AAAI Technical Report WS-12-02
Real-Time Analysis and Mining of Social Streams

Using Complex Event Processing for Semantic Requests in Real-Time Social Media Monitoring

Dominik Riemer, Ljiljana Stojanovic, Nenad Stojanovic
FZI Research Center for Information Technologies
Haid und Neu Str. 10 14
76131 Karlsruhe, Germany
{riemer, stojanovic, nstojano}@fzi.de

Abstract

Social media analytics has been attracting considerable attention in both research and industry due to the increasing popularity of social media usage. As a subset, social media monitoring describes the process of continuous monitoring of a subject matter in social media. From our point of view, the key requirements for such systems are i) high throughput and real time processing of incoming data, ii) a user friendly way to define complex situations of interests that make use of formalized background knowledge and iii) capabilities to perform actions based on gained insights instead of a pure monitoring system. In this paper, we propose a system for (pro) active, real time social media monitoring. Firstly, we describe the conceptual architecture of our system and necessary pre processing steps. Secondly, we introduce our concept of semantic requests that is capable to extend event pattern definitions with background knowledge. Finally, we show the usefulness of this system in two different domains: Real time political opinion tracking and proactive establishment of relationships with consumers in order to perform a new form of real time marketing. The main advantage of our approach is a simplified, expressive way to formulate event patterns in social media applications.

Introduction

The emergent use of social media by a steadily increasing number of people from a wider range of age groups also leads to new business opportunities for companies. In research, the field of social media analytics has gained increased attention. During the last years, a large number of vendors[1] offer so-called social media monitoring. Social Media Monitoring (SMM) deals with a continuous process of monitoring a specific subject of matter. In many cases, SMM focuses on image reputation, e.g. early identification of unwanted, potentially harmful messages by users to avoid spreading of these information peaces to a larger audience.

But besides image reputation, there are important areas that might be useful for companies or other institutions as well. The possibility of early identification of users facing problems or looking for specific products can enable institutions to proactively notify consumers, even before a customer relationship has been established. Furthermore, through the use of social media sources, market researchers can measure the influence of real-world events (e.g., political tv debates) directly while they happen and react on sensible topics in a proper way much faster than current opinion research methods can deal with.

Current approaches on SMM in industry and practice mainly focus on pure monitoring tasks like detecting negative messages or aggregating over crawled data to find an overall sentiment score for a specific brand. We think that future SMM systems will address many more use cases with a strong aspect of proactive behaviour that overcomes current engagement approaches in case of detected critical situations. Figure 1 shows our vision of future SMM systems: They will be able to anticipate the future behaviour in social media streams and act proactively in order to realize new opportunities or to avoid potential problems. Furthermore, there is a huge potential in predictions based on prior gathered knowledge through social media channels (Asur and Huberman 2010). However, this requires complex pattern matching as well as inclusion of historical knowledge through application of data mining methods and semantic technologies to detect and react on situations of interest.

The strong requirement of real-time processing fits perfectly with the technology of Complex Event

[1] An incomplete list can be found at http://www.socialmedia.biz/2011/01/12/top-20-social-media-monitoring-vendors-for-business/ (last accessed: 12/16/2011)

Processing (CEP). CEP has the task of processing multiple events with the goal of identifying meaningful event patterns. An event pattern is an expression formed by using a set of events and a set of event operators and represents a situation of interest (Chakravarthy and Mishra 1994). CEP systems can easily handle a large number of events (while an event is defined as everything that happens or is contemplated to happen (Luckham 2002)) and detect patterns on continuous event streams. In contrast to a database, where queries are formulated on an already existing dataset, queries (called patterns) in event processing are created at first and then applied on a continuous stream of events. An event pattern is constructed using an event processing language. Although there is currently a large number of different languages, many approaches use a SQL-like syntax extended with operators specific to event processing like sequences (an ordered occurrence of a specified number of events) or time windows (that restrict pattern matching to a temporal constraint). In this paper, we concentrate on SQL-like languages. An overview of different languages can be found at (Etzion and Niblett 2011).

