The xLiMe system: Cross-lingual and cross-modal semantic annotation, search and recommendation over live-TV, news and social media streams

Lei Zhang *, Andreas Thalhammer, Achim Rettinger, Michael Färber, Aditya Mogadala, Ronald Denaux

Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
Expert System Iberia, Madrid, Spain

A B S T R A C T

Modern Web search engines still have many limitations: search terms are not disambiguated, search terms in one query cannot be in different languages, the retrieved media items have to be in the same language as the search terms and search results are not integrated across a live stream of different media channels, including TV, online news and social media. The system described in this paper enables all of this by combining a media stream processing architecture with cross-lingual and cross-modal semantic annotation, search and recommendation. All those components were developed in the xLiMe project.

Keywords:
Semantic annotation
Semantic search
Semantic recommendation
Cross-lingual
Cross-modal

1. Motivation

The amount of entities in large knowledge graphs (KGs) has been increasing rapidly, enabling new ways of semantic information access, like keyword and semantic queries over entities and concepts mentioned in heterogeneous media items. While *entity search* has become a standard feature, Web search engines are still limited in their semantic processing capabilities: it is not possible to disambiguate search terms manually, search terms in one query cannot be in different languages, the retrieved media items have to be in the same language as the search terms and search results are not collected across a live stream of different media channels.

In this work, we demonstrate a system that intends to break the barriers in between languages and modalities for a seamless semantic access to media streams. We first introduce a real-time processing software architecture and an annotation data model (Section 2) before describing the components for cross-lingual annotation of multilingual text from multiple channels, such as Live-TV, social media and online news (Section 3). The annotated cross-channel media stream allows multilingual and cross-lingual semantic search (Section 4) as well as cross-media recommendation (Section 5). We measure the scalability of the complete system in terms of several metrics (Section 6) before comparing the features of our system to related approaches (Section 7).

This paper provides an overview of components developed in the xLiMe project[1] for the processing of media content across languages, modalities and channels using *explicit semantics*.

2. Cross-lingual and cross-modal processing of semantic media streams

The processing of different multimedia streams is a cost-intensive task. It has been best performed in a distributed manner. Unfortunately, the various sources, their individual particularities, and their distributed processing pose a huge challenge for data integration. As such, we consider three different contributions: (1) multimedia sources; (2) intelligent processing and (3) semantic integration.

The media sources include online news, social media and TV content. All of these sources are multilingual media streams with different and – in the case of social media – changing velocity. The processors include annotation tools for text (i.e., entity linking) and video (i.e., optical character recognition), accompanied by speech-to-text processing (i.e., automatic speech recognition) in the case

* Corresponding author at: Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.

E-mail addresses: l.zhang@kit.edu (L. Zhang), thalhammer@kit.edu (A. Thalhammer), rettinger@kit.edu (A. Rettinger), michael.faerber@kit.edu (M. Färber), aditya.mogadala@kit.edu (A. Mogadala), rdenaux@expertsystem.com (R. Denaux).

[1] http://www.xlime.eu

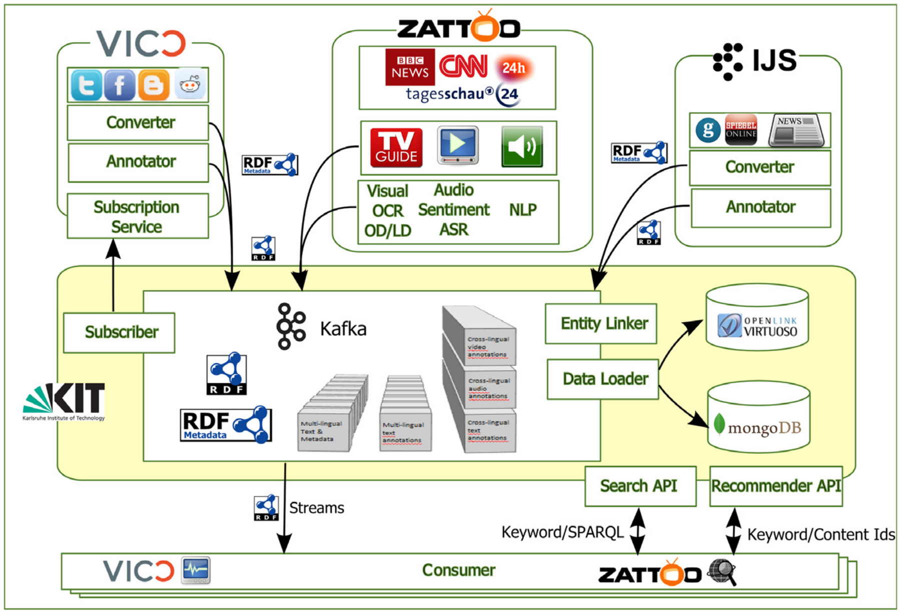


Fig. 1. The xLiMe architecture.

of the audio streams of the TV content. Most of this processing involves a high demand in computational power and sophisticated machine learning models. The semantic integration of different media streams poses the challenge to identify a common model that suits the diversity of the data sources and the output of the processing engines. Further, we combine the processed data with additional background knowledge from knowledge bases.

2.1. System architecture

The architecture of the xLiMe project is divided into multiple components (see Fig. 1). For practical reasons, the multimedia source and initial processing infrastructure is respectively attached to the institutions that provide the respective data. Raw data as well as (intermediately or fully) processed data is directly sent to an Apache™ Kafka message broker that enables a multitude of different topics (communication channels). The partners that provide data processing capabilities provide meta and provenance data in accordance to the xLiMe data model (that will be introduced in the next section). As also the raw data is pushed to the message broker, every partner that has processing capabilities and tools can provide enhanced or alternative services. Along with the message broker, a triple store (i.e., Virtuoso) and a NoSQL database (i.e., MongoDB) provide further data integration and query capabilities. Like this, individual hooks are subscribed to specific Kafka topics and constantly load data into the respective store.

The xLiMe triple store is individually queryable and enables restrictions and aggregates on multiple modalities, languages, and sources. The same accounts for the NoSQL database. This enables the flexible operation of services that build on live streaming data in combination with additional background knowledge. Based on the integrated data in the triple store and NoSQL database, xLiMe components enable us to ask complex questions using SPARQL queries and to search for different media channels using keyword queries.

2.2. Data model

The xLiMe data model is defined as an RDF vocabulary and tailored specifically to the different modalities: text, audio, and video. It extends other vocabularies such as Dublin Core,[2] SIOC,[3] and KDO.[4] Its main scheme is depicted in Fig. 2. Similarly to the Web Annotation Model,[5] it enables to relate text and (parts of) video or audio streams to real world entities. In the xLiMe project we refrained from using the Web Annotation Model in order to reduce the amount of unnecessary blank nodes and thus, at query time, joins. The predicates that define the start and stop positions can be used in a flexible manner and may define character positions, in the case of text, or milliseconds/frame numbers in case of audio/video. In order to describe rectangular fragments of videos or images, there is a specific class that defines the visual position. In any case, the recognized entity should relate to a resource in the knowledge base.

