilitates language-neutral representations (see Subsection 3.3 for a justification). These English "interlingua" structures are then mapped onto semantic structures, enriched with Frames from the FrameNet lexicon [13], modeled as RDF triples, and stored in a semantic repository.
2. **Text planning**: Conceptual summarization is approached by assessing the relevance of the semantic structures produced by the language analysis step in relation to a specific entity which constitutes a topic of interest for the end user and to which the generated summary is tailored. In addition to determining the relevance of contents, our text planning component also attempts to guarantee a degree of coherence in the summary to be generated by sorting relevant contents in a sequence that satisfies certain choerence constraints, e.g. grouping together in the text contents making reference to the same entities. Relevance calculations are based on relative cooccurrence metrics of word senses and references to entities detected in the original documents during language analysis, the cooccurrence metrics being obtained from pre-existing corpora of annotated documents.
3. **Natural language generation**: Following this planning step, linguistic generation starts by transferring the lexemes associated to the semantic structures to the desired target language, using available multilingual lexical resources. Then, the structure of the sentence is determined and all grammatical words are introduced and linked with syntactic relations. Finally, all morphological agreements between the words are resolved, the words are ordered and punctuation signs are introduced.

3 Language analysis

3.1 Tokenization and disambiguation

Language analysis starts by determining sentence and token boundaries using Bohnet et al.'s [6] tools. Rather than addressing tokenization at word level, however, our analysis pipeline treats each sequence of words referring to a specific entity as an atomic unit of meaning. In doing so, we seek to avoid unnecessary internal analysis of multiword expressions which may not even have a strictly compositional meaning (as, e.g., *United States of America*), and also to eventually obtain predicate-argument structures in which the arguments are not just words, but expressions with an atomic meaning.

To determine the disambiguated senses of individual words and the entities referred to by single words or phrases, we use Babelfy[3] [29]. Babelfy addresses both Word Sense Disambiguation and Entity Linking against BabelNet [30], a large multilingual semantic network organized around *Babel synsets* resulting from mapping WordNet synsets and Wikipedia pages. The large coverage of BabelNet allows Babelfy to annotate both Named Entities and conceptual meanings. All multiword expressions annotated by Babelfy are considered by the following modules as a single token.

3.2 Deep-syntactic parsing

Once the texts are clean, tokenized and the words are disambiguated against BabelNet, they are sent to a parsing module that carries out in sequence Surface-Syntactic and Deep-Syntactic Parsing.

For **Surface-Syntactic Parsing**, we use Bohnet et al.'s [6] joint lemmatizer, part of speech tagger, morphology tagger and dependency parser, which follows a transition-based approach with beam-search. Trained on a surface syntactic treebank, the joint parser produces the surface syntactic tree for an unseen sentence.[4]

For **Deep-Syntactic Parsing**, we use a SSynt-DSynt transducer. The objective of the transducer is to identify and remove all functional words (auxiliaries, determiners, void prepositions and conjunctions) in the surface-syntactic tree and to generalize the syntactic dependencies obtained during the previous stage, while adding subcategorization information for lexical predicates. Two different transducers have been developed. One is based on a statistical model

[3] www.babelfy.org

[4] The details on Bohnet et al.'s system can be obtained from the original work. It suffices to note here that it produces very competitive scores for all the tasks it performs, for a wide range of languages.

and the other is rule-based. The statistical transducer (see [1, 2] for details) is trained on parallel SSynt and DSynt corpora (see for instance [24] for an example in Spanish). The DSyntS-SSyntS transducer has the potential to be trained for any language in which there are parallel DSynt and SSynt available treebanks. Currently, it is the case for English and Spanish. The rule-based transducer is implemented a graph-transduction grammars that have access to language-specific lexicons to remove the void prepositions and conjunctions [28], when any is available. The rule-based version is available for English, Spanish, German, and French.

3.3 Mapping to abstract representations and frame assignment

For mapping deep-syntactic structures to more abstract linguistic representations, large-scale lexical resources are needed. Unfortunately, such resources are available, at this point, only for English; see, e.g., PropBank [21], FrameNet [13], VerbNet [36], and the mappings between them (SemLink [32]). For this reason, we chose to map all input languages to English.

After the SSynt-DSynt transduction, the obtained structure does not contain any functional words, which tend to be idiosyncratic. The nodes are labeled with meaningful lexemes.[5] Using multilingual resources such as BabelNet (see Sections 3.1 and 5.1), it is possible to obtain the translations of these words into English. Once this is done, the combination of the subcategorization information in the deep-syntactic structure and SemLink allows us to obtain Frame annotations on top of connected predicate-argument structures. The latter follow the principles of the Meaning-Text Theory model, with the addition of a subset of relations such as *Location*, *Time*, etc., which facilitate the further processing. During this step, shared argumental positions are made explicit and idiosyncratic structuring such as the representation of raising and control verbs is generalized.

4 Text Planning: Planning the Summary

Before a summary can be generated, it is necessary to determine, on the one hand, the content that is to be communicated to the user and, on the other hand, the discourse structure of the determined content. These two tasks are commonly referred to in NLG literature as *text* or *document planning* [34].

The production of summaries in our system assumes that summaries are generated in response to a user query in which an entity in the semantic repository is specified. The entity must correspond to a BabelNet synset, identified by any of the multilingual lexicalizations associated to it in BabelNet. If multiple synsets match the string introduced by the user, the user is asked to choose from the available meanings. The summary to be produced should contain content in the semantic repository that is relevant to the queried entity. The task of the text planning module is to determine the set of most relevant content elements and generate an ordered list out of them.

4.1 Ranking semantic structures

Following previous graph-based approaches to text planning [31, 10], our approach adopts a graph view on the content of the semantic repository where nodes correspond to predicate-argument structures produced by the analysis pipeline, and edges indicate entity-sharing relations between nodes. For each query, a *query graph* is created that contains as nodes all predicates that have the user-specified entity as one of its arguments. This initial set is extended recursively with other nodes that share at least one argument with relations already in the graph, up to a fixed depth. The resulting graph serves to constrain the planning task to a set of related contents, in a similar fashion to past works such as [25, 9, 7].

Given a query graph, we formulate the planning of summaries as a *ranking problem*, similar in spirit to other text planning implementations based on ranking contents [9, 7, 14]. In our ranking formulation all nodes in a query graph must be ranked according to some function that indicates their relevance. Consequently, the text plans produced by our method are sorted lists of nodes in a query graph. The ranking starts from an initial distribution of relevance obtained from co-occurrence counts in a corpus of texts analyzed with Babelfy. For each entity annotated in the corpus, we thus estimate its probability of being annotated in the same document as any other entity. Predicate-argument structures are then assigned an initial rank according to the probabilities of their arguments co-occurring with the queried entity.

Some predicates may have none of their arguments annotated in the Babelfy corpus. In this case, we have no empirical basis to assess their relevance. In order to ameliorate this situation, we distribute the relevance from those nodes that do have some probability assigned to them to their neighboring nodes in the query graph. This is achieved by iteratively multiplying the initial distribution of relevance to nodes of the query graph with an adjacency matrix of the query graph which has been modified so that it can be interpreted as a Markov Chain. That is, given a node, its transition probabilities are calculated from the relevance scores of its target nodes and normalized according to the sum of relevance of all nodes reachable from the initial node. This procedure, which is similar to web ranking [11], produces a near-stationary distribution of relevance in which the initial relevance scores have been adjusted according to the graph topology.