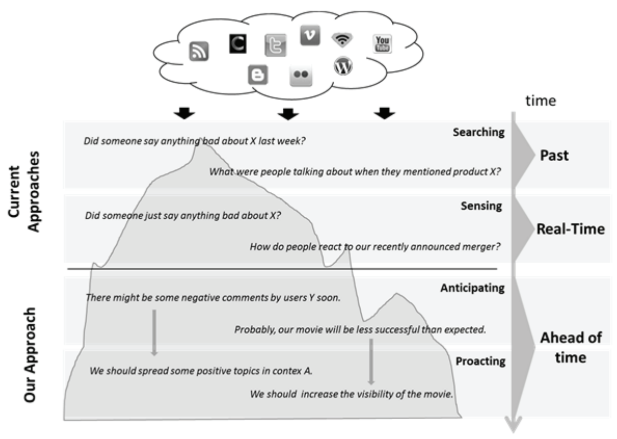

time
Current Approaches
Searching
Did someone say anything bad about X last week?
What were people talking about when they mentioned product X?
Past
Sensing
Did someone just say anything bad about X?
How do people react to our recently announced merger?
Real-Time
Our Approach
Anticipating
There might be some negative comments by users Y soon.
Probably, our movie will be less successful than expected.
Proacting
We should spread some positive topics in contex A.
We should increase the visibility of the movie.
Ahead of time

Figure 1: Potentials of proactive future SMM systems

While streaming data is becoming more important, e.g., through the emergence of the web of things and social media, real-time processing of this data is a challenging task as it does not fit with data preparation models such as indexing used in web search today. Because operations on streaming data itself are time-consuming, improvements of methods for pattern definition prior to runtime are promising. This enables new advantages for design-time pattern generation, as we argue that user-centric pattern generation and deployment is the less time-critical part of event processing. Therefore, a more intelligent pattern definition process that is supported by semantic capabilities can help to get more meaningful insights. Furthermore, the application of CEP on social media streams promises to handle a large number of messages (which is necessary for the use cases mentioned in section 2) and to detect patterns over a flow of events rather than simple filtering.

One of the current research problems in complex event processing is about adding semantic capabilities to pattern definition, event formats and processing. Most existing approaches concentrate on stream reasoning to detect patterns that rely on formalized background knowledge during runtime (Della Valle et. al. 2009). While this is a good solution for knowledge bases that change frequently, the trade-off between processing costs due to inference methods and real-time processing might be too high in cases of rather static background knowledge. However, most event processing engines today do not provide semantic capabilities at all.

In this paper, we introduce a new concept of semantic requests that are embedded into existing event processing languages. Semantic requests are declarative expressions that allow users to express their interest on a more abstract level. In detail, we describe semantic requests for design-time modeling that are translated into standard event processing patterns before deployment in the engine. Main advantage of our approach is the definition of event patterns on an abstract level which simplifies user-centric pattern generation and allows usage of semantic requests based on previously specified domain knowledge in event processing engines.

This paper is structured as follows: Section 2 describes several use cases that stress the potential of semantic requests in real-time social media monitoring. Section 3 presents the conceptual architecture of the SMM system we have developed. In section 4, we focus on the conceptual model for semantic requests in event pattern languages and present a solution of the design-time translation model. Section 5 gives an overview about specific tools for visual pattern definition. Afterwards, we describe our validation approach and present related work in this area.

Use Cases

The above mentioned advantages of combining event processing and semantics in order to get better results than today available social media monitoring platforms can be shown in two examples from both the marketing and the politics domain.

In companies, the establishment and binding of customer relationships plays an important role. The social web leverages companies to directly establish relationships not only with customers but with consumers which leads to new business opportunities. Taking the example of a company with many coffee shop stores, one might be interested in sending coupons or offers to consumers that are inside a maximum radius around a store and, e.g., talking about their wish of a break or drinking a coffee

through a social media channel. A declarative, manual description of such a pattern would be very long and hardly readable with an increasing number of places relevant for that company. Additionally, the maintenance of such patterns can be a problem as there is no division between technical representation and domain knowledge. This example clearly shows the benefit of semantic background knowledge and the business opportunities through proactive behaviour of SMM systems.
Furthermore, intelligent monitoring of social media can detect problems customers are facing even before they are actively contacting a company. This scenario can be extended to observation of customers of competitive products to acquire new customers.
Facing the challenge of real-time tracking of reactions on public events like tv debates in politics, early identification of topics voters might be most interested during the debate can lead to strategic advantages. Today, the important task to highlight the own partys' positive topic and the competiting partys' negative ones is called spinning. From a market research perspective, the real-time measurement of voters' reactions in social media channels, aggregated by candidates, parties, topics or regions seems to be very useful for providing best-suited actions on monitored situations.