Another particularity of the xLiMe data model – that is not depicted in Fig. 2– is extensive use of named graphs where we use the W3C provenance data model[6] in order to provide meta data for the respective processing of one or more media items.

3. Cross-lingual semantic annotation

In this section, we present *X-LiSA* [1], an infrastructure for *cross-lingual semantic annotation*, which supports interfaces for annotating media data with resources in knowledge bases. It helps to bridge the ambiguity of unstructured data and its formal semantics as well as to transform such data in different languages into a unified representation.

3.1. System architecture

The architecture of *X-LiSA* is shown in Fig. 3, where *cross-lingual groundings extraction* is performed offline to generate the indexes used by the online *cross-lingual semantic annotation*.

Cross-lingual groundings extraction. For matching words and phrases in different languages against entities in knowledge bases,

---

2 http://dublincore.org/documents/dces/
3 http://www.w3.org/Submission/sioc-spec/
4 http://render-project.eu/resources/kdo/
5 http://www.w3.org/TR/annotation-model/
6 https://www.w3.org/TR/prov-dm/

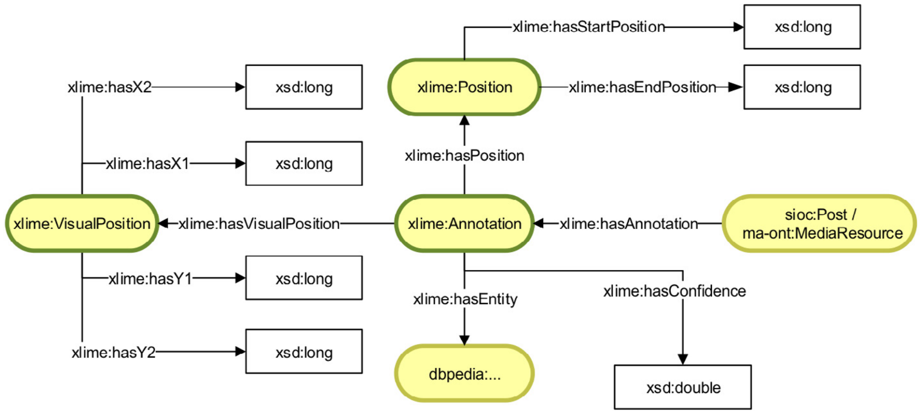


Fig. 2. The xLiMe annotation model.

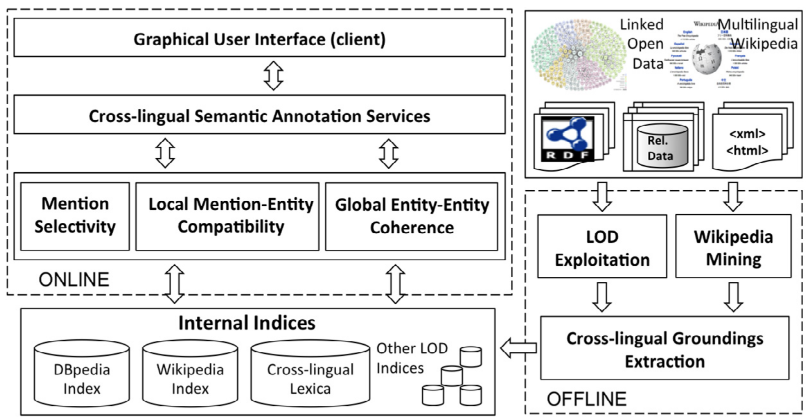


Fig. 3. The system architecture of *X-LiSA*.

X-LiSA relies on the cross-lingual grounding extraction, where we construct the cross-lingual lexica, called *xLiD-Lexica*, by exploiting multilingual Wikipedia to extract the cross-lingual groundings of entities. As Wikipedia provides several useful structures, such as titles of pages, redirect pages, disambiguation pages and link anchors, which associate entities with words and phrases, also called *labels or surface forms*, all of them can be used to refer to the corresponding resources. In addition, Wikipedia pages in different languages that provide information about the equivalent resources are often connected through the cross-language links. Based on the above sources, for each entity grounded in one language we extract its possible surface forms in different languages. Beside that, we also derive the co-occurrence associations between words/phrases in multiple languages and entities in knowledge bases. More details can be found in [2,3].

Mention detection. The first challenge of semantic annotation lies in *mention selectivity* with the goal of detecting the boundaries of mentions in text that are likely to denote entities. To address the challenges of correctness, completeness and emergence of the detected mentions, we employ our recent work [4] that aims to detect both named and nominal entities. Such entity mentions serve as the input of entity disambiguation.

Entity disambiguation. For each mention, its candidate entities are then extracted using *xLiD-Lexica*. While the feature of *mention-entity compatibility* captures the most likely entity behind the mention based on the cross-lingual groundings and the entity that best fits the context of the mention based on the cross-lingual relatedness [5], *entity-entity coherence* collectively captures the related entities as annotations. These features are then employed by the *graph-based disambiguation* to determine the final entity for each mention [6].

3.2. Functionality description

We demonstrate *X-LiSA* in terms of the *cross-lingual lexica*, the *online annotation service* and the use case of *media annotation and querying*.

Cross-lingual lexica. Firstly, we show the extracted cross-lingual lexica *xLiD-Lexica*.[7] The datasets are available as both RDF triples in N-Triples format and plain text files in JSON format. Based on these datasets, we built a SPARQL endpoint and Web interface such that users can easily access the information using SPARQL query language or through the Web interface.

Online annotation service. *X-LiSA* supports interfaces for annotating raw text and Web pages in different languages.[8] A screenshot of the cross-lingual semantic annotation service is shown Fig. 4, where the input is the URL of a German news article, the knowledge base is DBpedia and the output language is English. In order to allow not only users but also software agents to access the functionality of text annotation, we also provide the service, which takes textual data as input and yields the output of annotations in XML.