4.2 Producing a coherent text plan

As pointed out above, the goal of the text planning module is not only to determine the relevance of content elements with respect to the user query, but also to define a discourse structure for these elements, i.e., an ordering of the elements that enforces a certain degree of coherence in the resulting text. We do so by ensuring that new predicate-argument structures are added to a text plan only if they are semantically related to the content elements already in the plan. More precisely, we guarantee entity-coherence by performing a *graph exploration* of the query graph, which consists in visiting only those nodes that are connected to nodes that have been already visited. This notion of entity-based coherence is inspired by theories of local coherence such as the Centering Theory [18, 33].

Since the edges in the content graph capture argument-sharing relations between predicates, the sequence of visited nodes is such that, for every node, at least one of its arguments is either the requested entity, or an argument that has already been introduced into the plan. The traversal of the query graph is done in a greedy way. Starting from the set of predicates that have the queried entity as their argument, the most relevant node from all those that available is always selected. Once a node is selected, all the nodes in the query graph connected to it become available for selection. The traversal of the graph produces the ordered sequence that constitutes the input of the surface generation pipeline.

[5] This assumption is not entirely true since our DSyntSs still contain, e.g., support verbs (as *deliver* in *John delivered his first speech in the Congress*), which are generally assumed to be void of meaning as well. In genuine MTT DSyntSs, support verbs do not appear as such either. However, we think that this simplification can be tolerated without a too significant loss of quality.

5 Multilingual text generation

The predicate-argument structures produced by the text planning module are obtained from translating words from the source documents into English (see Section 3.3). In order to generate multilingual text, however, it is necessary to map them to linguistic structures that serve as a starting point for multilingual linguistic generation, which, in turn, requires language-specific lexical resources that capture the lexical and syntactic characteristics of each language. In the next subsection, the creation of such multilingual lexical resources is explained in detail.

5.1 Multilingual lexical resources

For multilingual generation, we need to create lexicons for each language we cover. These lexicons must not only contain language-specific vocabulary, but also be linked to our pivot language, namely English. Given that BabelNet senses annotated during the analysis stage are language-independent, we use them as the cross-linguistic link. Below, we detail the creation procedure and structure of the language-specific lexicons used to go from predicate-argument structures and BabelNet synsets to each language.

The languages supported by our multilingual generation pipeline (English, French, Spanish and German) have a satisfactory amount of NLP resources. The experimental compilation of the corresponding language-specific lexicons was done in different stages. First of all, three texts in each language were randomly selected. Thus, a set of eight texts (around 2,400 tokens) was used as base for the language-specific lexicons.[6] Given that word sense ambiguity is a problem inherent to any language, it was necessary to disambiguate and recognize the right sense of a lexical unit before assigning any specific BabelNet id to it. Babelfy, which as explained in Section 3.1, is connected to BabelNet, was used for disambiguation, using the API offered to remotely access the service. As output of this step, a list of unique BabelNet ids (1,013 items in total) was obtained, which served as the basis for creating the lexicons. This list has then been locally enriched with the word form linked to each id in each language. Using this list as base, for each LU, its part of speech, its lemma, its BabelNet id and its government pattern, i.e., its subcategorization frame, are stored. Within the government pattern, the information collected for each argument includes its part of speech, the preposition introducing it (if it is required by the described LU) and the corresponding case. Below, the entries for the same specific BabelNet id in German (a language with case) and in Spanish are shown.

```
   SPANISH                                  GERMAN
"contar_VV_01":_verb_ {              "sagen_VV_01":_verb_ {
   lemma = "contar"                     lemma = "sagen"
   bn = bn:00091011v                    bn = bn:00091011v
   gp = {                                  gp = {
      I = {dpos = "N"}                        I = {dpos = "N"    case = "nom"}
      II = {dpos = "N"}                       II = {dpos = "N"    case = "acc"}
      III = {dpos = "N" prep = "a"}}}         III = {dpos = "N"    case = "dat"}}}
```

From the English structure, the system thus turns to the lexicons to obtain information about the specific characteristics of the sentences to be generated in each language. If no specific information is added, the system interprets that there are no restrictions with respect to the argument in question. Thus, the four compiled parallel language-specific lexicons serve in a direct way for the multilingual generation pipeline, allowing the mapping from English to any of the other languages involved. Potentially, the mapping could be even done not only from English to other language, but from any other language included in the system to each other.

[6] Although it can be argued that the work is based on a small sample of vocabulary, the sample is big enough to test the adopted methodology.

5.2 Hybrid NLG system

The lexical resources described in Section 5.1 are meant to be used together with generation grammars, which are rule systems that produce successively the different layers of representation mentioned in Section 2. In this section, we describe the different submodules of the NLG pipeline, together with their alternative Machine Learning implementations. In order to understand better the process, Figure 2 includes some intermediate structures of this pipeline.

1. Mapping to output language predicate-argument structures
Starting from the structures provided by the text planning module (see Section 4), first, some idiosyncratic transformations are made to adjust the structures to the predicate-argument format understood by our generation pipeline, and then, the English labels of the nodes are translated into the desired target language using the lexicons detailed in Section 5.1.

2. Mapping to syntactic structures
Once genuine predicate-argument structures in the target language are available, the first task is to find which node in each structure is most likely to be the root of the dependency tree. That is, we want to identify what will be the main verb of the sentence, or the word that triggers its appearance. The main node is typically a word (i) that is predicate, (ii) that has more participants than any other predicate of the structure, and (iii) that is not involved in a semantic relation of secondary relevance. Adjectives, adverbs, prepositions and nouns are possible alternatives to verbs when no verb is available. Around the main node, the deep-syntacticization module builds the rest of the syntactic structure of the sentence. In particular, it is able to decide if a main predicate has to be introduced, or what will be realized as an argument, an attribute, or a coordination.

The procedure of the retrieval of deep-syntactic target structures has been successfully tested on around 39,000 sentences: more than 99% of the semantic structures are mapped to well-formed deep-syntactic structures. In the rest of the cases, the generator is unable to produce any syntactic tree and a fallback message is returned.

The next step in the procedure is to obtain surface-syntactic structures, i.e., to generate all functional words and labeling the dependencies with SSynt relations. In the same fashion as for SSynt–DSynt transduction in the case of analysis, we use two alternative approaches for DSynt–SSynt transduction in the case of generation. For languages with limited amount of annotated data (as, e.g., French or German), a rule-based system is preferred, but if multilayered corpora of reasonable size are available (as Spanish and English), training statistical tools is also possible.

For rule-based transduction, we use an adapted version of the MARQUIS generator [38]. MARQUIS had been designed for data-to-text generation. It starts from air quality and meteorology time series, and uses language-specific resources that contain a fine-grained description of all the concepts and words in the air quality domain. Generation in the context of abstractive summarization is a case of text-to-text generation. That is, we cannot focus on the concepts of a specific domain. Rather, any concept can be present in a semantic structure, and there are no lexical resources that are complete enough to contain all of them. As a consequence, MARQUIS's graph-transduction grammars had to be adapted.