Objectives

As stated before, we identified three main objectives of SMM systems:

- High throughput and real-time processing. This objective also includes the ability to integrate multiple source types, such as detecting interdependencies between sources and data enrichment capabilities for value-added filtering. In the case of tv debates, there is not only a need for fast results in order to accelerate decision making process, those events are also followed by a large audience, thus high throughput can be expected.
- User-friendly pattern definition and usage of formalized background knowledge, which focuses on providing a way to define abstract patterns by domain experts that are translated to a machine-readable processing language at design-time. This will enable:
- Capabilities to perform actions based on insights. The action part enables the system to automatically perform actions; a detailed description is not part of this paper.

Overall Architecture

As follows, we give an overview about the system architecture. As seen in Figure 2, we shortly describe implemented components and provide a brief explanation on features and challenges we faced during the development.

System Overview.
The system is comprised of 8 components: Input Adapters for integration of multiple real-time sources, Crawling for processing of non real-time sources, Pre-Processing to enrich messages with additional information (e.g., geospatial enrichment) using an extensible pool of modules, an event processing engine that is responsible for event-driven pattern matching, a persistency component to enable usage of historical events and provision of background knowledge, pattern translation and frontend applications.

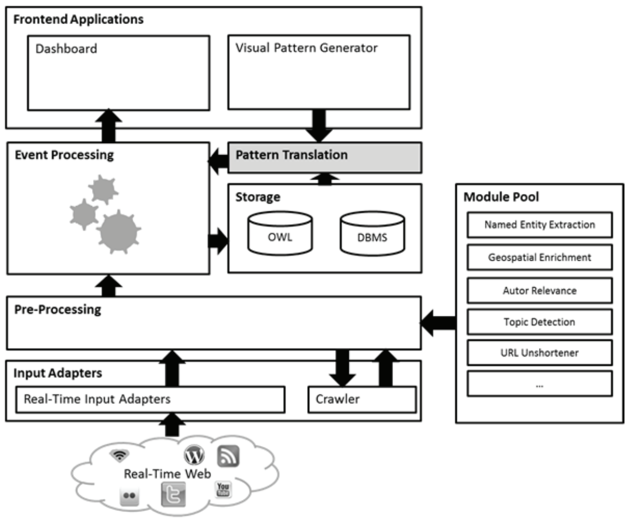


Figure 2: Conceptual Architecture of the SMM System

Input Adapters.
As we focus on real-time processing of social media messages, input adapters have been especially developed for real-time data sources. One of the main requirements is support of different source types. Our prototype implements several real-time content providers such as the Twitter Streaming API and Superfeedr. A crawling module crawls incoming hyperlinks from microblog services and allows considering shared real-time links in social networks as well.
Currently, four source types are supported that are detected through content, whitelists and HTTP header information. Source types were chosen based on different characteristics of messages.

- Updates. Updates are short status messages created by users. They have a maximum length (usually 140 characters) and are often tagged with geospatial information.

- Stories. Our definition describes stories as web pages with textual information. Examples for stories are news pages or blog posts.
- Photos. This source type includes photos without textual information and communities that allow users exchanging photos (e.g., Flickr). In this case, we also consider tags and descriptions linked with a photo.
- Videos. In our case, videos are defined in the same way as the photo type in relation to moving images.

Pre-Processing.
During the pre-processing phase, incoming messages are enriched with extracted additional information like named entities. Modules are divided into the two types functional modules and enrichement modules. Functional Modules serve to provide functionality needed for better content analysis, e.g. boilerplate detection as well as link extraction and unshortening. Enrichment modules are loosely coupled software modules (partially integrated as external webservices). Currently, our system integrates modules for geospatial enrichment, named entity recognition, sentiment analysis, topic detection, language detection and computation of reputation. For every new message, a number of events (e.g. specifying the content, geographical origin or author information) is generated in dependency of the identified source type.

Event Processing.
This component performs processing on events generated by the pre-processing component. Due to the large number of expected messages, our event processing network is comprised of two independent event processing agents (EPA) that can be physically distributed. On the first level, we filter incoming messages with low relevance based on low user rank, user and keyword blacklists and pages without a size of content. The second EPA performs matching of user-defined patterns against the incoming stream and notifies registered event consumers.

Frontend Applications.
The Visual Pattern Generator is a user-faced frontend that allows users to generate personalized patterns of interest. Patterns can be defined on different dimensions, e.g. relevance, location and context. Real-time results of are shown in a Dashboard. A more detailed description will be given in section 5.