Media annotation and querying. Within the context of xLiMe project, *X-LiSA* has been widely used to annotate textual data from mainstream new sites, social media and Live-TV, where the following partners have contributed large datasets, which are delivered as streams:

7 http://km.aifb.kit.edu/sites/xlid-lexica
8 http://km.aifb.kit.edu/sites/xlisa

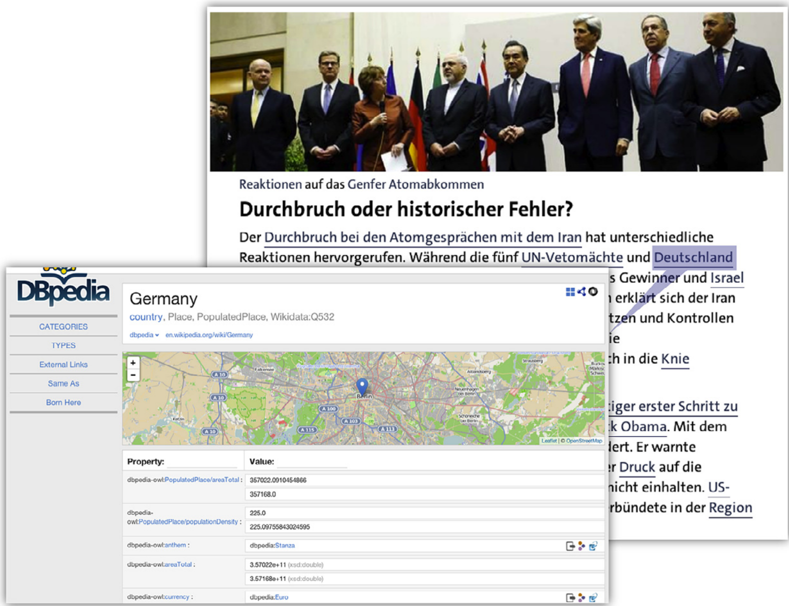


Fig. 4. Annotation service for web pages.

```
PREFIX xlime: <http://xlime-project.org/vocab/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT COUNT(DISTINCT ?media) as ?count ?entity WHERE {
    ?s xlime:hasAnnotation ?a .
    ?s dcterms:source ?media .
    ?s dcterms:created ?date .
    ?a xlime:hasEntity ?entity .
    ?entity dbpedia-owl:manufacturer dbpedia:Mercedes-Benz .
    FILTER (?date > xsd:date(now()-3600*24*14) && ?date < now()) .
}   GROUP BY ?entity ORDER BY DESC(?count)
```

Fig. 5. SPARQL query example.

- *JSI NewsFeed*[9] : news articles crawled from online news sites across the world.
- *VICO*[10] : social media text crawled from forums, blogs, social networks, review sites and others.
- *Zattoo*[11] : text extracted from visual and audible TV data based on the technologies, such as optical character recognition (OCR) and automatic speech recognition (ASR).

Based on the xLiMe annotation model introduced in Section 2.2, we model the annotated media data as RDF triples, which are stored in our triple store. In addition, a SPARQL endpoint is provided for querying the annotated data. For example, given the query "*Which cars produced by Mercedes-Benz were mentioned most in the last two weeks?* ", the SPARQL query shown in Fig. 5 can be used to retrieve the answers.

4. Multilingual and cross-lingual semantic search

X-LiSA offers opportunities for dealing with complex queries on the media data. However, the formal queries, e.g., SPARQL, hinder casual users in expressing their information needs as they might be not familiar with the query's syntax or the underlying ontology. Because keyword search are easier to handle for casual users, we present *XKnowSearch!*, a novel system for entity-based *multilingual* and *cross-lingual semantic search* by translating keyword queries in different languages to their semantic representation [7]. With the help of *X-LiSA* for cross-lingual and cross-modal semantic annotation, *XKnowSearch!* bridges the language barriers between keyword queries and media data and also facilitates query disambiguation and expansion in both manual and automatic manners.

4.1. System architecture

By employing *X-LiSA* for offline semantic annotation, *XKnowSearch!* enables keyword search by capturing keyword queries and media data at the semantic level and also bridging the language barriers. The architecture of *XKnowSearch!* is shown in Fig. 6. In the following, we discuss the online processing components.

Query interpretation. The search process starts with a keyword query in any language, which can even contain keywords in multiple languages. Instead of retrieving media items directly by keywords, XKnowSearch! first finds the *query entity graphs* (*QEGs*), which are subgraphs of the semantic graph of the knowledge base with nodes representing entities and edges describing their relations such that for each query keyword there is at least one entity in the subgraphs matching it.

The first step of *query interpretation* is *keyword matching*. To address the challenge of matching query keywords in different languages to entities, we also make use of *xLiD-Lexica* described in [2,3]. After obtaining the matching entities, the *top-k graph exploration* is then performed on the graph of the knowledge base for finding the top-*k* optimal QEGs. The resulting QEGs represent different semantic interpretations of the keyword query. Thus it can help users to refine the query and influence media item ranking according to the search intents. More details about our approach to query interpretation can be found in [8].

User interaction. Different interpretations of the keyword query, i.e., the generated QEGs, are then presented to users for *selecting* the one that fulfills their search intents. The selected QEG can be further *refined*. From an entity in the QEG, users can navigate its description and the connected entities through their relations in the knowledge base, such that they can add additional entities into the QEG or delete unnecessary ones. After that, the entities in the refined QEG constitute the *query entity vector* (*QEV*), where

9 http://newsfeed.ijs.si
10 http://www.vico-research.com/en
11 http://developer.zattoo.com

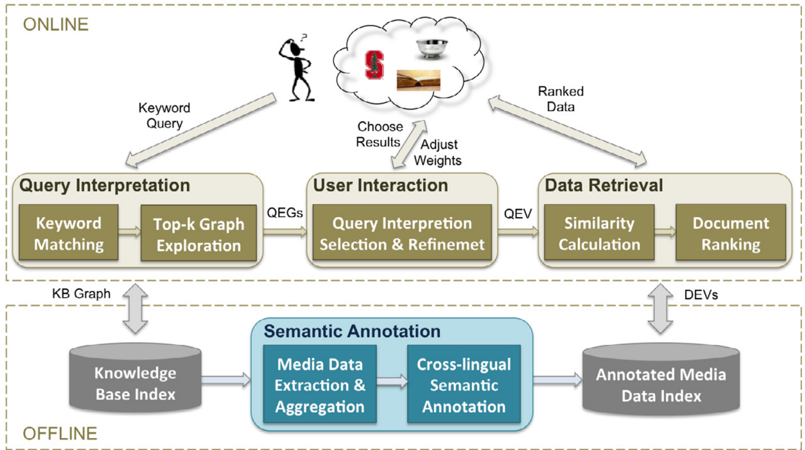


Fig. 6. The system architecture of *XKnowSearch!*.

each entry contains the weight of the corresponding entity, which is calculated by the top-k graph exploration algorithm for query interpretation and can also be adjusted by users. These weights will be leveraged for ranking of retrieved media items in the next component.