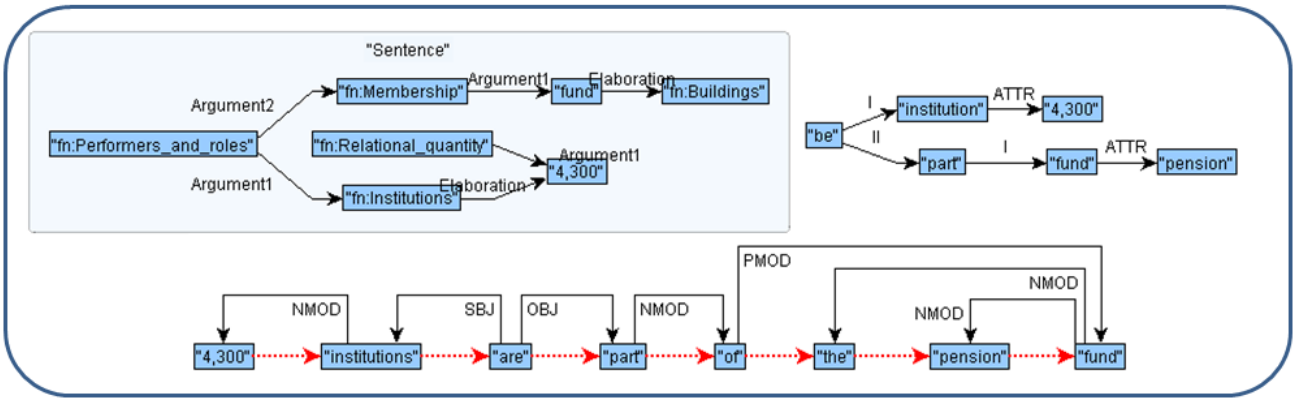

Figure 2. Sample text plan (top left), deep-syntactic structure (top right) and surface-syntactic + morphologic structures (bottom)

For machine learning-based transduction, we developed a series of Support Vector Machine-based transducers; cf., [3] for details.

3. Morphological agreement resolution and surface form retrieval

During the generation of syntactic structures, morphological features of individual words are already inserted (e.g., nominative case for a German subject). During the transition to the morphological structure, agreement is established (using the introduced morphological features and the fine-grained syntactic relations in the SSyntSs) and surface forms of the words are retrieved using a full-form dictionary.

In order to obtain the full-form dictionary, we run the morphological tagger of our surface syntactic parser on a large collection of texts and store each possible combination of surface form, lemma and morphological features. We can therefore retrieve a surface form given a lemma and a set of morphological features. The size of the text collection is crucial in order to ensure a large coverage. For instance, for English, we use the entire Gigaword corpus.[7]

4. Linearization of Unordered Syntactic Dependency Trees

The linearization of unordered syntactic dependency trees, i.e., word order determination, is performed with the state-of-the-art Bohnet et al. [5] linearizer. This linearizer is trained on a surface-syntactic treebank. It produces a statistical model that is capable of determining the word order in a sentence by using mainly surface-syntactic relations and part-of-speech tags.

6 Conclusions and Future Work

In this paper, we presented work in progress for the production of abstractive summaries about specific entities from contents obtained from the analysis of multiple texts. We have covered the resources, tools and techniques applied to obtain the summaries, placing special emphasis on text planning and the multilingual generation component.

In the future, we plan to evaluate the pipeline in one or more domains and use the results to determine what components require improvement. Our first application domain will be the production of multilingual summaries from news articles in the scope of the MULTISENSOR project [8]. We expect our natural language processing tools to perform better in journalistic texts than in more specialized domains that may require models obtained from domain-specific corpora. Crucially, the entities and concepts found in press articles are also more likely to be covered by BabelNet, which plays a crucial role in deciding what contents go into the summary. Specialized domains, e.g. medical or legal texts, may use terminology and make reference to entities only found in specialized kwnowledge and lexical resources. Creating multilingual resources and tools for specific domains is one of the major limitations of applying an abstractive approach to summarization.

The evaluation of our approach will involve both a quantitative evaluation where system-produced summaries are compared to a gold standard of manually written (abstractive) summaries, and a qualitative evaluation in which users will be handed a questionnaire designed at reviewing various facets of the texts: relevance, coherence, grammaticality, readability, etc. Considering the pipeline architecture of our system and the problems introduced by error-propagation, an individual evaluation of each module will also be conducted to identify the most problematic areas. We are particularly interested in finding ways to cope with noisy output from the text analysis component during text planning and linguistic generation, in order to avoid generating ungrammatical or meaningless sentences.

With respect to the lexicons used in the surface generation module, although BabelNet seems very useful in order to obtain interconnected language-specific resources, some issues have been identified which will have to be dealt with in the future. First of all, languages with a very productive compositional process (e.g., German) have BabelNet synsets for which there is no direct correspondence in other languages (in other words, they correspond to more than one synset). Second, and partly as a consequence of the first issue, not all BabelNet synsets correspond to a term in a specific language. Third and last, the procedure for compiling BabelNet synsets can be optimized: if a sequence of lexical units is considered as a multiword unit, then synsets are duplicated (one synset is assigned for each single unit and another one for the multiword unit).

As far as analysis is concerned, we plan to incorporate alternative surface-syntactic parsers based on recurrent neural networks [12, 4], which have been found to be particularly beneficiary for out-of-vocabulary words.

Acknowledgements

This work has been supported by the European Commission under the contract number FP7-ICT-610411.

[7] https://catalog.ldc.upenn.edu/LDC2003T05
[8] http://www.multisensorproject.eu/

REFERENCES

[1] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, 'Deep-syntactic parsing', in *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, (2014).

[2] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, 'Data-driven deep-syntactic dependency parsing', *Natural Language Engineering*, 1–36, (2015).

[3] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, 'Data-driven sentence generation with non-isomorphic trees', in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 387–397, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[4] M. Ballesteros, C. Dyer, and N.A. Smith, 'Improved transition-based parsing by modeling characters instead of words with lstms', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 349–359, Lisbon, Portugal, (September 2015). Association for Computational Linguistics.

[5] B. Bohnet, A. Björkelund, J. Kuhn, W. Seeker, and S. Zarriess, 'Generating non-projective word order in statistical linearization', in *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*, pp. 928–939, Jeju Island, Korea, (July 2012). Association for Computational Linguistics.

[6] B. Bohnet and J. Nivre, 'A transition-based system for joint Part-of-Speech tagging and labeled non-projective dependency parsing', in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1455–1465, Jeju Island, Korea, (2012).

[7] N. Bouayad-Agha, G. Casamayor, and L. Wanner, 'Content Determination from an Ontology-based Knowledge Base for the Generation of Football Summaries', in *Proceedings of the 13th European Natural Language Generation Workshop (ENLG)*, pp. 27–81, Stroudsburg, PA, USA, (2011). Association for Computational LInguistics.

[8] J.C.K. Cheung and G. Penn, 'Unsupervised sentence enhancement for automatic summarization', in *Proceedings of the 2014 Conference on EMNLP*, pp. 775–786, Doha, Qatar, (October 2014). Association for Computational Linguistics.

[9] D. Dannélls, 'The value of weights in automatically generated text structures', *Lecture Notes in Computer Science*, **5449 LNCS**, 233–244, (2009).

[10] S. Demir, S. Carberry, and K.F. McCoy, 'A Discourse-aware Graph-based Content Selection Framework', in *Proceedings of the 6th International Natural Language Generation Conference*, pp. 17–27, Stroudsburg, PA, USA, (2010). Association for Computational LInguistics.

[11] Michelangelo Diligenti, Marco Gori, and Marco Maggini, 'A unified probabilistic framework for web page scoring systems', *IEEE Transactions on knowledge and data engineering*, **16**(1), 4–16, (2004).

[12] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N.A. Smith, 'Transition-based dependency parsing with stack long short-term memory', in *Proceedings of the 53rd Annual ACL Meeting and the 7th International Joint Conference on Natural Language Processing*, pp. 334–343, Beijing, China, (July 2015). Association for Computational Linguistics.