Semantic Pattern Translation.
As mentioned in the introduction, we aim to provide users a way of a more abstract, semantic definition of patterns that are automatically translated into machine-readable rules during deployment. Pattern translation deals as a mediating system between user-defined situations of interests that include semantic requests and the used event processing language. This is the main contribution of this paper and will be explained in section 4.

Storage.
Storage enables persistency of events and detected patterns on event streams. This is the basis for pattern detection based on historical knowledge. Additionally, the storage component holds the definition of formalized domain knowledge and provides API access to the ontology.

Semantic Requests

In this section, we present our concept of a model that extends widely adopted event processing languages (EPL) with semantic requests. Semantic requests allow the definition of concepts in an EPL by making use of background knowledge. After presenting the basic model that consists of pure design-time semantic requests and runtime unusuality detection, we focus on so-called design-time relevancy semantic matching. Afterwards, the transformation process is shown in more detail and is finally explained by a description of the implementation that includes applied examples.

Conceptual Model

Event patterns are one of the most important assets in event processing systems (Sen and Stojanovic 2010). However, in the specific field of social media monitoring with should support decision making processes, there is a need for semantic representation of domains of interests in event patterns. Most work currently focuses on stream reasoning by applying background knowledge during runtime event processing. As shown before, real-time social media monitoring can be compared with searching on infinite web streams. In web search, web data is usually crawled and then indexed for enabling efficient search on the data corpus. The efficiency of this corpus can be highly improved by applying powerful but time-consuming optimizations on storage and indexing. From a user perspective, web search is done to fulfill an information need immediately. For that reason, there is less time to improve queries and most effort is put into data optimization. The problems appearing in stream search are completely different. Here, queries are formulated at first and then matched against streaming data. While manipulations on incoming data are time-consuming, the largest potential for improvement is in intelligent processing of queries at design-time in order to get better stream search results. Our comparison of stream search and web search is illustrated in Figure 3.
Therefore, we argue that in some cases (e.g., quite static domain knowledge that only changes at a low frequency), integration of formalized knowledge makes more sense during design-time of event patterns. Our model extends event operators with semantic requests that translate event patterns at design-time with background knowledge-related information. Figure 4 introduces the model of semantic requests on 3 stages: Relevance, Importance and

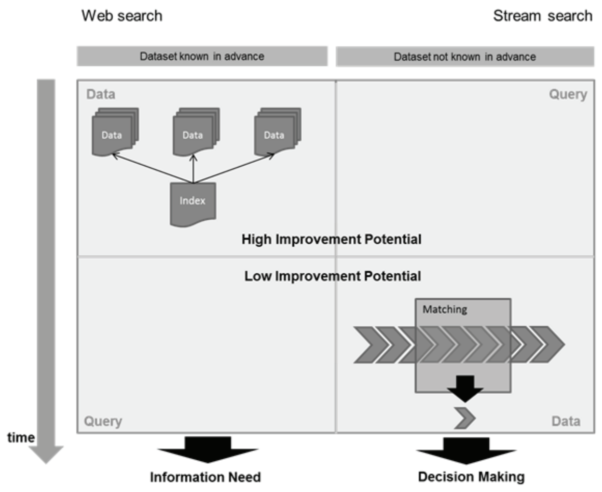


Figure 3: Challenges of Web Search and Stream Search

Unusuality. As stated there, both relevancy and importance semantic requests can be translated at design-time (by extending requests with background knowledge and redeploying them in case of changes of the knowledge base), while unusuality detection happens during runtime (because unusuality has a strong requirement for highest up-to-dateness). The first stage, Relevance, will be the main focus of this paper. Taking the domain of social media monitoring, users might be interested in filtering for specific concepts. For instance, a company might be interested in a pattern that looks for blog posts that compares own products with a similar products of another company. Currently, human pattern definition is a declarative process and would, taking this example, result in a complex query which is hardly readable and maintainable.

Obviously, integration of background knowledge would lead to a more abstract view on the pattern.