We consider *user interaction* as beneficial because it enables the interactive query disambiguation and expansion according to users' search intents. Although refinement can be made more precisely on QEGs than on keywords, user interaction is optional in our system. Users can also search the media items directly without interactive query refinement. In this case, the QEG with highest score obtained by the query interpretation component is selected to generate the QEV.

Data retrieval. For media data retrieval, the entities in the QEV are used to find relevant media items. However, the media items without mentioning the entities in the QEV could also be relevant when they contain entities that are related to the ones in the QEV. Therefore, integrating the related entities into the query can help to cover more complementary information and thus improve the performance of data retrieval. Based on the above observation, we first construct the *expanded query entity vector* (*EQEV*) by *automatically expanding* the QEV with additional related entities.

For each media data item, we construct the *data entity vector* (*DEV*), where the entries contain the confidence scores of the annotations (i.e., the linked entities), which are generated by the offline cross-lingual semantic annotation. It is noted that all the entities in both EQEV and DEV are grounded in the same hub language such that they serve as the bridge to overcome the language barrier between keyword queries and media data. The semantic similarity between the EQEV and each DEV can be calculated based on standard similarity measures, such as cosine similarity, which is then used for ranking of retrieved media items.

4.2. Functionality description

We show four major features of *XKnowSearch!*[12] with the goal of addressing the challenges that traditional keyword search suffers from.

Query flexibility. While traditional keyword search systems do not allow users to be involved in the search process to perform query refinement, XKnowSearch! supports two search modes, namely *direct search* and *indirect search*. The direct search mode performs similar to the current Web search engines like Google. It takes a keyword query as input and retrieves the relevant media items directly without user involvement in the search process. The indirect search mode provides the opportunity for users to understand the meaning of the query entities and the underlying semantic relations between them, such that users are able to refine and extend the information needs. While the direct search enables users to search in a familiar and convenient manner, the indirect search provides users a *more flexible way* to influence the search process according to their search intents.

Query disambiguation. Keywords are naturally ambiguous and this problem is more serious in the multilingual setting because the same keywords could have different meanings in different contexts or languages. In XKnowSearch!, query disambiguation can be performed both automatically and manually. On the one hand, the query interpretation component *automatically eliminates the ambiguity* of the keyword query by taking advantage of the context, i.e., other query entities, and exploiting the semantic graph of the KB to generate the top-k QEGs. On the other hand, users can also *disambiguate the query manually* by selecting the most appropriate QEG and further refining it. As query interpretation, QEG is more informative and expressive than keywords such that users can obtain information about not only entities but also relations between them.

Query expansion. The query keywords are often incomplete in the sense that instead of the full entity name, only the aliases, acronyms and misspellings are usually given by users. *XKnowSearch!* supports query keywords matching entities in their *incomplete* forms. In addition, keyword queries might contain concept names representing a set of associated entities. In *XKnowSearch!*, the matching concepts are automatically expanded into sets of individual entities, which has been discussed in [8] in detail. As query interpretation, QEG is more informative and expressive than keywords such that it can help users to *manually expand* the query by navigating the knowledge graph and adding more intended entities that are used for media data retrieval. The resulting query entities can be *automatically expanded* with additional related entities in XKnowSearch!, which are then used for document retrieval. In the retrieved media items, the entities specified manually by users are distinguished with the ones automatically expanded using different colors (cf. Fig. 7).

Cross-lingual search. Modern Web search engines are still limited in their semantic processing capabilities: search terms in one query cannot be in different languages, the retrieved Web contents have to be in the same language as the search terms and results are not integrated across a live stream of different media channels including online news, social media and Live-TV. In this regard, *XKnowSearch!* aims to break the barriers in between languages and modalities for a seamless semantic access to media streams.

12 http://km.aifb.kit.edu/sites/XKnowSearch

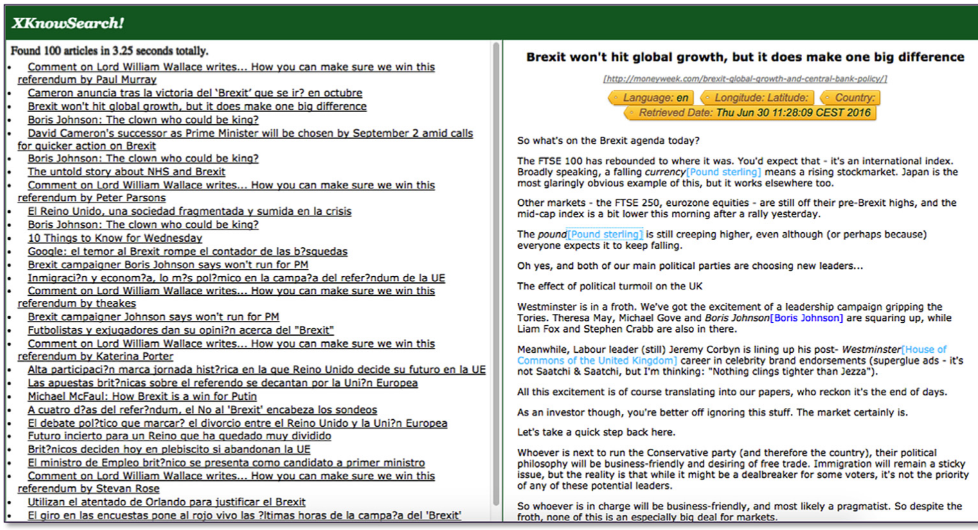


Fig. 7. Example of retrieved news articles for query " 英国 *boris johson*".

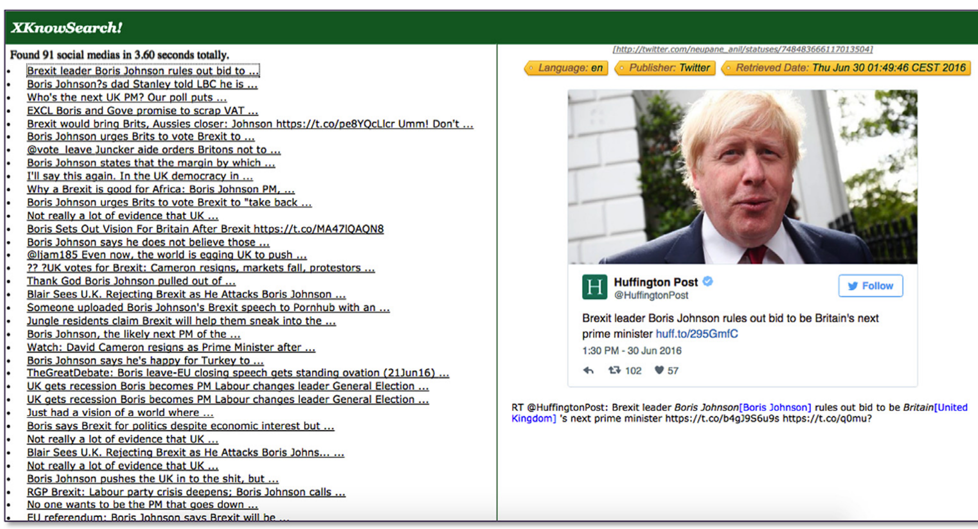


Fig. 8. Example of retrieved social media posts for query " 英国 *boris johson*".