[13] Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato, 'The FrameNet database and software tools', in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pp. 1157–1160, Las Palmas, Canary Islands, Spain, (2002).

[14] A. Freitas, J.G. Oliveira, E. Curry, S. O'Riain, and J.C. Pereira da Silva, 'Treo: Combining entity-search, spreading activation and semantic relatedness for querying linked data', in *In: 1st Workshop on Question Answering over Linked Data (QALD-1) Workshop at 8th Extended Semantic Web Conference (ESWC*, (2011).

[15] K. Ganesan, C. Zhai, and J. Han, 'Opinosis: A graph based approach to abstractive summarization of highly redundant opinions', in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 340–348, Beijing, China, (August 2010). Coling 2010 Organizing Committee.

[16] P.-E. Genest and G. Lapalme, 'Framework for abstractive summarization using text-to-text generation', in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pp. 64–73. Association for Computational Linguistics, (2011).

[17] P.-E. Genest and G. Lapalme, 'Fully abstractive approach to guided summarization', in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 354–358, Jeju Island, Korea, (July 2012). Association for Computational Linguistics.

[18] B. J Grosz, S. Weinstein, and A. K Joshi, 'Centering: A framework for modeling the local coherence of discourse', *Computational linguistics*, **21**(2), 203–225, (1995).

[19] V. Gupta and G.S. Lehal, 'A survey of text summarization extractive techniques', *Journal of Emerging Technologies in Web Intelligence*, **2**(3), 258–268, (2010).

[20] Atif Khan, Naomie Salim, and Yogan Jaya Kumar, 'A framework for multi-document abstractive summarization based on semantic role labelling', *Applied Soft Computing*, **30**, 737–747, (2015).

[21] P. Kingsbury and M. Palmer, 'From TreeBank to PropBank', in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pp. 1989–1993, Las Palmas, Canary Islands, Spain, (2002).

[22] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N.A. Smith, 'Toward abstractive summarization using semantic representations', in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1077–1086, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[23] Fei Liu and Yang Liu, 'From extractive to abstractive meeting summaries: Can it be done by sentence compression?', in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 261–264, Suntec, Singapore, (August 2009). Association for Computational Linguistics.

[24] Simon M., A. Burga, and L. Wanner, 'AnCora-UPF: A multi-level annotation of Spanish', in *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pp. 217–226, Prague, Czech Republic, (2013).

[25] Kathleen R. McKeown, 'The text system for natural language generation: An overview', in *Proceedings of the 20th Annual Meeting on Association for Computational Linguistics*, ACL '82, pp. 113–120, Stroudsburg, PA, USA, (1982). Association for Computational Linguistics.

[26] Y. Mehdad, G. Carenini, and R.T. Ng, 'Abstractive summarization of spoken and written conversations based on phrasal queries', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1220–1230, Baltimore, Maryland, (June 2014). Association for Computational Linguistics.

[27] Igor Mel'čuk, *Dependency Syntax: Theory and Practice*, State University of New York Press, Albany, 1988.

[28] Simon Mille and Leo Wanner, 'Towards large-coverage detailed lexical resources for data-to-text generation', in *Proceedings of the First International Workshop on Data-to-text Generation*, Edinburgh, Scotland, (2015).

[29] A. Moro, A.l Raganato, and R. Navigli, 'Entity linking meets word sense disambiguation: a unified approach', *Transactions of the Association for Computational Linguistics*, **2**, 231–244, (2014).

[30] Roberto Navigli and Simone Paolo Ponzetto, 'Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network', *Artificial Intelligence*, **193**, 217–250, (2012).

[31] M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott, 'ILEX: an Architecture for a Dynamic Hypertext Generation System', *Natural Language Engineering*, **7**, 225–250, (2001).

[32] Martha Palmer, 'Semlink: Linking Propbank, VerbNet and FrameNet', in *Proceedings of the Generative Lexicon Conference (GenLex-09)*, Pisa, Italy, (2009).

[33] Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman, 'Centering: A Parametric Theory and Its Instantiations', *Computational Linguistics*, **30**(3), 309–363, (2004).

[34] Ehud Reiter and Robert Dale, *Building Natural Language Generation Systems*, Cambridge University Press, New York, NY, USA, 2000.

[35] H. Saggion and G. Lapalme, 'Generating indicative-informative summaries with sumum', *Computational linguistics*, **28**(4), 497–526, (2002).

[36] Karin Kipper Schuler, *VerbNet: A broad-coverage, comprehensive verb lexicon*, Ph.D. dissertation, University of Pennsylvania, 2005.

[37] Lucy Vanderwende, Michele Banko, and Arul Menezes, 'Event-centric summary generation', *Working notes of DUC*, 127–132, (2004).

[38] L. Wanner, B. Bohnet, N. Bouayad-Agha, F. Lareau, and D. Nicklaß, 'MARQUIS: Generation of user-tailored multilingual air quality bulletins', *Applied Artificial Intelligence*, **24**(10), 914–952, (2010).

Combining Dictionary- and Corpus-Based Concept Extraction

Joan Codina-Filbà[1] and **Leo Wanner**[2]

Abstract. Concept extraction is an increasingly popular topic in deep text analysis. Concepts are individual content elements. Their extraction offers thus an overview of the content of the material from which they were extracted. In the case of domain-specific material, concept extraction boils down to term identification. The most straightforward strategy for term identification is a look up in existing terminological resources. In recent research, this strategy has a poor reputation because it is prone to scaling limitations due to neologisms, lexical variation, synonymy, etc., which make the terminology to be submitted to a constant change. For this reason, many works developed statistical techniques to extract concepts. But the existence of a crowdsourced resource such as Wikipedia is changing the landscape. We present a hybrid approach that combines state-of-the-art statistical techniques with the use of the large scale term acquisition tool BabelFy to perform concept extraction. The combination of both allows us to boost the performance, compared to approaches that use these techniques separately.

1 Introduction

Concept extraction is an increasingly popular topic in deep text analysis. Concepts are individual content elements, such that their extraction from textual material offers an overview of the content of this material. In applications in which the material is domain-specific, concept extraction commonly boils down to the identification and extraction of *terms*, i.e., domain-specific (mono- or multiple-word) lexical items. Usually, these are nominal lexical items that denote concrete or abstract entities. The most straight-forward strategy for term identification is a look up in existing terminological dictionaries. In recent research, this strategy has a poor reputation because it is prone to scaling limitations due to neologisms, lexical variation, synonymy, etc., which make the terminology be submitted to a constant change [15]. As an alternative, a number of works cast syntactic and/or semantic criteria into rules to determine whether a given lexical item qualifies as a term [3, 4, 7], while others apply the statistical criterion of relative frequency of an item in a domain-specific corpus; see, for example, [1, 10, 22, 24, 25]. Most often, state-of-the-art statistical term identification is preceded by a rule-based stage in which the preselection of term candidates is done drawing upon linguistic criteria.

However, most of the state-of-the-art proposals neglect that a new generation of terminological (and thus conceptual) resources emerged and with them, instruments to keep these resources updated. Consider, for instance, BabelNet http://www.babelnet.org [21] and BabelFy http://www.babelfy.org [20]. BabelNet captures the terms from Wikipedia[3], WikiData[4], OmegaWiki[5], Wiktionary[6] and Wordnet [19] and disambiguates and structures them in terms of an ontology. Wikipedia is nowadays a crowd-sourced multilingual encyclopedia that is constantly being updated by more than 100,000 active editors only for the English version. There are studies, cf., e.g., [11], which show that observing edits in the Wikipedia, one can learn what is happening around the globe. BabelFy is a tool that scans a text in search of terms and named entities (NEs) that are present in BabelNet. Once the terms and NEs are detected, it uses the text as context in order to disambiguate them.

