Complex Event Extraction from Real-Time News Streams

Alexandra La Fleur
Corporate Semantic Web Research Group
Freie Universität Berlin
alexandra.lafleur@fu-berlin.de

Kia Teymourian
Computer Science Dept
Rice University
kiat@rice.edu

Adrian Paschke
Corporate Semantic Web Research Group
Freie Universität Berlin
paschke@inf.fu-berlin.de

ABSTRACT

Information overload on news data is a known problem these days. People and organizations have an increasing demand for extraction of relevant information from massive amounts of news data arriving in real-time as news streams. In this paper, we present a novel approach for real-time extraction of news, based on user specifications and by using background knowledge from specific news domains. We create a powerful filtering service which limits the news data to the concrete and essential preferences of a user. In our approach, enrichment of real-time news with background knowledge is a preprocessing step. We use a Complex Event Processor to detect complex events from the enriched articles and match them to the user specified query. Each time a news article is matched, its result is notified to the user immediately. Our experimental evaluation shows that our approach is feasible for detecting news in real-time with high precision and recall.

1. MOTIVATION

The sequential ordering of news articles reporting about daily events happening in the world can include special meanings for businesses that cannot be extracted by considering the news articles separately. For example in the stock market domain the sequences of news about the companies, products or their company-company relations are of special interest. Extraction of such relations between news articles that build complex events require steady monitoring of news data streams. In addition, such systems require access to semantic background knowledge about the news domain, in order to be able to understand the relations between concepts/entities. Existing systems are able to treat single news articles, but our goal is to additionally recognize complex relations in multiple news articles over time.

We provide an approach that combines semantic technologies with natural language processing techniques and news data stream processing using Complex Event Processing (CEP) [11] to evaluate news articles in real-time and extract meaningful complex events from the news data stream. This knowledge-based approach allows the computer to understand the topics and build up connections to related data. We process news data in real-time, without storage need, and notify users about the matched queries immediately. Additionally, CEP offers the possibility to observe events in respect to an event order, which is a valuable feature in the news domain. With this, we can detect complex events and their relations over time. Correlations between different news articles can be declared in the query, and automatically be recognized by the processing engine.

This is of special interest for business people, because today's businesses are more time critical and decision makers need to react to complex business events very close to real-time. Our approach provides a solution, so that news can be filtered in real-time based on background knowledge and the *right responsible actor can know about the important business news at the right time*.

In the following, we first present our approach for filtering news streams based on background knowledge in real-time and explain the architectural concept and components in detail (Sec. 2). Then, we describe our proof-of-concept implementation and the used tools to realize each component (Sec. 3). We have done two kind of evaluations of our concept, one based on the type of news (business, sport, technology, entertainment, health) and one to proof the precision and recall of news extraction (Sec. 4). Finally, we discuss state-of-the-art news extraction systems and compare these to our approach (Sec. 5), summarize our contribution and provide an outlook on future (Sec. 6).

SEMANTiCS'15, September 15-17, 2015, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2814864.2814870.

2. COMPLEX EVENT EXTRACTION

Our approach is designed on a data processing workflow that is used to process the incoming news data stream in different steps. In this section, we fist describe some abstract examples of informal user queries that we are interested in (more concrete examples are provided in Sec. 4) and later describe our data stream workflow.

For this approach we focus on user queries that observe sequences in multiple articles. For example the interest in two news articles where the first news articles (news A) includes news about a politician and a concrete organisation (company X) that is followed by another news article (news B) which includes another company (company Y) that is a subsidiary of the first company (company X). The following pseudo code query shows how these can be described by using an event processing language.

"all newsA (type=politician and entity=company X) followed by all newsB (parentOrganisation=company X)"

Using machine learning and natural language processing techniques, a *Named Entity Recognizer* extracts named entities and concepts within a news article which represent the topics and content. For each article, we determine whether it contains entities/attributes of subscribed user queries. If so, we enrich this article with the knowledge that we retrieve from knowledge bases.

At this state, we have a set of news articles that contain semantically annotated and enriched entities. To evaluate each news article in respect to the user queries, a Complex Event Processor handles the enriched data as simple raw events. The user query serves as an event detection pattern that the raw events are passed through. Once an event matches the pattern, an event listener fires and notifies the user about the matching news article immediately.

As described above, the news data stream is processed in different steps and the data items are mapped to other enriched/processed data types. Figure 1 depicts the main idea of our concept for knowledge-based complex event extraction from real-time news streams.