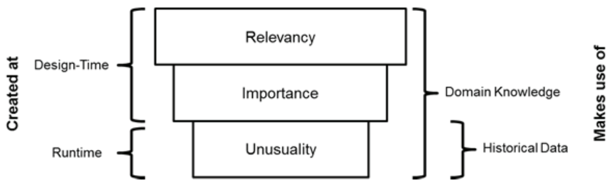


Figure 4: Conceptual Model of Semantic Requests

The second level restricts extension of patterns based on the importance of messages. The large amount of messages that must be expected for some topics requires filtering of unimportant information. As an example, we want to look for important financial tweets that are about automotive manufactures. While filtering for automotive companies can be done by relevancy semantic requests, important tweets containing financial information requires knowledge about characteristic terms in that context. For that reason, relevancy-based semantic requests at this level are extended with frequent terms in these concepts based on domain knowledge. This can be done by calculating the weight of relationships between entities (that have been gathered during runtime) and only considering entities which weights exceed a certain threshold in order to refine patterns (Fang and Guo 2011). In respect to the dynamic flow of information in real-time SMM, where messages might become important within a very short period in time (e.g., as often seen in disaster reporting), this requires a highly scalable computation of the current importance of a message. Note that patterns can be redeployed in case of recently detected important relationships.

Finally, the third stage of our conceptual model concentrates on the mining of historical knowledge gathered from previously matched patterns. Based on the partial fulfillment degree of current patterns and the probability of complete fulfillment in history, unusual messages can be detected. Unusuality from our point of view is based on two features, unexpectness and importance (Sen et. al. 2011). In this paper, we focus on the first stage, introducing semantic requests at design-time to allow definition of pattern restrictions in an abstracted form that are enriched at deployment time with domain knowledge.

Modeling Design-Time Semantic Requests

We now show our approach of modeling semantic matching requests (the presented first stage) at design-time. Generally, definition of patterns can be done in two ways: Either a completely visual pattern design environment, e.g., a web application, that translates created patterns to the internal event processing language, or manual definition of patterns in that language.

Our model presumes the availability of modeled domain knowledge. According to the presented marketing use case, we designed a simplified ontology with the main concepts *Company, Department, Product, Competitor, Place, Person* and *Topic* and relevant relations between these classes. The ontology is the basis for creation of semantic requests in event patterns.

Taking the example of a marketing-related SMM task, the visual pattern designer offers the functionality to select a branch. Based upon this information, users can browse through the knowledge base and select related concepts like products within this branch. Furthermore, products can be restricted on properties, e.g., products that are available in a specific country. The selected concepts and restrictions are then extended with the instances found in the ontology. On the manual pattern definition level, the following initial set of semantic requests is currently supported that follow standard event processing operators. Examples are based on the event processing language used by the engine Esper[2].

[2] http://esper.codehaus.org

sr:anything(property, operator, concept, [restrictions]). Defines a pattern that matches one of the individuals belonging to the specified concepts under given restrictions.

As an example, the statement

```
select * from TwitterEvent where
sr:anything(content, like, Company,
[hasBranch:TechnologyBranch])
```

is translated to the event pattern:

```
select * from TwitterEvent where
content like '%Microsoft%' OR content
like '%Google%' OR content like
'%Oracle%'.
```

sr:anythingRelated(property, operator, concept, [restrictions])
This pattern extension includes not only individuals of a concept, but also any related concepts (e.g., products or people working in a company).
Example:

```
select * from TwitterEvent where
sr:anythingRelated(content, like,
Company,
[hasName:Apple])
```

is translated to:

```
select * from TwitterEvent where
(content like '%Apple%' OR content like
'%iPhone%' OR content like '%iPad%' OR
content like '%Tim Cook%')
```

sr:all(property, operator, concept, [restrictions]).
Similar to the sr:anything request, but uses an AND operator to look for messages that include all matching individuals.

sr:allRelated(property, operator, concept, [restrictions]).
Similar to the sr:anythingRelated operator, but uses an AND operator to look for messages that include all matching individuals.

sr:near(property, udf, concept, restriction, geoRelation, radius).
This pattern enables geospatial functions in semantic requests and returns locations inside a specified radius around places, e.g. related to a company. The following pattern filters for tweets that are geotagged within a certain place related to a company.

```
select * from TwitterEvent as te where
sr:near(te.geo,
de.fzi.udf.Geo.isNearby, Company,
[hasName:Starbucks], hasPlace, 1)
```

Translation:

```
select * from TwitterEvent as te where
(de.fzi.udf.Geo.isNearby(te.geo, -
15.3453, 70.2343, 1) OR
de.fzi.udf.Geo.isNearby(te.geo, 20.232,
30.585, 1) OR
de.fzi.udf.Geo.isNearby(te.geo, -78.54,
23.34, 1))
```

These operators can be used to formulate more complex event queries. For instance,the following pattern looks for 10 StoryEvents in an hour that have a negative sentiment and include Google-related terms:

```
select * from pattern[every [10]
StoryEvent((sr:anythingRelated(content,
like, Company, [hasName:Google]) AND
sentiment < -5)] where
timer:within(1 hour)]
```

We are currently working on extending the set of semantic requests in order to support more complex definitions.