In the light of this significant change of the terminological dictionary landscape, it is time to assess whether dictionary-driven concept extraction cannot be factored in into linguistic and corpus-driven concept extraction to improve the performance of the overall task. The three techniques complement each other: while linguistic criteria filter term candidates, statistical measures help detect domain-specific terms from these candidates, and dictionaries provide terms from which we can assume that they are semantically meaningful.

In what follows, we present our work in which we incorporate BabelFy, and by extension BabelNet and Wikipedia, into the process of domain-specific linguistic and statistical term recognition. This work has been carried out in the context of the MULTISENSOR Project, which targets, among other objectives, concept extraction as a basis for content-oriented visual and textual summaries of multilingual online textual material.

The remainder of the paper is structured as follows. In Section 2, we introduce the basics of statistical and dictionary-based concept extraction. In Section 3, we then outline our approach. The set up of the experiments we carried out to evaluate our approach and the results we achieved are discussed in Sections 4 and 5. In Section 6, we discuss the achieved results, while Section 7, finally, draws some conclusions and points out some future work.

2 The Basics of statistical and dictionary-based concept extraction

Only a few proposals for concept extraction rely solely on linguistic analysis to do term extraction, always assuming that a term is a nominal phrase (NP). Bourigault [5], as one of the first addressing the task of concept extraction, uses for this purpose part-of-speech (PoS) tags. Manning and Schütze [16], and Kaur [14] draw upon regular expressions of PoS sequences.

[1] NLP Group, Pompeu Fabra University, Barcelona, email: joan.codina@upf.edu
[2] Catalan Institute for Research and Advanced Studies (ICREA) and NLP Group, Pompeu Fabra University, Barcelona, email: leo.wanner@upf.edu

[3] http://www.wikipedia.org
[4] wikidata.org
[5] omegaWiki.org
[6] wikitionary.org

More common is the extension of statistical term extraction by a preceding linguistic feature-driven term detection stage, such that we can speak of two core strategies for concept extraction: the statistical (or corpus-based) concept extraction and the dictionary-based concept extraction. As already pointed out, concept extraction means here "term extraction". Although resources such as BabelNet are considerably richer than traditional terminological dictionaries, they can be considered as the modern variant of the latter. Let us revise the basics of both of these two core strategies.

2.1 Statistical term extraction

Corpus-based terminology extraction started to attract attention in the 90s, with the increasing availability of large computerized textual corpora; see [13, 6] for a review of some early proposals. In general, corpus-based concept extraction relies on corpus statistics to score and select the terms among the term candidates. In the course of the years, a number of different statistics have been suggested to identify relevant terms and best word groupings; cf., e.g., [2].

As a rule, the extraction is done in a three-step procedure:

1. **Term candidate detection.** The objective of this first step is to find words and multiword sequences that could be terms. This first step has to offer a high recall, as the terms missed here will not be considered in the remainder of the procedure.
2. **Compute features for term candidates.** For each term candidate, a set of features is computed. Most of the features are statistical and measure how often the term is found as such in the corpus and in the document, as part of other terms, and also with respect to the words that compound it. These basic features are then combined to compute a global score.
3. **Select final terms from candidates** Term candidates that obtain higher scores are selected as terms. The cut-off strategy can be based on a threshold applied to the score (obtained from a training set, in order to optimize precision/recall) or on a fixed number of terms (in that case, the top N terms are selected).

In what follows, we discuss each of these steps in turn.

2.1.1 Term candidate detection

The most basic statistical term candidate detection strategies are based on n-gram extraction. Any n-gram in a text collection could be a term candidate. For instance, Foo and Merkel [9] use unigrams and bigrams as term candidates.

n-gram based concept extraction is straightforward to implement. However, it produces too many false positives, which add noise to the following stages. As already mentioned above, for this reason, most of the works use linguistic features such as part-of-speech patterns or NP markers [16, 10] for initial filtering. See [23] for an overview.

2.1.2 Feature Extraction

Once the term candidates have been selected, they need to be scored in order to be ranked with respect to the probability that they are actual terms.

Most of the proposed metrics are based on term frequency TF, as the number of occurrences of a term in a text collection. In Information Retrieval, TF is contrasted to IDF (Inverse Document Frequency), which penalizes the most common terms. For the task of term extraction, IDF of a term candidate can be computed drawing upon a reference corpus, while the frequency of the candidate term in the target domain corpus can be assumed to be TF, such that we get: $TF_{target} * IDF_{ref}$ [16].

Other measures have been developed specifically for term detection. The most common of them are:

- **C-Value** [10]. The objective of the C-Value score is to assign a *termhood* value to each candidate token sequence, considering also its occurrence inside other terms. The C-value expands each term candidate with all its possible nested multiword subterms that will become also term candidates. For instance, the term candidate *floating point routine* includes two nested terms: *floating point*, which is a term, and *point routine*, which is not a meaningful expression.
  The following formula fomarlizes the calculation of the C-Value measure:

$$\begin{cases} log_2 |t| \, TF(t), & t \text{ is not nested} \\ log_2 |t| \, \left(TF(t) - \frac{\sum_{b \in T_t} TF(b)}{P(T_t)} \right) & \text{otherwise} \end{cases} \quad (1)$$

  where t is the candidate token sequence, T_t the set of extracted candidate terms that contain t, and $P(T_t)$ the number of the candidate terms.
- **Lexical Cohesion** [22]. Lexical cohesion computes the cohesion of multiword terms, that is, at this stage, any arbitrary n-gram. This measure is a generalization of the Dice coefficient; it is proportional to the length of the term and the frequency:

$$LC(t) = \frac{|t| log_{10} \left(TF(t) \right) TF(t)}{\sum_{w \in t} TF(w)} \quad (2)$$

  where $|t|$ is the length of the term and w the number of words that compound it.
- **Domain Relevance** [25]. This measure compares frequencies of the term between the target and reference datasets:

$$DR(t) = \frac{TF_{target}(t)}{TF_{target}(t) + TF_{ref}(t)} \quad (3)$$

- **Relevance** [24]. This measure has been developed in an application that focuses on Spanish. The syntactic patterns used to detect term candidates are thus specific for Spanish, but the term scoring is language-independent. The formula aims to give less weight to terms with lower frequency in the target corpus and a higher value to very frequent terms, unless they are also very frequent in the reference corpus or are not evenly distributed in the target corpus:

$$Relevance(t) = 1 - \frac{1}{log_2 \left(\frac{TF_{target}(t) + DF_{target}(t)}{TF_{ref}(t)} \right)} \quad (4)$$

  where $TF(t)$ is the relative term frequency, while $DF(t)$ is the relative number of documents in which t appears. The document frequency tries to block those terms that appear many times in a single document.
- **Weirdness** [1]. Weirdness takes into account the relative sizes of the corpora when comparing frequencies:

$$Weirdness(t) = \frac{TF_{target}(t) \cdot |Corpus_{ref}|}{TF_{ref}(t) \cdot |Corpus_{target}|} \quad (5)$$

2.1.3 *Term selection*

Each of the metrics in the previous subsection produces a score for each term candidate.The final step is to use the scores produced by the chosen metric to filter out the terms under a given threshold.

