conTEXT – A Mashup platform for Lightweight Text Analytics

Ali Khalili[1], Sören Auer[2], and Axel-Cyrille Ngonga Ngomo[1]

[1] AKSW, Institute of Computer Science, University of Leipzig
{khalili,ngonga}@informatik.uni-leipzig.de
[2] Institute of Computer Science, University of Bonn and Fraunhofer IAIS
auer@cs.uni-bonn.de

Abstract. Social media technologies such as Weblogs, Microblogging, Wikis and Social Networks have become one of the most important parts of our daily life as they enable us to communicate and share stories with a lot of people. The more the amount of published information grows, the more important are solutions for accessing, analyzing, summarizing and visualizing information. While substantial progress has been made in the last years in each of these areas individually, we argue, that only the intelligent combination of approaches will make this progress truly useful and leverage further synergies between techniques. conTEXT aims to provide a user-friendly and lightweight Mashup platform enabling end-users to use sophisticated NLP techniques for analyzing and visualizing their content. it provides a flexible text analytics architecture of participation by innovative combination of different pieces of services for content collection and analysis. Named Entity Recognition (e.g. DBpedia Spotlight, FOX), Relation Extraction (e.g. BOA), Sentiment Analysis (e.g. Vivekn), Social Media (e.g. Twitter, Fcebook, Google+, LinkedIn), and Visualization (e.g. Exhibit, D3js) are some of the example services and APIs currently utilized in conTEXT.

1 Introduction

Currently, there seems to be an imbalance on the Web. Hundreds of millions of users continuously share stories about their life on social networking platforms such as *Facebook*, *Twitter* and *Google Plus*. However, the conclusions which can be drawn from analysing the shared content are rarely shared back with the users of these platforms. The social networking platforms on the other hand exploit the results of analysing user-generated content for targeted placement of advertisements, promotions, customer studies etc. One basic principle of data privacy is, that every person should be able to know what personal information is stored about herself in a database (cf. OECD privacy principles[1]). We argue, that this principle does *not* suffice anymore and that there is an *analytical information imbalance*. People should be able to find out what patterns can be discovered and what conclusions can be drawn from the information they share.

[1] http://oecdprivacy.org/#participation

We showcase *conTEXT* – a text analytics Mashup, which helps to mitigate the analytical information imbalance by allowing end-users to use sophisticated NLP techniques for analysing and visualizing their content, be it a weblog, Twitter feed, website or article collection. The architecture of conTEXT comprises different services for content access, content analysis (currently mainly *Named Entity Recognition*, *Relation Extraction* and *Sentiment Analysis*) and visualization. Different exchangeable components can be plugged into this architecture. Users are empowered to provide manual corrections and feedback on the automatic text processing results, which directly increase the semantic annotation quality and are used as input for attaining further automatic improvements. An online demo of the conTEXT is available at `http://context.aksw.org`.

conTEXT empowers users to answer a number of questions, which were previously impossible or very tedious to answer. Examples include:

- Finding all articles or posts related to a specific person, location or organization.
- Identifying the most frequently mentioned terms, concepts, people, locations or organizations in a corpus.
- Showing the temporal relations between people or events mentioned in the corpus.
- Discovering typical relationships between entities.
- Identifying trending concepts or entities over time.
- Find posts where certain entities or concepts co-occur.

conTEXT lowers the barrier to text analytics by providing the following key features:

- No installation and configuration required.
- Access content from a variety of sources.
- Instantly show the results of text analysis to users in a variety of visualizations.
- Allow refinement of automatic annotations and take feedback into account.
- Provide a generic architecture where different services and APIs for content acquisition, natural language processing and visualization can be plugged together.

It is worth mentioning that we have published the idea and technical details of conTEXT plus a comprehensive evaluation of the platform at [3].

2 conTEXT Mashup Architecture and Workflow

Figure 1 shows the Mashup architecture together with the process of text analytics in conTEXT. The process starts by collecting information from the web or social web. conTEXT utilizes standard information access methods and protocols such as RSS/ATOM feeds, SPARQL endpoints and REST APIs as well as customized crawlers for SlideWiki, WordPress, Blogger and Twitter to build a corpus of information relevant for a certain user.