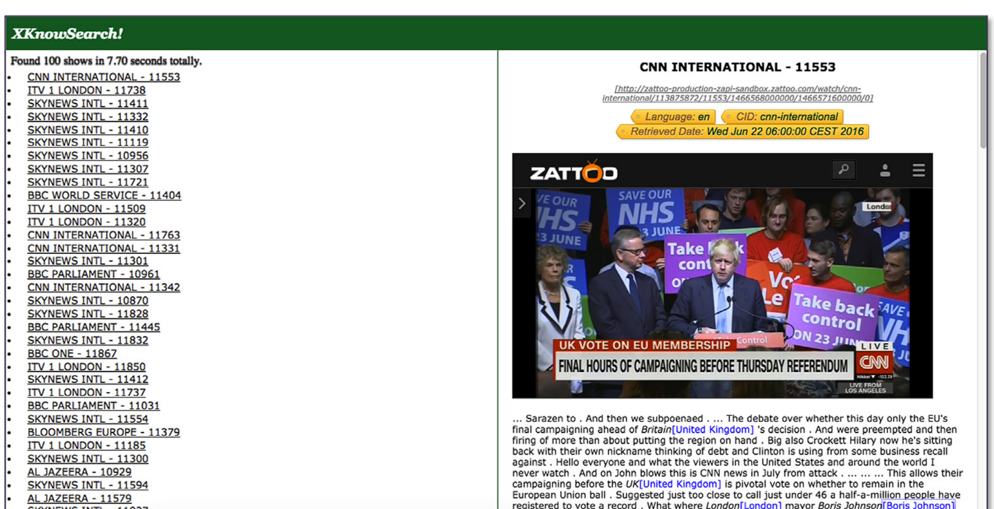
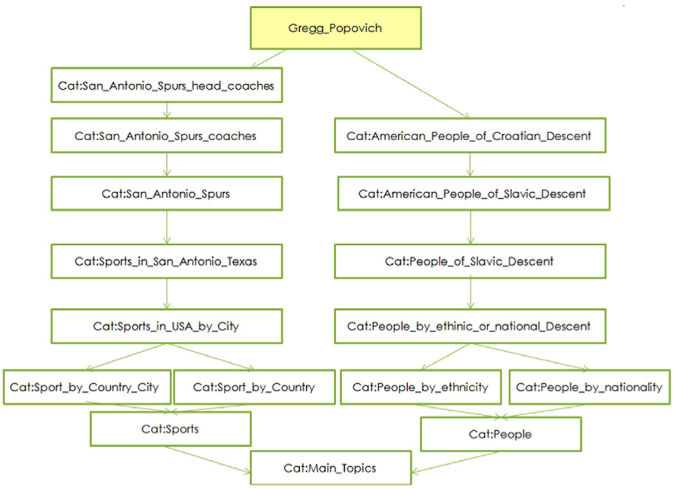
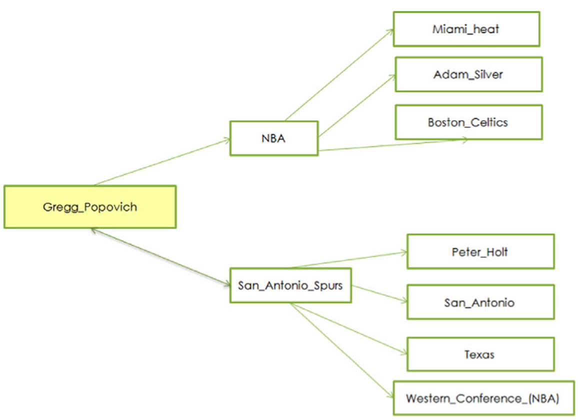


Fig. 9. Example of retrieved TV segments for query "英国 *boris johson*".



(a) Hierarchical entity expansion.



(b) Traversal entity expansion.

Fig. 10. Entity set expansion for DBpedia entity Gregg_Popovich.

Firstly, it enables cross-lingual search in the sense that users can use keyword queries in any language (even in multiple languages) to retrieve multilingual media items. For this purpose, we use the multilingual knowledge base as an interlingua to connect keyword queries and media items across languages. Through the semantic integration of different media streams, *XKnowSearch!* also supports search across different media channels by identifying a common model that suits the diversity of the data sources and combining the processed data with additional background knowledge. Some examples of the retrieved news articles, social media posts and TV segments for the keyword query " 英国 *boris johson*" are shown in Figs. 7–9, respectively.

5. Cross-lingual and cross-media semantic recommendation

Semantic recommendation is considered as a very important problem in various communities such as Social Media, Information Retrieval and Semantic Web etc. Extending the problem to cross-language and cross-media (e.g. news, social media and TV etc.) provide interesting applications. The goal of a cross-lingual and cross-media semantic recommendation system is to find the similar media items posted across languages, modalities and channels. Here, we focus on a knowledge-centric approach to semantic recommendation using explicit semantics, which can be extracted from the data by annotating it with an external knowledge graph. This allows the semantic annotations to be further used for finding similar items.

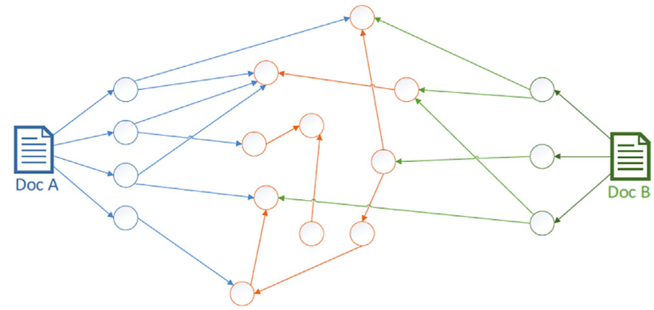


Fig. 11. Bipartite graph constructed from subgraphs of both documents.

5.1. *Knowledge-based recommendation approach*

Given a document, we first use *X-LiSA* to add semantic annotations (i.e. mostly entities). These documents could have belonged to different media such as News, Social Media text such as Twitter, Blogs, TV subtitles/speech transcriptions etc. Once the documents are enriched with entity annotations, the following steps are further executed to achieve entity set expansion:

- **Hierarchical:** Finds the categories and their ancestors using depth information.
- **Traversal:** Finds neighboring entities based on the path length and number of paths.