Modeling Design-Time Semantic Requests

Built upon the previously described extension of event patterns with semantic requests, we now show the translation process that takes semantically enriched patterns as an input and produces standard EPL patterns that can be understood by existing event processing engines. Figure 5 shows the tasks during transformation.

- *Visual Pattern Design.* This tool provides a graphical, web-based user interface to define patterns of interests with respect to background knowledge. As patterns can also be created using standard sql-based syntax, this process step remains optional to the user.
- *Semantic Request Extraction.* The next component extracts semantic requests from the query given as an input and passes extracted requests to the SPARQL query generation module.
- *SPARQL Query Generation.* Dependent on selected semantic requests, we construct SPARQL queries of a given semantic request and pass them to the ontology API. This component outputs matching individuals of the defined background knowledge.
- *EPL Generation.* Results from semantic requests are embedded to an extended EPL that does not depend on semantic requests.

- *Deployment.* Finally, the generated and extended pattern is deployed in the event processing engine.

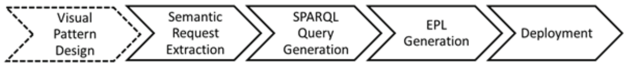


Figure 5: Translation Process

Tools

Our implementation provides the prototype of a visual pattern designer and a dashboard that shows relevant messages based on defined patterns. The currently used Visual Pattern Designer allows selection of event properties based on several dimensions, specific properties of a message. Implemented dimensions are namely context (which allows for definition of semantic requests), source, content, geo, author and relevance. Figure 6 shows the different views of the Visual Pattern Generator. Note that we consider relationships over independent sources, so that users are able to define a pattern that looks for stories related to a specific hashtag as well. Some of the capabilities shown in the figure (like user reputation) are out of scope of this paper.
The Dashboard is an experimental web application that displays incoming messages which matched registered patterns. Users can explore messages by different views: The stream view shows a continuous stream of current results and a detailed view shows all existing properties for a given message including relationships to other messages. Additionally, we provide simple interfaces for displaying pattern statistics and geographically enriched results. Figure 7 clarifies the general structure of our dashboard.

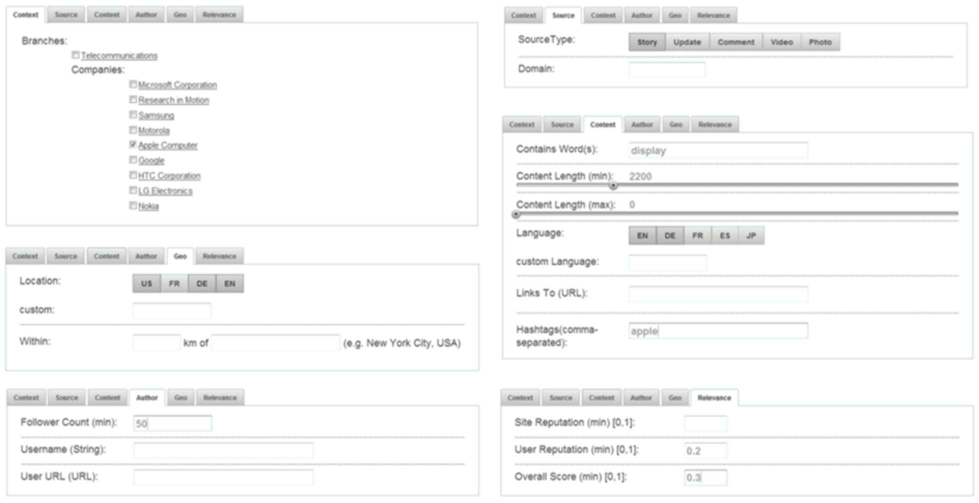
Validation

The validation of this approach is ongoing. We are evaluating the advantages of our system in two levels: Firstly, we focus on performance issues like processing capabilities of a large message volume (which is a crucial point as some of the given use cases often generate a high level of attention in the social web). Secondly, we intend to demonstrate the usefulness and correctness of our model of semantic requests. Hereby, we intend stating that the approach simplifies the process of manual pattern generation in terms of better maintainability and higher expressivity. For that reason, we perform user studies.
Both objectives are evaluated using a real-world dataset. For that reason, we gathered a set of tweets and other media sources collected in the run up to the elections in the German state Berlin that took place in September 2011 and which is currently being assessed. A political ontology contains domain-specific knowledge about corresponding parties, people and topics that have gained attention during these elections.