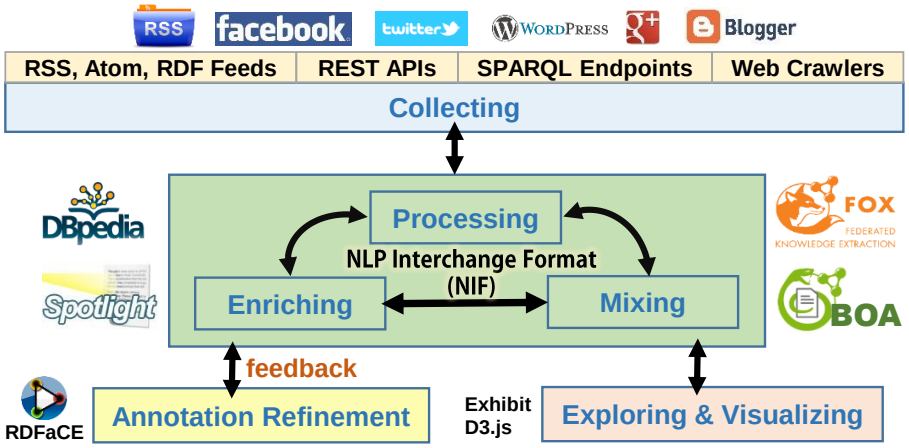

Fig. 1. Text analytics workflow in conTEXT.

The assembled text corpus is then processed by NLP services. While conTEXT can integrate virtually any NLP services, it currently implements interfaces for *DBpedia Spotlight* [4] and the *Federated knOwledge eXtraction Framework* (FOX) [6] for discovering and annotating named entities in the text. DBpedia Spotlight annotates mentions of *DBpedia* resources in text thereby links unstructured information sources to the Linked Open Data cloud through DBpedia. FOX is a knowledge extraction framework that utilizes a variety of different NLP algorithms to extract RDF triples of high accuracy from text. Unlike DBpedia Spotlight, which supports all the DBpedia resource types, FOX is limited to `Person`, `Location` and `Organization` types. On the other hand, since FOX uses ensemble learning to merge different NLP algorithms, leads to a higher precision and recall (see [6] for details).

The processed corpus is then further enriched by three mechanisms:

- DBpedia URIs of the found entities are de-referenced in order to add more specific information to the discovered named entities (e.g. longitude and latitudes for locations, birth and death dates for people etc.).
- Entity co-occurrences are matched with pre-defined natural-language patterns for DBpedia predicates provided by *BOA* (BOotstrapping linked datA)[2] in order to extract possible relationships between the entities.
- The sentiment of articles is analyzed by the help of *Vivekn*[5] which is an open source sentiment analysis service.

The processed data can also be joined with other existing corpora in a *text analytics mashup*. Such a mashup of different annotated corpora combines information from more than one corpus in order to provide users an integrated view. Analytics mashups help to provide more context for the text corpus under analysis and also enable users to mix diverse text corpora for performing a com-