An example of each of these approaches for the entity Gregg_Popovich, who is an American basketball coach and heads the NBA team San Antonio Spurs, can be found in the Fig. 10.

Once the entities inside documents are expanded using hierarchical and traversal based approaches, they can be used to calculate document similarity. Firstly, a subgraph is constructed from the entities identified for each document. As shown in Fig. 11, the subgraphs of both documents are used to find the bipartite graph and graph-based similarity is then applied by computing the pairwise entity similarities based on the hierarchical and traversal scores [9].

5.2. *Recommendation applications*

To comprehend the knowledge-based approach in the context of xLiMe, we provide two scenarios where we compare news, social media and TV content in two different aspects:

- For a given TV show, find related News and Social Media content as recommendations.
- For a given News article, find the related TV shows as recommendations.

TV → Social Media/News. For recommending social media/news for TV streams, we use the audio of TV streams. Audio is converted from speech to text with state-of-the-art speech transcribing software to be further annotated for recommendation. Similarly, annotations are also accomplished on social media or news articles such that they can be compared to the TV content with the annotations obtained over speech transcriptions.

News → TV. This application reverse the scenario of the first application. Here, the goal is to identify the TV streams. Similar to the earlier scenario, we use the audio of TV streams to be converted to text. Annotation and comparison based on it are then applied on news and transcribed speech as described in the Section 5.1. Fig. 12 depicts the elucidated scenario where the news articles are obtained from different languages to be compared with the TV streams.

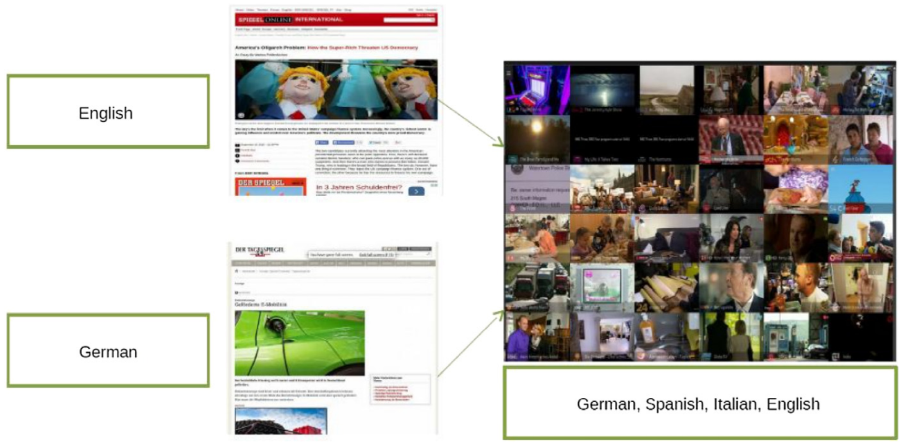


Fig. 12. Comparison between news and TV shows in different languages.

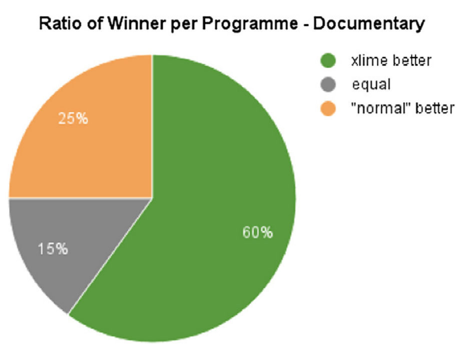


Fig. 13. Percentage of the tested documentaries where xLiMe recommendations were rated better.

5.3. User study

To comprehend the effectiveness of recommendations provided by the xLiMe system to users, we performed A/B testing to identify the user preference for similar TV show recommendations. In the following, we present results of user preference from xLiMe technology and the baseline of Zattoo's "normal" recommendation that uses only electronic program guide (EPG) data.

The TV show recommendations from xLiMe system have been integrated with ZATTOO productive environment.[13] Recommendation results for different TV shows were presented to users and each user was asked to judge the relatedness of the recommendations displayed. Around 16 TV shows of different genres including documentaries, magazines/talk shows and news were chosen for evaluation. While the results did not show significant differences between the two recommendation approaches for magazines/talk shows and news, recommendations for documentaries provided by xLiMe were voted better than Zattoo's "normal" recommendations. As shown in Fig. 13, for 60% of the tested documentaries, xLiMe recommendations have received a better rating (i.e. more test users preferred xLiMe recommendations), whereas for 25% Zattoo's "normal" recommendations were the winner.

6. Empirical analysis and lessons learned

In this section, we discuss some metrics (in particular about data ingestion, annotation throughput and overall latency) we have gathered to provide an idea of the volume and velocity of the data that we can process with the xLiMe architecture. In addition, we provide lessons learned about the actual usage of semantic technologies in the applications within the xLiMe project.

Data ingestion. In terms of data ingestion, we have been ingesting a subset of the available volume of social media items, which we filter based on a number of keywords related to the domains relevant to the use-cases. Although volume can vary substantially, we were able to ramp up from around 100K microposts at the beginning of the project to an average of around 500K microposts per day (as we optimized our entity linking annotators and introduced new use-cases). Depending on the topics being filtered, we reached maximums of around 2 million microposts in a day (these occurred when monitoring the 'Brexit' topic, on the day of the referendum and the days after that as people were commenting on the results). Similarly, for news articles, we could vary the number of news articles between 10K and 350K per day, depending on the demand in terms of languages and topics to be monitored and indexed. Finally, for TV programs, we generally only ingested data that we could process due to the limited number of computing resources for ASR and the various video analyses. This varied from analyzing only 6 channels to analyzing up to 23 channels.

The average number of media items ingested from different channels per day is shown in Table 1. In general, data ingestion of textual data could be greatly increased, as it does not require much in terms of storage and annotation can be delayed or parallelized. Ingestion of video streams is more complicated as we do not store this data, but only keep temporary copies of parts of the stream for processing; furthermore analysis and annotation of video streams is more expensive, which limits the amount of parallelism that can be afforded.

In principle, the xLiMe system support 13 languages including English, German, French, Italian, Portuguese, Spanish, Russian, Chinese, Slovenian, Catalan, Serbian, Croatian and Basque. However, in the xLiMe project we only gathered the data for the languages required by the use cases. While all these 13 languages are supported for news streams, we only deal with social media data in English, German and Spanish, and only followed TV channels in German, English, Spanish, Italian and French. Nevertheless, once media can be converted to text we could support the above 13 languages for all the media channels.