Entity Enrichment Step: We enrich the extracted named entities from each given news article with background knowledge extracted from knowledge bases that includes domain knowledge about the specific news domain. They provide an expressive language to express information, concepts and interests [13]. For the retrieval of the required background knowledge we use SPARQL queries. That for, we dynamically build up SPARQL queries to determine whether an entity features a specific attribute from the user query.

If the entities of a news article do not correspond the attributes of a user query this news item is discarded, since it is not of interest to the user. With this prefiltering, we limit the data to the essentials. If the entities do correspond to the user query, we enrich them with these attributes. This means that if the user query contains the interest in organizations, and the news article contains the entity Microsoft, this news article is enriched with type "organization".

Complex Event Detection Step: The Complex Event Processor receives the user query as input, treats the enriched news articles as events, and matches them to the pattern. An event listener notifies the user in real-time as soon as an article matches the interest query.

Pattern descriptions are based on the event specification language SNOOP [5]. Temporal event operators define combinations of events in respect to each other. Typical operators are conjunction (AND), disjunction (OR), negation (NOT) and sequence (SEQ).

For our use case to detect news articles in a timely manner, the SEQ operator (->) is the most important one. The events are evaluated as sub expressions and stored locally, until the rest of the expression becomes true. First, the left hand expression must turn true to evaluate the right hand expression. For instance, the pattern *"every NewsA -> every NewsB"* has the operational semantic to match for every event NewsA followed by every event NewsB[1]. Considering the following example news article event sequence:

[A_1, B_1, C_1, B_2, A_2, D_1, A_3, B_3, E_1, A_4, F_1, B_4]

[2] the pattern matches on

- B_1 for combination {A_1, B_1},
- B_2 for combination {A_1, B_2},
- B_3 for combination {A_1, B_3}, {A_2, B_3} and {A_3, B_3}
- B_4 for combination {A_1, B_4}, {A_2, B_4}, {A_3, B_4} and {A_4, B_4}

This example shows the event consumption policy of the "every" operator and the variety of the matched events. The SEQ operator allows the definition of an intended order over occurring news articles, and the every keyword ensures that we don't miss any information.

Now, we will introduce simple example queries that fit our domain.

An example pattern that retrieves all news events independent of their content looks like this:

```
every (a=NewsEvent)
```

Another possible pattern fires if a news article contains Barack Obama as an entity. The engine processes the customized value() method, which checks for the entry "Barack Obama" in the news articles enriched entity list.

```
every (a=NewsEvent(
value('entity_ _ _ _Barack Obama'))=true)
```

It is also possible to query specific attributes from a knowledge graph like DBpedia. The following pattern returns all news articles that include organizations.

```
every (a=NewsEvent(
value('rdf:type_ _ _ _Organization')=true))
```

The interconnection of different attributes and entities is possible by separating the value()-methods with comma. This pattern returns all articles about organizations that produce microchips.

```
every (a=NewsEvent(
value('rdf:type _ _ _Organization')=true,
value('dbpedia-owl:product_ _ _ _Microchip')=true))
```

With the help of the SEQ operator, illustrated as an arrow, we can observe multiple news articles in a timely order. We declare the interest in two articles where the first one is about Microsoft and a financial problem, and that is followed by an article about a politician and Microsoft.

```
every (a=NewsEvent(
value('entity_ _ _ _Microsoft')=true,
value('entity_ _ _ _Financial Problem')=true))
-> every (b=NewsEvent(
value('rdf:type_ _ _ _dbpedia-owl:OfficeHolder')=true,
value('entity_ _ _ _Microsoft')=true))
```

Real Time Detection: The term real-time system is defined as *"a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure."* [8] and *"a computer system where the correctness of the system behavior depends not only on the logical results of the computations but also on the physical time when these results are produced. By system behavior we mean the sequence of outputs in time of a system."* [7].

In our system, real-time processing means the processing of data as data stream which happens immediately after data generation. We administer information directly on the stream and do not have latencies due to data storage. Since the respond time of our system depends on the respond times of the external components we execute news enrichment in parallel to handle the volume of data.

The Complete Process: The complete process of detecting complex events from news stream is summarized in the following Algorithm 1.

[1] In our context, NewsA and NewsB represent different news articles.