The goal of the evaluation is to assess the usefulness by comparing the efficency of the request generation in case of having the system and not having. Efficiency will be measured by the time users are saving by declaring a semantic request inside of a pattern. Furthermore, we will show the correctness of the system by checking the statement quality in terms of completeness of generated patterns according to the modeled background knowledge.

Related Work

There is a number of tools available for processing information from the social web. For instance, methods and architectures that aim to derive marketing insights through user-generated content have been presented by (Chen 2010). The authors introduce a system called MI2



Branches:
☐ Telecommunications
Companies:
☐ Microsoft Corporation
☐ Research in Motion
☐ Samsung
☐ Motorola
☑ Apple Computer
☐ Google
☐ HTC Corporation
☐ LG Electronics
☐ Nokia

Location: US FR DE EN
custom:
Within: km of (e.g. New York City, USA)

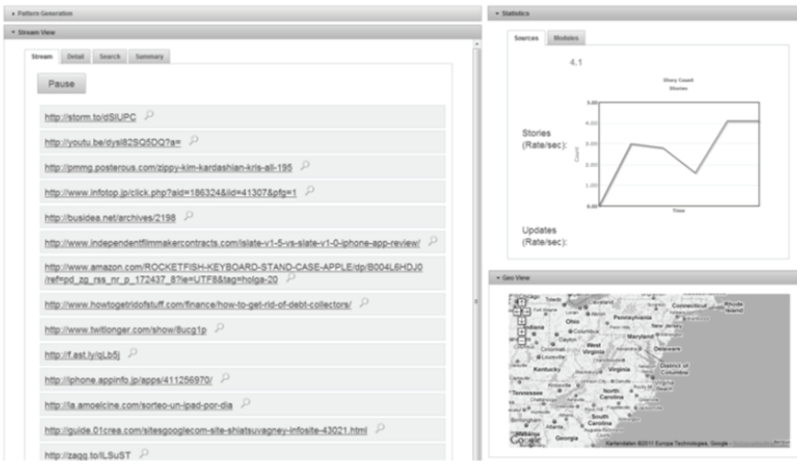
Follower Count (min): 50
Username (String):
User URL (URL):

SourceType: Story Update Comment Video Photo
Domain:

Contains Word(s): display
Content Length (min): 2200
Content Length (max): 0
Language: EN DE FR ES JP
custom Language:
Links To (URL):
Hashtags(comma-separated): apple

Site Reputation (min) [0,1]:
User Reputation (min) [0,1]: 0.2
Overall Score (min) [0,1]: 0.3

Figure 6: Selection Capabilities of the Visual Pattern Generator



```
http://storm.to/dSIUPC
http://youtu.be/dysl82SQ5DQ?a=
http://pmmg.posterous.com/zippy-kim-kardashian-kris-all-195
http://www.infotop.jp/click.php?aid=186324&iid=41307&pfg=1
http://busidea.net/archives/2198
http://www.independentfilmmakercontracts.com/islate-v1-5-vs-slate-v1-0-iphone-app-review/
http://www.amazon.com/ROCKETFISH-KEYBOARD-STAND-CASE-APPLE/dp/B004L6HDJ0
/ref=pd_zg_rss_nr_p_172437_8?ie=UTF8&tag=holga-20
http://www.howtogetridofstuff.com/finance/how-to-get-rid-of-debt-collectors/
http://www.twitlonger.com/show/8ucg1p
http://f.ast.ly/qLb5j
http://iphone.appinfo.jp/apps/411256970/
http://la.amoelcine.com/sorteo-un-ipad-por-dia
http://guide.01crea.com/sitesgooglecom-site-shiatsuvagney-infosite-43021.html
http://zaqq.to/lLSuST
```

Figure 7: Experimental Dashboard

(Marketing Intelligence 2.0) that collects data from message boards as well as public news websites and aims to analyse company-related developments. The system uses external components to perform several tasks like topic extraction and sentiment analysis on previously crawled data. Schirru et.al. present an architecture of a system for domain-specific topic and trend detection in the blogosphere (Schirru et. al. 2010). This work concentrates on clustering and weighting terms around specific topics by using data mining techniques. However, most work currently does not focus on real-time processing of social media from a technical as well as a use case-driven point of view.