Annotation throughput. In terms of annotation throughput, this varied between 100 and 300 Kb/s for text analytic tasks such as entity linking; this means we are able to process incoming text in near real time. The throughput for video annotation tasks, such as OCR, varied between around 300 to 2000 Kb/s, which for video means that we can only process a few frames per second (per annotator), in order to maximize our resources, we randomly

[13] http://zattoo.com/

Table 1
Statistics about media data generated per day.

Media channels	Avg. #Media-items	Avg. #Entity-annotations	Avg. #Triples
News	60.8K	796.1K	10.68M
Social media	469.4K	1.01M	13.74M
TV	42.4K	75.1K	1.74M
Total	572.6K	1.88M	26.16M

select a few frames per second of incoming video. This allows us to maximize the number of video streams that can be processed by a single machine, at the cost of potentially missing text in a stream if they are on screen for less than a second. The throughput for the ASR component is about 140 Kb/s for a single core; i.e. we need 4 cores or recent CPUs in order to provide "real-time" processing of a single channel. As shown in Table 1, we have measured the throughput of loading annotations and triples into Virtuoso or MongoDB. For the volumes we have encountered, neither loader has become a bottleneck because we only store textual data and annotations.

Overall latency. The latency between ingestion of the data and being able to query the data is between 5 and 10 s for news articles and social media items. During this time, the input text is transferred to the annotators, the text is annotated (entity linking and disambiguation is performed), the result is formatted into the xLiMe data model, pushed to Kafka and distributed to the consumers; one of which loads the data to Virtuoso and another one converts to JSON and loads into MongoDB; both databases perform any indexing tasks before they can return the new data during querying. On top of this latency, there can be more time latency introduced by the providers of the data. The newsfeed also introduces some latency between publication of the news articles, discovery of the article and scrapping etc. Similarly, our social media stream introduces some latency as data is processed from the various source APIs (Twitter, Facebook). For TV streams the latency is around 2 min. The main reason for this delay is that ASR analysis requires some form of context which we achieve by chunking the input stream in 40 s segments. Instead of streaming the video, we first gather 40 s segments, preprocess this to pass it to ASR (and OCR for video analysis) components. These components then analyze the chunks and produce results which, in the case of ASR, need to be passed to a text analyzer. By avoiding this chunking of video segments (which is an implementation issue rather than a limitation of the overall architecture), this latency could be greatly reduced, but we did not spend time pursuing this as we did not have this requirement within the xLiMe project. On top of these 2 min of latency on our side, there is also about a half minute delay between initial broadcast of the TV signal and availability via IPTV.

The above metrics indicate that the xLiMe architecture is suitable for providing near real-time annotations of cross-lingual and cross-modal media. The main bottlenecks we have encountered are the computing resources for video and sound analysis. For the requirements in xLiMe, both the Kafka message bus as the databases (Virtuoso and MongoDB) could be handled by single machines; i.e. we did not have to distribute these systems over multiple machines. For systems that require larger volumes and velocities of data, we expect that this architecture should be able to scale by using the distributed capabilities of Kafka, Virtuoso and MongoDB.

Lessons learned. The xLiMe system is dependent on semantic technologies to a large extent. In particular, the xLiMe data model is defined in terms of various RDF vocabularies. By modeling the data in RDF, a shared data processing infrastructure has been created for data providers, data annotators and client applications to collect, annotate and search multimedia and multilingual data, which facilitates the definition of interfaces for various xLiMe stakeholders and relieves them from designing custom solutions for collecting, annotating and searching the data. Along with the data processing infrastructure, a triple store (i.e., Virtuoso) has been used to provide indexing and querying mechanisms for easier access to the data. We initially developed our end-user applications using the triple store as the back-end, which allowed us to quickly prototype the applications. However, some of the more advanced functionality required the execution of many queries per user interaction, which could not be achieved in adequate response time with the triple store.

Within the xLiMe project, we found it easier to define custom databases specifically tuned for end-user applications with the goal of reducing the response time required. In particular, we have implemented a secondary database and index based on a NoSQL database (i.e., MongoDB) to provide some useful functionality out-of-the-box such as text indexing and defining of custom indices, which allowed finer-grained control about what needs to be indexed. Furthermore, converting the RDF data to custom objects allowed to pre-aggregate the data in a way that is closer to that required by the end-user applications (at the cost of potentially not being able to re-use the database for other applications).

In summary, using semantic technologies, e.g., by modeling data in RDF, enables us to create an extensible integration platform for the xLiMe stakeholders. While the triple store (e.g., Virtuoso) can provide a uniform manner of accessing the data, some features in it are missing (e.g., text indexing). Therefore, a secondary database with more general functions (e.g., MongoDB) can help with supporting ad-hoc indexing and querying to improve the performance of various applications.

7. Comparison to other systems

For comparing the xLiMe system with other system, we introduce the following features, which expose the characteristics of the xLiMe system:

- *Cross-lingual keyword search*: Does the system support cross-lingual keyword search (i.e., keyword search where the entered keywords of one query can be in a different languages and the language of the documents to be retrieved do not need to match the languages in the query)?
- *Semantic search*: Does the system support a semantic search functionality, including a word-sense-disambiguation?
- *Possibility to use complex queries:* Can complex queries be executed beyond searching with a set of keywords for a set of entities?
- *Cross-modal search:* Does the system provide a search over different modalities (e.g., text, social media, video, images, audio)?
- *Interactive query refinement:* Does the system provide a (optional) step between keyword entering and document retrieval, where the query can be refined by the user?
- *live-updates:* Does the system also work with live-data streams?

In the following, we consider the systems which facilitate at least one of the mentioned aspects. Table 2 gives an overview of the related systems. We can see that especially the capability of searching cross-lingually and the possibility to refine the query interactively is rarely given for those systems. We now present

Table 2
Overview of related systems.

Project/System	Cross-lingual keyword search	Semantic search	Possibility to use complex queries	Cross-modal search	Interactive query refinement	Live-updates
BrexitAnalyzer		✓	✓		✓	
CAPER		✓		✓		✓
EUTV		✓	✓	✓	✓	✓
Khreshmoi		✓	✓	✓		
MultiSensor		✓	✓	✓		✓
NoTube				✓		
PHEME	✓	✓	✓	✓		✓
TOSCA-MP		✓		✓		
TrendMiner		✓	✓			✓

more details about the related systems and describe the differences to xLiMe:

Brexit Analyzer[14]: This systems is designed for analyzing tweets in real-time. It is based on the GATE TwitIE system: As part of the pipeline, tweets get parsed, topics get detected, and sentiment analysis is performed. For formulating semantic search queries, a dedicated graphical user interface was developed. The analysis possibilities range from getting the most frequently mentioned words, topics, themes, sentiments, and URLs. In contrast to xLiMe, no other data than tweets are used and no cross-lingual search is provided.