Taking the terms sorted by their scores, we expect to have a decreasing precision as we move down to the list, while recall increases. The F-score reaches a maximum around the point where precision and recall cross. The list should be truncated at this point, defining the minimum threshold. But, of course, each dataset provides a different threshold that needs to be set after observing different training sets. Some authors (as, e.g., Frantzi et al. [10]) set an arbitrary threshold; others just measure precision and recall when truncating the list after some fixed number of terms [8].

When more than one metric is available, the different metrics can be combined to produce a single score. There are two main strategies to do it: The first one is to feed a machine learning model with the different metrics and let it learn how to combine these metrics [26]. The simplest procedure in this case is to calculate a weighted average tuned by linear regression; cf., e.g., [22]. The second strategy is to come up with a decision for each metric, trained with its own threshold, and then apply majority voting [27].

2.2 Use of terminological resources for terminology detection

The problem of the use of traditional terminological resources for concept (i.e., term) identification mentioned in Section 1 is reflected by the low recall usually achieved by dictionary-based concept extraction. For instance, studies on the medical domain with the Gene Ontology (GO) terms show a recall between 28% and 53% [17]. To overcome this limitation, different techniques have been developed in order to expand the quantity of matched terms. Thus, Jacquemin [12] uses a derivational morphological processor for analysis and generation of term variants. Other authors, like Medelyan [18], use a thesaurus to annotate a training set for the discovery of terms within similar contexts.

BabelNet is a new type of terminological resource. It reflects the state of the continuously updated large scale resources such as Wikipedia, WikiData, etc. At least in theory, BabelNet should thus not suffer from the coverage shortcoming of traditionally static terminological resources.[7]

BabelFy takes all the n-grams (with $n \leq 5$) of a given text that contain at least one noun, and checks whether they are substrings of any item in BabelNet. To perform the match, BabelFy uses lemmas.

We can thus hypothesize that an approach that draws upon BabelNet is likely to benefit from its large coverage and continuous update.

3 Our Approach

In the MULTISENSOR project, term recognition is realized as a hybrid module, which combines corpus-driven term identification with dictionary-based term identification that is based on BabelFy. Combining corpus-driven and dictionary-based term identification, we aim to enrich BabelFy's domain-neutral strategy with domain information in order to be able to identify domain-specific terms.

Based on the insights from [8, 27], who compare different metrics, we decided to implement the C-Value measure and the Weirdness metric. The C-Value measure serves us to measure the termhood of a candidate term, while the Weirdness metric reveals to what extent a term candidate is domain specific.

However, the Weirdness metric requires some adaptation. The original Weirdness metric can namely range from 0 to infinite, which is not desirable. To keep the possible values within a limited range, we changed the quotient between probabilities to a quotient between IDF's. As a result, Equation 5 is transformed to:

$$DomWeight(t) = \frac{IDF_{ref}(t)}{IDF_{target}(t)} \quad (6)$$

BabelFy offers an API that annotates terms of a given text found in one of the resources it consults (WordNet, Wikipedia, WikiData, Wiktionary, etc.), distinguishing between named entities and concepts. Cf. Figure 1 for illustration. The figure shows the result of processing a sentence with BabelFy's web interface. As can be observed, BabelFy annotates nouns (including multiword nouns), adjectives and verbs (such as *working* or *examine*). In accordance with the goals of MULTISENSOR, we keep only nominal annotations and discard verbal and adjectival ones. Furthermore, BabelFy can be considered a general purpose thesaurus, which is not tailored to any specific domain. For this reason, during domain-specific term extraction as in MULTISENSOR, not all terms that have been annotated by BabelFy should be considered as part of the domain terminology.

To ensure the domain specificity, we index the documents for which the $IDF(t)$ is computed in a Solr index,[8] with a field that indicates the domain to which each of them belongs. This allows us an incremental set up in which new documents can always be indexed and the statistics can be continuously updated.

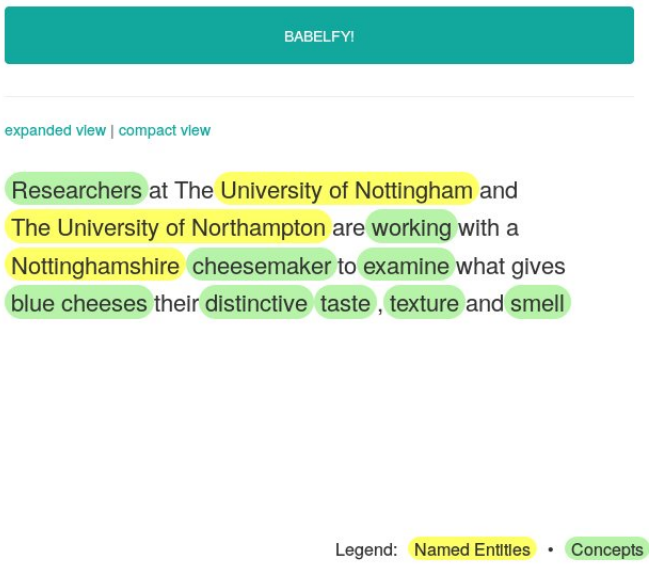

Figure 1. Concepts and named entities detected in a sentence using the BabelFy web interface

The documents indexed in Solr comprise the texts of these documents, together with all the term candidates in them. To index the term candidates, and in order to allow for queries that may match either a full term or parts of it (which can be, again, full terms), we use lemmas (instead of word forms) and underscores between the lemmas to indicate the beginning, middle, and end of the term. The first

[7] Note, however, that even if the Wikipedia is continuously updated, BabelNet is updated in a batch mode from time to time, producing a delay between the crowdsourced changes and their availability in BabelNet.

[8] http://lucene.apache.org/solr

lemma of the term is suffixed with an underscore, the middle lemmas are prefixed and suffixed with underscores, while the last lemma is prefixed with an underscore (for instance, the term candidate *real time clocks* would be indexed as *real_ _time_ _clock*).

At the beginning, the index is filled with the documents that conform the reference and domain corpora. When a new document arrives, we check in both corpora the frequencies of the term candidates as well as the frequencies of their parts as terms and as parts of other terms. To extract these frequencies, several partial matches are required, which can be specified taking advantage of the underscores within the term notation. For instance, to obtain the frequency of the expression *real time* as a term, without that it is part of a longer term, we must search for *real_ _time*. To obtain the frequency of the same sequence of lemmas as part of longer terms, the corresponding query would be *real_ _time_ OR _real_ _time_ OR _real_ _time*. In this last query, the first part would match terms starting with the sequence under consideration (as, e.g., *real time clock*); the second part will match terms that contain the sequence in the middle (as, e.g., *near real time system*); and the last part seeks terms ending with sequence (as, e.g., *near real time*).

Queries in Solr provide the number of documents matching the query. This implies that a document with a multiple occurrence of a term will be counted only once. In some of the formulas of Section 2.1.2, document frequencies are considered, while in others it is the term frequency. In order to minimize this discrepancy, and weight evenly very long and very short documents, we split long documents into groups of about 20 sentences.

To generate term candidates for the statistical term extraction, all NPs in the text are detected. The module takes as input already tokenized sentences of a document. Tokens are lemmatized and annotated with POS and syntactic dependencies. To detect NPs, we go over all the nodes of the tree in pre-order, finding the head nouns and the dependent elements. A set of rules indicates which nouns and which dependants will form the NP. The system includes sets of rules for all the languages we work with: English, German, French and Spanish. Each term candidate is expanded with all the subterms (i.e., n-grams that compose them). The term candidates and all the substrings they contain are then scored using the $C - Value$ and $DomWeight$ metrics. Those with a $DomWeight$ below 0.8 and nested terms with a lower $C - Value$ than the term they belong to are filtered out. The remaining candidates are sorted by decreasing $C - Value$ and, when there is a tie, by $DomWeight$.