[2] Example based on Esper EPL Reference, June 2015

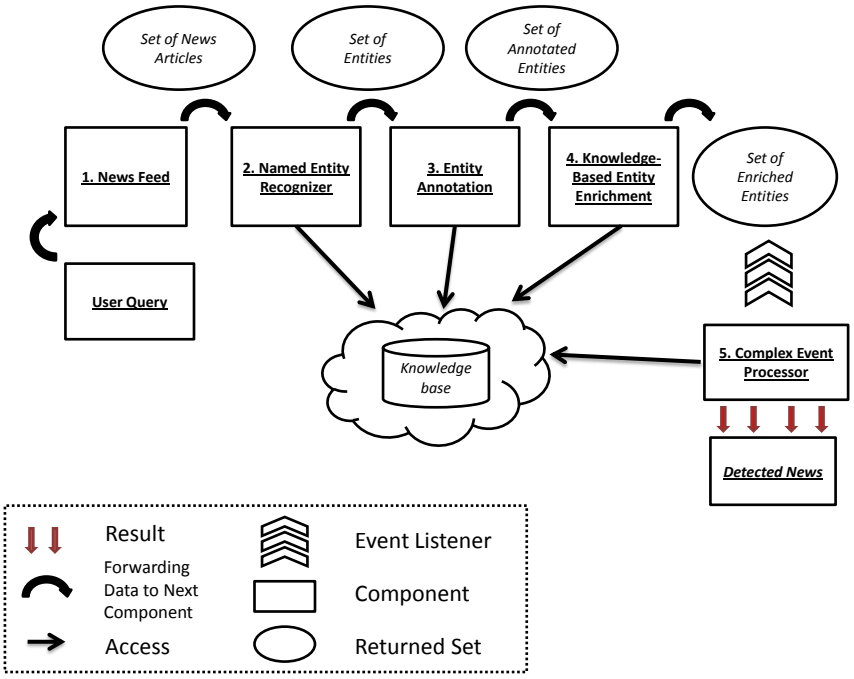

Figure 1: Overview of our Knowledge-Based Complex Event Extraction System

3. IMPLEMENTATION

A news feed provides data as a set of attribute-value pairs where we extract the link, the title and the content of a news article. From these attributes, we build a news item which is ready to be processed by the Named Entity Recognizer.

Saif et al.[15] evaluated the semantic concept extraction outputs from three popular Named Entity Extraction Systems: AlchemyAPI[3], Zemanta[4] and OpenCalais[5]. Because of the best results for the extraction of entities and entity-concept mappings, we use AlchemyAPI as a Named Entity Recognizer for our system. It is an online natural language processing service available through a REST endpoint and an access key.

The DBpedia data set[6] supplies a large ontology which is derived from Wikipedia since Wikipedia is nowadays one of the central knowledge sources of mankind. The DBpedia project builds a large-scale, multilingual knowledge base by extracting structured data from Wikipedia editions in about 111 languages [9]. The DBpedia 3.9 English version currently describes 4.0 million "things" with 470 million "facts" in it.

To access the data held by this ontology, we use the Apache Jena library[7] to access our DBpedia endpoint (our local mirror of DBpedia) via SPARQL. We check the existence of specific attributes of a DBpedia resource with

ASK <resource> <attribute> <value>

and attach them to the entity, if existent. For all entities belonging to the same news article, we collect these attributes together and store them in a Key-Value-Map.

Esper[8] is used as Complex Event Processing engine. It is designed to make it easier to build Complex Event Processing applications, because it convinces with a high throughput, low latency and the ability to process complex computations, like the detection of patterns in events[9]. Complex events can be extracted from an event stream by the SQL-like Event Processing Language (EPL). Event pattern matching is used to trigger complex event listeners to execute related event processing [16]. Unlike traditional databases, Esper stores queries to run the real-time data against these queries. After thorough enrichment of news articles, the transformation into events, that hold an occurrence time stamp and the news data, is necessary for the recognition by the Complex Event Processor in the correct order[10]. If a match between pattern and article exists, the listener activates and can access the recognized items.

4. EVALUATION

We deliver an evaluation of entity types that are present within specific news categories as an interesting additional benefit. This information can be leveraged to define beneficial patterns, since we know what is likely to be detected. Another appropriate evaluation task for our system is to compute the relevance and correctness of returned data.