CAPER [10][15] : CAPER stands for "Collaborative information, acquisition, processing, exploitation, and reporting for the prevention of organized crime." The system supports the automatic collection and analysis of unstructured text and audiovisual contents such as video, audio, and images. By creating a semantic network of entities, a limited semantic search functionality is provided. The focus of the project is on providing evidence and warnings for better assessing threads and for better understanding foreign countries and cultures. Contrary to xLiMe, no cross-lingual functionality is provided and no TV data is used.

EUTV [11]: The system of EUTV is designed for aggregating multimedia information streams coming from media RSS and audio and video sources. The system is similar to xLiMe as it also provides a semantic search and incorporates several modalities. Also query refinement and live updates are possible. The focus is here purely on TV data and not on cross-lingual search.

Khreshmoi[16] : This system is similar to xLiMe, as it links information extracted from unstructured or semi-structured biomedical texts and images to semantically-structured data in knowledge bases. Hence, similar to xLiMe, knowledge bases are involved. Multilingual queries are possible, but no cross-lingual queries. Khreshmoi has no focus on the timeliness of the data in the system (no live updates) and also no query refinement step.

MultiSensor [12][17] : MultiSensor, which stands for "Mining and Understanding of multilinguaL contenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation", is one of the most related projects compared to xLiMe. In the MultiSensor system various views disseminated via TV, radio, websites and social media are semantically integrated. In the heart of its multimedia retrieval framework, the multilingual media is analyzed. Thereby the focus is on determining topics in the media and on providing multiple views on the topics. Compared to xLiMe, the MultiSensor system targets for providing a more holistic view on the multilingual multimedia data, pointing out differences in the presentation of information.

NoTube [13][18] : NoTube is similar to xLime in that it integrates Web data and TV data. However, NoTube is designed for recommendation: the task is to provide personalized news, a personalized TV guide and adaptive ads. In total, the aim is to enhance the TV experience. No cross-lingual search is provided, as the recommendations are displayed only in the given language.

PHEME[19] : PHEME provides a framework for analyzing user-generated content (e.g., social network texts) and is therefore similar to xLiMe. PHEME is, however, designed for considering veracity, the fourth big data dimension: For each input text the system determines whether the contained information is correct or not. PHEME is very similar in the technological design to the xLiMe architecture: As xLiMe, it is based on Apache Kafka, it provides APIs for structured data querying, and it uses RDF for modeling the data. PHEME, however, only considers tweets and RSS texts and no interactive query refinement step is integrated.

TOSCA-MP[20] : This system is related to xLiMe in the sense that it provides multimodal information extraction and semantic search over TV, radio, and online content. Ontologies are used for annotation and search over media repositories. Cross-lingual search is not provided.

TrendMiner [14][21] : TrendMiner provides real-time methods for the cross-lingual mining and summarization of large-scale streaming data. The TrendMiner text processing pipeline only provides a language detection module, but no cross-lingual search. The architecture is limited to the processing of social-media text. Hence, it is not so complex as the xLiMe system, which incorporates more modalities and more search functionalities.

8. Conclusions and future work

In this work, we demonstrated a system which was built to break the barriers in between languages, channels and modalities for a seamless semantic access to media streams. Access is provided by multilingual keyword search, iterative entity search or SPARQL queries. TV-segments, social media posts and news articles matching the query can be monitored live in a media stream.

Regarding future work, the described system is relying on explicit semantics only, restricting it to given entities in knowledge bases that can be annotated in text. Recent progress in cross-lingual and cross-modal representation learning enables a different retrieval approach that is not restricted to existing entities in knowledge bases. Integrating those two approaches without losing the explainability of explicit semantics is a promising future research direction.

14 https://gate4ugc.blogspot.de/
15 http://www.fp7-caper.eu/
16 http://khresmoi.eu/
17 http://www.multisensorproject.eu/
18 https://notube.tv/
19 https://www.pheme.eu/
20 http://tosca-mp.eu/
21 http://www.trendminer-project.eu/

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 611346.

References

[1] L. Zhang, A. Rettinger, X-LiSA: cross-lingual semantic annotation, PVLDB 7 (13) (2014) 1693–1696.
[2] L. Zhang, M. Färber, A. Rettinger, xLiD-Lexica: Cross-lingual Linked Data Lexica, in: LREC, 2014, pp. 2101–2105.
[3] L. Zhang, A. Rettinger, S. Thoma, Bridging the Gap between Cross-lingual NLP and DBpedia by Exploiting Wikipedia, in: NLP & DBpedia 2014 At ISWC, 2014.
[4] L. Zhang, Y. Dong, A. Rettinger, Towards entity correctness, completeness and emergence for entity recognition, in: WWW (Companion Volume), 2015, pp. 143–144.
[5] L. Zhang, T. Tran, A. Rettinger, A theoretical analysis of cross-lingual semantic relatedness in vector space models, in: ICTIR, 2015, pp. 241–250.
[6] L. Zhang, A. Rettinger, Philipp, Context-aware entity disambiguation in text using markov chains, in: WI, 2016.
[7] L. Zhang, M. Färber, A. Rettinger, XKnowSearch! exploiting knowledge bases for entity-based cross-lingual information retrieval, in: CIKM, 2016.
[8] L. Zhang, A. Rettinger, J. Zhang, A knowledge base approach to cross-lingual keyword query interpretation, in: ISWC, 2016.
[9] C. Paul, A. Rettinger, A. Mogadala, C.A. Knoblock, P.A. Szekely, Efficient graph-based document similarity, in: ESWC, 2016, pp. 334–349.
[10] C. Aliprandi, J. Arraiza Irujo, M. Cuadros, et al., CAPER: Collaborative Information, Acquisition, Processing, Exploitation and Reporting for the prevention of organised crime, in: C. Stephanidis (Ed.), HCI International, Springer, Cham, Switzerland, 2014, pp. 147–152.
[11] G. Becchi, M. Bertini, A.D. Bimbo, A. Ferracani, D. Pezzatini, A distributed system for multimedia monitoring, publishing and retrieval, Procedia Comput. Sci. 38 (2014) 100–107.
[12] S. Vrochidis, I. Kompatsiaris, G. Casamayor, et al., Multisensor: development of multimedia content integration technologies for journalism, media monitoring and international exporting decision support, in: ICME, 2015.
[13] B. Schopman, D. Brickly, L. Aroyo, C. van Aart, V. Buser, R. Siebes, L. Nixon, L. Miller, V. Malaise, M. Minno, et al., NoTube: making the Web part of personalised TV, in: WebSci, 2010.
[14] D. Preotiuc-Pietro, S. Samangooei, T. Cohn, N. Gibbins, M. Niranjan, Trendminer: an architecture for real time analysis of social media text, in: RAMSS, 2012.