After processing the text with BabelFy, we obtain another list of term candidates, namely those that are found in BabelNet. Both lists are merged by intersection and again sorted according to their $C - Value$ and $DomWeight$ scores.

4 Experimental setup

The term extraction methodology described above has been tested for three different use cases. All three use cases are composed by a selection of 1,000 news articles, blogs and other web pages related to different domains. The reference corpus is a set of about 22,000 documents from different sources.

The first use case contains documents about household appliances, with information about both appliances as such and companies involved in the market of household appliances manufacturing and trading. The second use case is about energy policies; it includes news and web pages on green and renewable energy. The third use case covers yoghurt industry; it contains documents about yoghurt products, legal regulations concerning the production and trade with yoghurts, and diary industries.

Table 1. Number of documents and concepts annotated for each use case. The number of indexed chunks indicates in how many different text portions the documents have been split (at sentence boundaries)

Use Case	Name	Num. of documents	Num. of indexed chunks	annotated terms
0	Reference Corpus	21,994	43,808	—
1	Household Appliences	1,000	2,171	123
2	Energy Policies	1,000	1,565	80
3	Yoghurt Industry	1,000	2,096	118

The collection of documents for the three use cases has been extracted from controlled sources, which ensures that the texts within the collection are clean. The documents have been first processed with the goal to detect term candidates, i.e., tokenized, parsed and passed through the NP detector. Once processed, they have been indexed in a Solr index. In addition, all documents have been split into chunks of about 20 sentences to balance the length of the processed texts. In order to evaluate the performance of our hybrid term extraction, for each use case, a set of 20 sentences (from different documents) has been annotated as a ground truth by a team of three annotators.

Table 1 summarizes the information about the different use cases, the reference corpus, the number of original documents, the number of documents after indexing (with some of the documents split as mentioned above), and the number of manually annotated terms for each domain.

5 Evaluation

In order to evaluate the proposed approach to concept extraction, and to observe the impact of the merge of corpus-driven and dictionary-based extraction, we first measured the performance of both of them separately and then of the merge. Table 2 shows the precision and recall of the three runs.

Table 2. Results obtained by the different approaches and the hybrid system in the three use cases ('p' = precision; 'r'= recall)

Use Case	Corpus-driven		Dictionary-based		Hybrid	
	p	r	p	r	p	r
1	38.1%	93.5%	50.3%	76.4%	65.2%	71.54%
2	28.0%	97.3%	36.2%	74.68%	48.3%	70.9%
3	34.8%	79.5%	46.2%	68.4%	60.9%	57.3%
avg	33.6%	90.1%	44.2%	73.2%	58.1%	66.6%

It can be observed that the hybrid approach increases the precision by between 14% and 25% points and decreases the recall by between 7 and 24% . To assess whether the increase of precision compensates for the loss of coverage, we computed the F-score in Table 3.

The table shows that the F-score of the hybrid approach is 7% over the score of the BabelFy (i.e., dictionary-based) approach and 13% above the corpus-driven approach.

The results shown in Tables 2 and 3 have been calculated with all terms provided by corpus-driven and dictionary-based term extraction; only terms with a $DomWeight$ under 0.8 and nested terms

Table 3. F-scores obtained by the different approaches and the hybrid system in the 3 use cases

Use Case	Corpus-driven	Dictionary-driven	Hybrid
1	54.1%	60.7%	68.2%
2	43.5%	48.8%	57.4%
3	48.4%	55.1%	59.1%
avg	49.0%	55.1%	62.1%

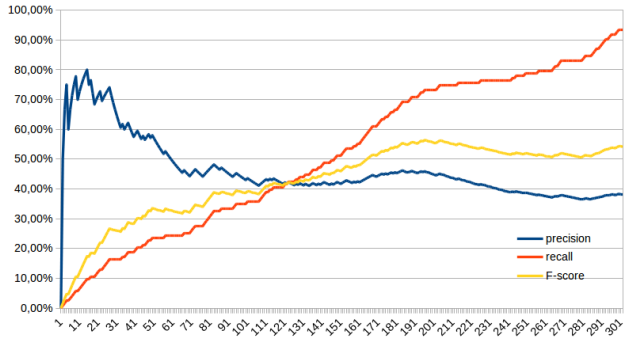

Figure 2. Evolution of precision, recall and F-score as we move down the list of terms generated by the corpus-driven term extraction and sorted by their score

with a $C - Value$ lower than the one of the term they belong to have been filtered out without any further threshold adjustment. In other words, the ordering of the terms according to their $C - Value$ and $DomWeight$ scores has not been considered. If we use only the top N terms with the highest scores, the precision of corpus-based term identification increases. In our current implementation, we do not implement a threshold to cut off the list because the users request the top N terms (with $N = 20$) as a concept profile of a document.

Figure 2 shows how precision, recall and F-score evolve as we move down the list of terms sorted by the score obtained with corpus-driven term extraction (recall that BabelFy does not provide any confidence score).

The score places the most relevant terms at the top of the list, increasing the precision by more than 25 points over the average (as can be observed in the precision/recall/F-score graph, the first 30 terms maintain a precision over 70%).

Figure 3 shows the evolution of precision, recall and F-score for the hybrid term extraction, keeping the ranking provided by the corpus-driven approach. In this case, hybrid term extraction maintains a 100% precision for the first 17 terms and ends with 95% of precision after the first 20 (a single term is wrong among them); 80% precision are maintained for the first 35 terms.

A baseline term identification that does not use scores would obtain a precision of 33%, or 44% using BabelFy and selecting 20 terms at random. When scores are used, the precision of the corpus-driven approach increases up to 47.7%. When both approaches are combined, the average precision for the three use cases increases to 73.6%, resulting in an overall increase of 26% compared to the individual techniques.

6 Discussion

The performance figures displayed in the previous section show that a combination of corpus-driven and dictionary-based term identification achieves better results than in separation, especially when the corpus-driven approach is preceded by a linguistic filtering stage.

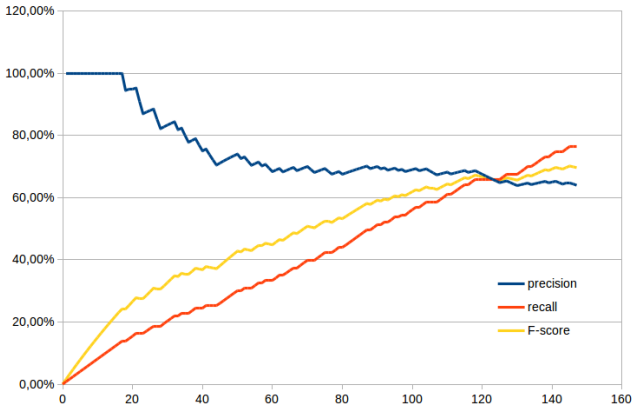

Figure 3. Evolution of precision, recall and F-score as we move down to the list of terms generated by the hybrid system, sorted by the score obtained by statistical metrics

Approaches that are based exclusively on linguistic features serve well to find very rare terms, but they tend to be language- and domain-dependent, which reduces their scalability and coverage. The same applies to approaches that use gazetteers.