4.1 Entity Type Evaluation

For this evaluation approach, we collected 80 news articles

[3] `http://www.alchemyapi.com/`, AlchemyAPI, Aug. 2014
[4] `http://www.zemanta.com/`, Zemanta, Aug. 2014
[5] `http://www.opencalais.com/`, OpenCalais, Aug. 2014
[6] `http://wiki.dbpedia.org/`, About DBpedia, Aug. 2015
[7] `https://jena.apache.org/`, Apache Jena, June 2015
[8] `esper.codehaus.org`, EsperTech, Aug. 2014
[9] `http://esper.codehaus.org/esper-4.6.0/doc/reference/en-US/html/technology_overview.html`, Chapter 1. Technology Overview, Aug. 2014
[10] The news articles are processed in parallel.

Algorithm 1: Knowledge-Based Complex Event Extraction from News Stream

```
Data: newsFeeder, userQuery
Result: ListQueue NewsURLs
initialize news feeder and initialize CEP engine;
while (newsFeeder.hasNews()) do
    NewsURL url= NewsFeed.getLastNewsArticle();
    String newsItem = NewsFeed.getNewsText(url);
    vector<String> namedEntities = NamedEntityRecognizer.getNamedEntities(newsItem);
    vector< vector<Resources> > enrichedENs = KnowledgeBaseAnnotator.annotateEntities(namedEntities);
    vector<SimpleEvent> newsEvents = convertToSimpleEvents(enrichedENs) ;
    for event ∈ newsEvents do
        cepEngine.sendEvent(event) ;
        if cepEngine.match(event) then
            NewsURLs.add(event.NewsURL);
        end
    end
end
```

from each news feed domain (business, sports, technology, entertainment and health[11]) on 31th August 2014. For each category, we evaluated the articles according to their entity types.

The diagram in Figure 2 displays the occurrences of entity types and gives an overview of what is likely to occur within specific news categories.

The x-axis represents the different news categories: business, sports, technology, entertainment and health. The y-axis displays the occurrence of each type in percent. The legend in the upper right hand corner assigns colors to the different DBpedia entity types. These types are represented by ontology classes[12]. We compare the occurrence of the most commonly appearing types (Organization, Person, Place, TopicalConcept and Work) in the different news categories.

Organization: simple bands or groups, companies, military units, political parties, sports leagues and much more

Person: this class is separated into many specializations. All kinds of job titles and sport titles, for instance, architect, handball player or coach, are persons. Politicians, organization members and royalties and many more are distinguished.

Place: can be natural like lakes and mountains, but also architectural structures, like buildings and arenas. Infrastructure like airports, stations and locks are also considered to be a place.

TopicalConcept: to this class belong all concept, subject and genre classes, for instance, mathematical concepts, movie genres and academic subjects.

Work: consists of everything that has to do with work, for instance, art work, documents, databases, software and books.

[11] We used news feeds from Reuters `http://www.reuters.com/news` and Bloomberg `http://www.bloomberg.com/`

[12] `http://mappings.dbpedia.org/server/ontology/classes/`, Ontology Classes, Aug. 2014

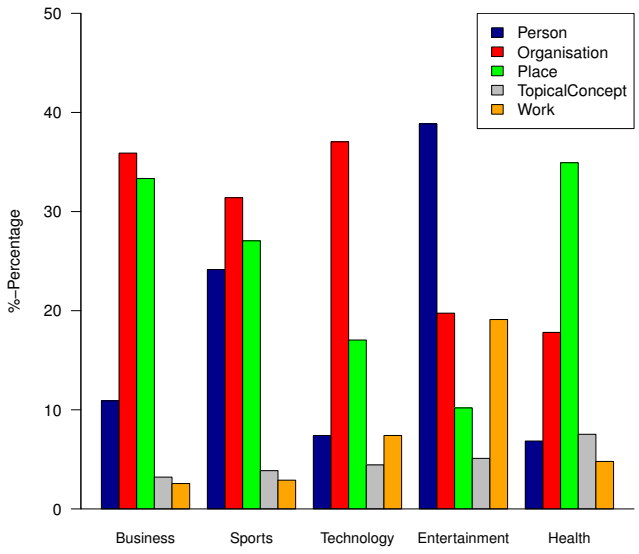

Figure 2: Comparison of Types in Different News Categories

In Figure 2 it is obvious that the types Person, Place and Organization depict the highest occurrence of types in all categories.

Entertainment news have the highest amount of persons with 39%, the amount of organizations and work depict about 20% each. This might be because they handle many topics like celebrity gossip, movies or best restaurant reviews.

Business news on the other hand are primarily about organizations with a percentage of 36% and places with 33%. Possible topics are companies, politics and countries in general.