Recently, some work in this area moves to the concept of real-time processing of messages. Mendes et. al present Twarql, a system that translates messages from the Twitter stream as Linked Open Data and allows SPARQL queries on this stream (Mendes et. al. 2010). The approach (presented in a brand tracking scenario) does not provide advanced event processing functionalities like searching for sequences of events or aggregation over event properties.

The concept of extending event processing languages with semantic background knowledge has been considered in (Teymourian et. al. 2011). This approach presents Event Query Pre-Processing (EQPP) as a method to rewrite queries based upon background knowledge. It is a general model of rewriting event queries at design-time, but is not specifically grounded in the social media domain and does not regard special operators for usage in this domain.

Finally, there has been some work on extending SQL queries, that are similar to sql-based event processing language, with ontological knowledge. Das et.al. present a system for supporting ontology-based semantic matching in RDBMS (Das et. al. 2004). The authors introduce several SQL operators that perform semantic matching and an indexing scheme that supports fast ontology matching operations.

Conclusions and Future Work

In this paper, we introduced a novel concept of integrating semantic requests into event processing languages. Our approach simplifies the generation of event patterns that make use of background knowledge. The conceptual model defines semantic requests on three levels, namely relevance, importance and unusuality. Taking the example of social media monitoring in the marketing domain, we defined several semantic requests on the relevancy level that are translated to a pure event processing language at design time. Additionally, we provided a prototypical implementation of our approach and an evaluation plan in a political domain.

Future work includes the extension of our model with design-time semantic requests for importance filtering and development of a solution for unusuality definition and detection at runtime.

Acknowledgements

This work is partly funded by the EU ICT projects ALERT and ReFLEX.

References

Asur, S.; and Huberman, B.A. 2010. Predicting the Future with Social Media. In: *Proceedings of the 2010 International Conference on Web Intelligence and Intelligent Agent Technology*. Toronto, Canada.

Chakravarthy, S.; and Mishra, D. 1994. Snoop: An expressive event specification language for active databases. *Data Knowledge Engineering* 14(1): 1 26.

Chen, H. 2010. Business and Marketing Intelligence 2.0, Part 2. *IEEE Intelligent Systems* 25.

Das, S.; Chong, E.; Eadon, G.; and Srinivasan, J. 2004. Supporting Ontology based Semantic Matching in RDBMS. In: *Proceedings of the 30th VLDB Conference*. Toronto, Canada.

Della Valle, E; Ceri, S.; van Harmelen, F; and Fensel, D. 2009. It's a streaming world! Reasoning upon rapidly changing information. *IEEE Intelligent Systems, 24(6).*

Etzion, O.; and Niblett, P. 2011. *Event Processing in Action*: Manning.

Fang, J.; and Guo, L. 2011. Calculation of Weight of Entities in Ontologies by Using Usage Information. *In: Proceedings of the 3rd International Workshop on Intelligent Systems and Application.* Wuhan, China.

Luckham, D. 2002. The power of events: *An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Reading, Mass.: Addison Wesley.

Mendes, P.; Passant, A.; and Kapanipathi, P. 2010. Twarql: Tapping Into the Wisdom of the Crowd. In: *Proceedings of the 6th International Conference on Semantic Systems*. Graz, Austria.

Schirru; R.; Obradovic, D.; Baumann, S.; and Wortmann, P. 2010. Domain Specific Identification of Topics and Trends in the Blogosphere. *Advances in Data Mining. Applications and Theoretical Aspects. Springer (2010).* 6171: 490 504.

Sen, S.; Stojanovic, N.; Shemrani, B.F. 2011. EchoPAT: a system for real time complex event pattern monitoring. In: *Proceedings of the 5th ACM International Conference on Distributed Event Based Systems.*New York, USA.

Sen, S.; Stojanovic, N.. 2010. GRUVe: A Methodology for Complex Event Pattern Life Cycle Management. *In: Proceedings of the 22nd international conference on Advanced Information Systems Engineering.* Hammamet, Tunisia.

Teymourian, K.; Rohde, M.; Hasan, A.; Paschka, A. 2011. Fusion of Event Data Stream and Background Knowledge for Semantic Enabled CEP. In: *Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web*. Bonn, Germany.