Corpus-driven term identification provides term candidates that are domain-specific and common enough to be considered terms, but may be semantically meaningless.

Both corpus-driven and dictionary-based approaches offer a high recall at the expense of low precision because each of them adds its own noise. When combining the two techniques, we increase the precision but lose some recall. However, the decrease of recall is over-compensated by a sufficient increase of precision that leads to the improvement of the F-score. This increase is more evident when we concentrate on terms with a higher score.

The use of an index like Solr to maintain the corpus data allows for the creation of an incremental system that can be updated with upcoming news, making the response dynamic when new concepts appear in a domain.

7 Conclusions and Future Work

We presented a hybrid approach to concept (i.e., term) identification and extraction. The approach combines a state-of-the-art corpus-driven approach with a dictionary lookup based on BabelFy. The combination of both increases the overall performance as it takes the best of both. While statistics are very good in detecting domain-specific terms, dictionaries provide terms which are semantically meaningful.

The use of BabelFy (and thus of BabelNet) allows us to avoid the typical limitation of dictionary-based term identification of coverage. As already argued above, BabelNet, which has been generated automatically from Wikipedia and other resources, is a crowdsourced terminological resource that can be considered to contain a critical mass of terms needed for our task.

Crowdsourced and continuously updated dictionaries ensure the availability of up-to-date resources, but there is still a time offset between the emergence of a new term and its inclusion in the Wikipedia. In the future, it can be insightful to observe the first occurrences of a term and assess its potential status of an emerging concept that cannot be expected to be already in the Wikipedia. This would allow us to give those terms an appropriate score and thus

avoid that they are filtered out.

A relevant topic that we did not look at yet in our current work is the detection of the synonymy of terms, which would further increase the accuracy of the retrieved concept profiles of the documents.

ACKNOWLEDGEMENTS

This work was partially supported by the European Commission under the contract number FP7-ICT-610411 (MULTISENSOR).

REFERENCES

[1] Khurshid Ahmad, Lee Gillam, Lena Tostevin, et al., 'University of surrey participation in TREC8: Weirdness indexing for logical document extrapolation and retrieval (WILDER)', in *Procedings of TREC*, (1999).

[2] Lars Ahrenberg. Term extraction: A review draft version 091221, http://www.ida.liu.se/larah03/publications/tereview_v2.pdf, 2009.

[3] Hassan Al-Haj and Shuly Wintner, 'Identifying multi-word expressions by leveraging morphological and syntactic idiosyncrasy', in *Proceedings of the 23rd International conference on Computational Linguistics*, pp. 10–18. Association for Computational Linguistics, (2010).

[4] Colin Bannard, 'A measure of syntactic flexibility for automatically identifying multiword expressions in corpora', in *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*, pp. 1–8. Association for Computational Linguistics, (2007).

[5] Didier Bourigault, 'Surface grammatical analysis for the extraction of terminological noun phrases', in *Proceedings of the 14th conference on Computational linguistics-Volume 3*, pp. 977–981. Association for Computational Linguistics, (1992).

[6] M Teresa Cabré Castellví, Rosa Estopa Bagot, and Jordi Vivaldi Palatresi, 'Automatic term detection: A review of current systems', *Recent advances in computational terminology*, **2**, 53–88, (2001).

[7] Paul Cook, Afsaneh Fazly, and Suzanne Stevenson, 'Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context', in *Proceedings of the workshop on a broader perspective on multiword expressions*, pp. 41–48. Association for Computational Linguistics, (2007).

[8] Denis Fedorenko, Nikita Astrakhantsev, and Denis Turdakov, 'Automatic recognition of domain-specific terms: an experimental evaluation.', in *SYRCoDIS*, pp. 15–23, (2013).

[9] Jody Foo and Magnus Merkel, 'Using machine learning to perform automatic term recognition', in *Proceedings of the LREC 2010 Workshop on Methods for automatic acquisition of Language Resources and their evaluation methods, 23 May 2010, Valletta, Malta*, pp. 49–54, (2010).

[10] Katerina T Frantzi, Sophia Ananiadou, and Junichi Tsujii, 'The c-value/nc-value method of automatic recognition for multi-word terms', in *Research and advanced technology for digital libraries*, 585–604, Springer, (1998).

[11] Martin Rudi Holaker and Eirik Emanuelsen, 'Event detection using wikipedia', Technical report, Institutt for datateknikk og informasjonsvitenskap, (2013).

[12] Christian Jacquemin, Judith L. Klavans, and Evelyne Tzoukermann, 'Expansion of multi-word terms for indexing and retrieval using morphology and syntax', in *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, EACL '97, pp. 24–31, Stroudsburg, PA, USA, (1997). Association for Computational Linguistics.

[13] Kyo Kageura and Bin Umino, 'Methods of automatic term recognition: A review', *Terminology*, **3**(2), 259–289, (1996).

[14] Gagandeep Kaur, SK Jain, Saurabh Parmar, and Anand Kumar, 'Extraction of domain-specific concepts to create expertise profiles', in *Global Trends in Computing and Communication Systems*, 763–771, Springer, (2012).

[15] Michael Krauthammer and Goran Nenadic, 'Term identification in the biomedical literature', *Journal of biomedical informatics*, **37**(6), 512–526, (2004).

[16] Christopher D Manning and Hinrich Schütze, *Foundations of statistical natural language processing*, volume 999, MIT Press, 1999.

[17] Alexa T McCray, Allen C Browne, and Olivier Bodenreider, 'The lexical properties of the gene ontology', in *Proceedings of the AMIA Symposium*, p. 504. American Medical Informatics Association, (2002).

[18] Olena Medelyan and Ian H. Witten, 'Thesaurus based automatic keyphrase indexing', in *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, pp. 296–297, New York, NY, USA, (2006). ACM.

[19] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller, 'Introduction to wordnet: An on-line lexical database*', *International journal of lexicography*, **3**(4), 235–244, (1990).

[20] Andrea Moro, Alessandro Raganato, and Roberto Navigli, 'Entity linking meets word sense disambiguation: a unified approach', *Transactions of the Association for Computational Linguistics*, **2**, 231–244, (2014).

[21] Roberto Navigli and Simone Paolo Ponzetto, 'Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network', *Artif. Intell.*, **193**, 217–250, (December 2012).

[22] Youngja Park, Roy J Byrd, and Branimir K Boguraev, 'Automatic glossary extraction: beyond terminology identification', in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7. Association for Computational Linguistics, (2002).

[23] Maria Teresa Pazienza, Marco Pennacchiotti, and Fabio Massimo Zanzotto, 'Terminology extraction: an analysis of linguistic and statistical approaches', in *Knowledge mining*, 255–279, Springer, (2005).

[24] Anselmo Peñas, Felisa Verdejo, Julio Gonzalo, et al., 'Corpus-based terminology extraction applied to information access', in *Proceedings of Corpus Linguistics*, volume 2001. Citeseer, (2001).

[25] Francesco Sclano and Paola Velardi, 'Termextractor: a web application to learn the shared terminology of emergent web communities', in *Enterprise Interoperability II*, 287–290, Springer, (2007).

[26] Jordi Vivaldi, Horacio Rodríguez, et al., 'Improving term extraction by system combination using boosting', in *Machine Learning: ECML 2001*, 515–526, Springer, (2001).

[27] Ziqi Zhang, José Iria, Christopher Brewster, and Fabio Ciravegna, 'A comparative evaluation of term recognition algorithms.', in *Proceedings of LREC*, (2008).