The amount of organizations, places and persons in sports articles is high, but balanced. These articles may be about

athletics, sports clubs, sport events and games.

Technology and health news articles are about organizations with 37% and 18% that, for instance, publish new devices and technology or medical standards. These news categories include technical and biological concepts, which explain the amount of topical concepts. Health news also depict a high percentage of places.

Certainly, these values may change over time, since news topics are dynamic and dependent on outer influences. The results are not universal valid, but still, provide a rating based on the evaluation day.

4.2 Precision and Recall Evaluation

For our Precision and Recall Evaluation, we proceeded to use a set of 263 news articles in english language[13] from news feeds in real-time received within one week (12th May 2015 - 18th May 2015). From these, we extract 30 random news articles as a base for this evaluation. With manual data investigation relating enriched information, the detection of connections between news articles helps to define meaningful patterns. We evaluate the data alerted by our system, compare them with our expectations from the analysis and calculate relevance scores with precision, recall and f-measure.

The formulas for precision and recall are adapted to our context:

$$precision = \frac{|\text{relevant News} \bigcap \text{retrieved news}|}{|\text{relevant news}|} = \frac{|\text{Correctly Detected News}|}{|\text{relevant news}|}$$

$$recall = \frac{|\text{relevant News} \bigcap \text{retrieved news}|}{|\text{retrieved news}|} = \frac{|\text{Correctly Detected News}|}{|\text{retrieved news}|}$$

We use these metrics to compute the precision and recall value as relevance index for the system evaluation.

For the following evaluation, we provide a natural language query for a general understanding, and the query in concrete Esper Pattern Language as input for our system.

Evaluation of Single News Articles

User Query 1: "Return all articles about the Eurozone."

```
every a = NewsEvent (value('entity_ _ _ _Eurozone')=true)
```

We retrieve three news articles for user query 1 whose content is about the eurozone and thus includes the entity `http://dbpedia.org/page/Eurozone`. One news article features the keyword "eurozone" but was not detected by our system because the Named Entity Recognizer did not extract eurozone as an entity.

The evaluation of single news articles is a basic feature of our system, so we concentrate on the recognition of two news articles in the following because it is of greater value for our users.

Evaluation of Two News Articles

User Query 2 "Return all two articles, where the first one includes Petroleum, and is followed by an article that is about Petroleum and the Industrial Revolution."

```
every a = NewsEvent (
value('entity_ _ _ _Petroleum')=true)
->
every b = NewsEvent (
value('entity_ _ _ _Petroleum')=true,
value('http://purl.org/dc/terms/subject_ _ _ _
http://dbpedia.org/resource/Category:
    Industrial_Revolution')=true)
```

We retrieve 10 news articles that were correctly detected. In general, one news article comprises the entities Petroleum (`http://dbpedia.org/page/Petroleum`) and Industrial Revolution (`http://dbpedia.org/page/Industrial_Revolution`), and five news articles contain Petroleum merely. Three news articles include they keyword "oil" but were not recognized as entities by the Named Entity Recognizer. From sequence observing patterns, we obtain notifications that include both articles in sequence. Since three news articles were not recognized, we miss six news articles in total because the one containing Petroleum and Industrial Revolution was not alerted in this context as a combination either.

In this example query we see that articles are always alerted in combination with each other. Therefore the news article containing Petroleum merely is alerted multiple times but in context of a different complex event. Due to the Complex Event Processing engine the news articles connected via sequence operator are stored locally until a match occurs (2).

User Query 3 "Return all two articles, where the first one includes Greece and the Federal Reserve System as entities, and is followed by an article that is about Greece and a politician."

```
every a = NewsEvent (
value('entity_ _ _ _Greece')=true,
value('entity_ _ _ _Federal Reserve System')=true)
->
every b = NewsEvent (
value('entity_ _ _ _Greece')=true,
value('http://www.w3.org/1999/02/22-rdf-syntax-ns#type_ _
_ _
http://dbpedia.org/ontology/OfficeHolder')=true)
```

For this pattern, our system recognizes all news articles correctly. The data for `http://dbpedia.org/page/Greece`, `http://dbpedia.org/page/Federal_Reserve_System` and `http://dbpedia.org/page/Alexis_Tsipras` is extracted and matched for the query.

User Query 4 "Return all two articles, where the first one is about Germany and Economic Growth, and is followed by an article that is about a country in Europe."

```
every a = NewsEvent (
value('entity_ _ _ _Economic growth')=true,
value('entity_ _ _ _Germany')=true)
->
every b=NewsEvent (
value('http://www.w3.org/1999/02/22-rdf-syntax-ns#type_ _
_ _
http://dbpedia.org/class/yago/EuropeanCountries')=true)
```

The base set features two news articles that are about the entities `http://dbpedia.org/page/Germany` and `http://dbpedia.org/page/Economic_Growth` and four articles about European countries (such as France, Luxembourg, Belgium, Greece and Netherlands). One news article includes the keyword "Poland" which was not recognized as an entity and therefore not recognized by our system as European

[13] The detection of different languages is dependent of the text analysis functionality of the Named Entity Recognizer. AlchemyAPI is capable of identifying more than 95 languages `http://www.alchemyapi.com/api/language-detection`, Language Detection, Aug. 2015.

country. Since this article was not detected, four news articles are missing (the two about Economic Growth and Germany in combination with this article).

User Query 5 "Return all two articles, where the first one is about General Motors, and is followed by an article that is about a a product that General Motors owns."

```
every a = NewsEvent (value(
'entity_ _ _ _General Motors')=true)
->
every b = NewsEvent (
value('http://dbpedia.org/ontology/owner_ _ _ _
http://dbpedia.org/resource/General_Motors')=true)
```

For this pattern, our system recognizes all news articles correctly. The recognized product that General Motors owns is Chevrolet (http://dbpedia.org/page/Chevrolet).

As a summary, Table 1 supplies an overview for the calculated Precision and Recall values. In general, we can state due to the Precision value that all detected news articles are correct in respect to the defined queries. We obtained between 60% and 100% recall in this evaluation. This issues from the quality of the Named Entity Recognition Service that sometimes does not extract an entity even though it is namely occurring in the text.

Still, an average percentage of 100% Precision, 83% Recall and 91% F-Measure is satisfactorily to proof the procedure of our concept for this semantic news extraction system.

5. RELATED WORK

Existing systems use different approaches to address the problem of news overload. They supply approaches that aggregate news data in a user friendly way, provide semantic recommendations or notify users concerning a predefined query. These approaches limit the amount of data reaching a user, who still has to search and validate the data to select the most valuable information for a specific use case.

News Aggregation and Alert Services: Google News[14] is one of the most popular news websites, where computers cluster and rank articles from various information sources from all over the world. Data warehouses store these distributed information to aggregate the data in a fast and flexible way. The view displays articles that have been crawled within the last 30 days. To search for older articles on a particular date range, Google News offers an archive search[15].

Similar articles are clustered and represented in a user-specific, aggregated view to satisfy the individual preferences of a user. It uses the approach of static hierarchical clustering of a collection. Thus, the clustering needs to be frequently recomputed to make sure that users can access the latest breaking news [12].

Using collaborative and content-based filtering techniques, user preferences are learned by user and community data, and the similarities of news content is compared to already rated news items to make recommendations [6]. To determine the relevance of an article, the frequency of the occurrence, number of requests, freshness, location, relevance and diversity[16], are evaluated. Individual adaptations in the settings menu, the search protocol and click history, personalize the appearance of articles. Thus, Google News provides a "personal newspaper" for each user, which adapts to the changing interests of a user over time [10].

Google Alerts[17] is an additional online alert service that monitors Google's search engine for new search results that match the users specified keywords. They are entered as natural language keywords with optional additional parameters for the exclusion of items or limitations to specific sites. Users create personal alerts and are then informed about new web pages, newspaper articles or blogs via e-mail or RSS feeds.

Semantic Processing of News: News@hand [2] is a news platform that uses semantic technologies and content-based approaches for personalized recommendations. Semantic annotation and ontologies structure items into groups while matching this to similarly structured user profiles of preferred items [14]. Concepts that have been involved in the interaction of a user session are collected [4] and enriched with further information from semantic relations. Cantador et al. [3] explain the principle,

"A user interested in natural disasters (superclass of hurricane) is also recommended items about hurricanes. Inversely, a user interested in skiing and snowboarding can be inferred with a certain confidence to be interested in other winter sports. Similarly, a user fascinated about the life of actors can be recommended items in which the name of Brad Pitt appears, due to that person could be an instance of the class actor. Also, a user keen on Spain can be assumed to like Madrid, through the locatedIn transitive relation between these concepts.".

News@hand provides five types of recommendations: driven by a concept-based query, personalized to a single user's profile, oriented to the interests shared by a group of users, combining content-based and collaborative recommendation techniques and considering the current topic context of the session [4]. The textual data of news articles is analyzed and automatically represented using NLP techniques. Nouns are then compared to the corresponding classes and concepts of the domain ontology and the similarity value between those is determined. If this value is above a certain threshold, the semantic concept is added as an annotation of the news item.

5.1 Differences to Our Approach

The existing systems mentioned aim to limit the problem of information overload and offer personalization techniques to make the information seek as comfortable as possible. Still, there are a lot of news articles recommended or alerted to users that they need to look through. The problem with Google News and Google Alerts is the limitation to textual search and keyword-based features. Synonyms or related terms of a concept can not be retrieved which leads to limited valuable information retrieval and a lack in information depth, because interesting and important information may be hidden.

Google Alerts offers user notifications, if something matches the defined query. Bansal et al. [1] mention the main problems with alert services: The crawler raises an alert whenever a new document is encountered, but this does not imply that this document is interesting for the user. Moreover, the number of alerts, if relevant to the user or not, could be large to handle, if the query is not precise enough for a given use case.

[14] https://news.google.com/, Google News, Aug. 2014
[15] http://news.google.com/newspapers, Archive Search, Aug. 2014
[16] https://support.google.com/, About Google News, Aug. 2014
[17] https://www.google.com/alerts, Google Alerts, Aug. 2014

Queries	Relevant News	Retrieved News	Correctly Detected News	Precision Value	Recall Value	F-Measure
Query 1	4	3	3	1	0.75	0.86
Query 2	16	10	10	1	0.62	0.77
Query 3	4	4	4	1	1	1
Query 4	22	18	18	1	0.81	0.9
Query 5	10	10	10	1	1	1
Average	11.2	9	9	1	0.83	0.91

Table 1: Precision and Recall Metrics Overview

Systems	Semantics	Stream-Based Processing	Data Storage	User Profiles	Interest Query	Notification
Google News		X	X	X		
News@hand	X	X	X	X		
Google Alerts			X	X	X	X
Complex Event Extraction	X	X			X	X

Table 2: Comparison of News Data Processing and Notification Systems

News@hand derives knowledge from user preferences and news articles on the other hand, but does not provide an alert service for matches. Furthermore, users still have to browse through returned articles.

Thus, the process of evaluating the recommended and retrieved data can not be automated in any of those systems.

Our system uses inference and coherent resolution techniques which find items within an article that are not explicitly mentioned, because knowledge is derived from the context of words within the sentences. Abstract and intuitive queries with semantic input types are defined and a semantic representation for news articles is provided. In the annotation process the articles are enriched with information from an ontology. The Complex Event Processing engine is able to manage semantic enriched news articles as incoming events and send notifications to the user.

The important scenario is that inspiration and recommendations about news articles is not the urgent necessity of our target audience. They want specific information and need it right away when it is available. Table 2 compares our system to existing ones.

6. CONCLUSION AND DISCUSSION

The main contribution of this paper is an approach for a knowledge-based news notification system in real-time. We showed the provision of users with personalized news articles by retrieving named entities from news articles and enriching them using a knowledge base. Articles are transformed into news events and a complex event processor matches the defined user query to the inspected enriched data. With this basic implementation we ensure the detection of news articles with event patterns.

We retrieve meaningful results matching an explicit pattern, assuming we use a sophisticated knowledge-base for our proof-of-concept implementation. A complete whole world ontology does not exist and some information will always be missing[18] which implies that the completeness of the detected news from our system is dependent of the richness of the used ontology. We assume that in business use cases such knowledge bases have to be prepared (with reasonable efforts) about the business domain. Furthermore, the quality of machine learning approaches is essential for the recognition of all relevant entities and concepts.

With Esper as an event processing engine, we are able to extract complex events over multiple news articles. Pattern operators help define user queries that are matched to real-time news articles from streams. This simplifies the news consumption in cases of searching for relevant articles and recognizing specific connections between different news articles.

Even though we process the data directly on the data stream, the process duration is dependent on response times of endpoints and data volume, and we have no possibility to intervene in these steps. Still, we can state that our system returns news articles in real time[19]. For further limitation and recommendation of returned news the system could be extended with some ranking functionality.

Our system returns meaningful results for the detection of news articles on the basis of a user query. An arbitrary number of news articles with arbitrary amounts of attributes and entities can be regarded over time using EPL. Still, the definition and expressiveness of such user queries can be improved distinctly. The following points are some important future work regarding our approach:

User Query Syntax: A user of our system should be able to define queries discretely. The definition of a news event exhibits a concrete syntax: the separation string "_ _ _ _" between attribute and value pair and the common EPL syntax is not visceral for a regular user. This can be improved by designing an own pattern language that is intuitive to learn and expand.

User Query GUI: The definition of user queries can be simplified by providing a graphical user interface with shortened attributes and entities for the user. People can understand the meaning of an absolute attribute description more intuitive than from an URL link. Furthermore, the existence of syntax errors is limited since the user does not enter any information and simply builds its queries with the help of the GUI.

[18]E.g., Barack Obama has the type OfficeHolder, but not PresidentsOfTheUnitedStates.

[19]based on out definition of real-time in Section 2

Discrimination of Attributes and Entities: It can be of great value to connect entities to specific attributes in a user query. Multiple entities have to be strictly separated, so that an attribute belongs to one precise entity. For example when a user is interested in news articles that exhibit both, a football player and an athlete, our system would also alert on articles that contain one single person that is both, football player and athlete. To make the intended approach work, we would first of all need a meaningful identifier to decide whether a strict discrimination is desirable (valueDiscr in the following example). Moreover, we need an intelligent storage procedure, because each entity needs to be connected to its information separately, so that we can automatically tell, whether a news article features two separate entities where one is football player and the other one is athlete.

```
''every a=NewsEvent(
valueDiscr('rdf:type_ _ _ _SoccerPlayer')=true,
valueDiscr('rdf:type_ _ _ _Athlete')=true)''
```

Connection of multiple News Articles: The definition of attributes and entities of multiple news articles can be connected to find detailed relations in news articles. At the current state, we are able to query for specific entities, as well as abstract concepts and attributes, but an open problem is the interconnection between multiple news articles. The following abstract query is not suitable for our system.

"Return all two articles where the first one is about a steel organization and is followed by another article about a politician and this steel organization out of the first article."

```
''every a=NewsEvent(
value('rdf:type_ _ _ _Steel_Organization')=true)
-> every b=NewsEvent(
value('rdf:type_ _ _ _Politician')=true,
value('entity_ _ _ _''The steel organization from the
    first article''))
```

At the current state, there is no mechanism to declare references to entities that are implicitly derived, like the organizations from "NewsEvent a" in the above example that exhibit the rdf:type property Steel_Organization. To make this kind of query work, we need a storage possibility to memorize the returned entities that are steel organizations, and automatically access the "steel organization from the first article" for the second part in NewsEvent b.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the "InnoProfile-Transfer Corporate Smart Content" project [20] funded by the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder-Entrepreneurial Regions. The authors would like to thank AlchemyAPI for providing access to their named entity extraction API.

8. REFERENCES

[1] N. Bansal and N. Koudas. Blogscope: A system for online analysis of high volume text streams. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07. VLDB Endowment, 2007.

[2] I. Cantador, A. Bellogín, and P. Castells. News@hand: A semantic web approach to recommending news. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '08. Springer-Verlag, 2008.

[3] I. Cantador, A. Bellogín, and P. Castells. Ontology-based personalised and context-aware recommendations of news items. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08. IEEE Computer Society, 2008.

[4] I. Cantador and P. Castells. Semantic contextualisation in a news recommender system. In *Workshop on Context-Aware Recommender Systems (CARS-2009)*, 2009.

[5] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S.-K. Kim. Composite events for active databases: Semantics, contexts and detection. In *VLDB '94*. Morgan Kaufmann Publishers Inc., 1994.

[6] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *WWW '07*. ACM, 2007.

[7] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.

[8] P. A. Laplante. *Real-Time Systems Design and Analysis: An Engineer's Handbook*. IEEE Press, 1992.

[9] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014.

[10] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10. ACM, 2010.

[11] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., 2001.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[13] A. Passant and P. N. Mendes. sparqlpush: Proactive notification of data updates in rdf stores using pubsubhubbub. In *SFSW*, 2010.

[14] O. Phelan, K. McCarthy, M. Bennett, and B. Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. In *Advances in Information Retrieval*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011.

[15] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In *ISWC'12*. Springer-Verlag, 2012.

[16] H. Weifeng, H. Di, and C. Juan. An osgi based rfid complex event processing system. In *EUC*. IEEE, 2010.

[20] Work performed while Kia Teymourian was affiliated with Freie Universität Berlin