[2] `http://boa.aksw.org`

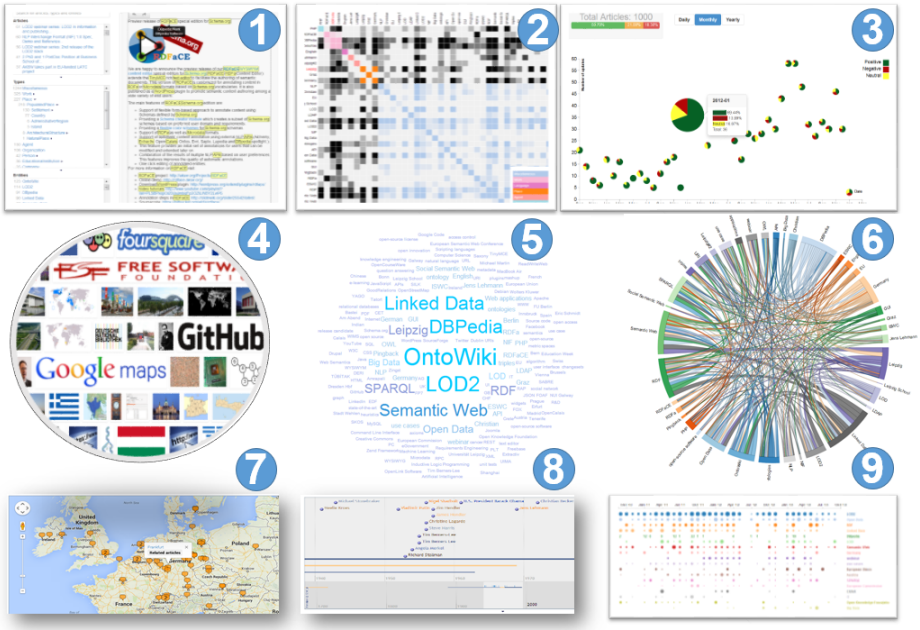

Fig. 2. Different views for exploration and visualization of an analysed corpus: 1) faceted browser, 2) matrix view, 3) sentiment view, 4) image view, 5) tag cloud, 6) chordal graph view, 7) map view, 8) timeline, 9) trend view.

parative analysis. For example, a user's Wordpress blog corpus can be integrated with corpora obtained from her Twitter and Facebook accounts. The creation of analytics mashups requires dealing with the heterogeneity of different corpora as well as the heterogeneity of different NLP services utilized for annotation. conTEXT employs *NIF* (NLP Interchange Format)[3][1] to deal with this heterogeneity. The use of NIF allows us to quickly integrate additional NLP services into conTEXT.

The processed, enriched and possibly mixed results are presented to users using different views for exploration and visualization of the data. *Exhibit*[4] (structured data publishing) and *D3.js*[5] (data-driven documents) are employed for realizing a dynamic exploration and visualization experience. Additionally, conTEXT provides an authoring user interface based on the *RDFa Content Editor* (RDFaCE)[6] to enable users to revise the annotated results. User-refined annotations are sent back to the NLP services as feedback for the purpose of learning in the system.

[3] `http://persistence.uni-leipzig.org/nlp2rdf/`
[4] `http://simile-widgets.org/exhibit3/`
[5] `http://d3js.org/`
[6] `http://rdface.aksw.org`

2.1 Exploration and visualization interfaces

The dynamic exploration of content indexed by the annotated entities facilitates faster and easier comprehension of the content and provide new insights. conTEXT creates a novel entity-based search and browsing interface for end-users to review and explore their content. On the other hand, conTEXT provides different visualization interfaces which present, transform, and convert semantically enriched data into a visual representation, so that, users can explore and query the data efficiently. Visualization UIs are supported by noise-removal algorithms which will tune the results for better representation and will highlight the picks and trends in the visualizations. For example, we use a frequency threshold when displaying single resources in interfaces. In addition, a threshold based on the Dice similarity is used in interfaces which display co-occurrences. By these means, we ensure that the information overload is reduced and that information shown to the user is the most relevant. Note that the user can chose to deactivate or alter any of these thresholds.

conTEXT allows to plugin a variety of different exploration and visualization modules, which operate on the conTEXT data model capturing the annotated corpora. By default, conTEXT provides the following views for exploring and visualizing the annotated corpora (cf. Figure 2):

- *Faceted browsing* allows users to quickly and efficiently explore the corpus along multiple dimensions (i.e. articles, entity types, temporal data) using the DBpedia ontology. The faceted view enables users to drill a large set of articles down to a set adhering to certain constraints.
- *Matrix view* shows the entity co-occurrence matrix. Each cell in the matrix reflects the entity co-occurrence by entity types (color of the cell) and by the frequency of co-occurrence (color intensity).
- *Sentiment view* shows the overall sentiment of the corpus as well as the sentiment of the individual articles in the corpus.
- *Image view* shows a picture collage created from the entities Wikipedia images. This is an alternative for tag cloud which reflects the frequent entities in the corpora by using different image sizes.
- *Tag cloud* shows entities found in the corpus in different sizes depending on their prevalence. The tag cloud helps to quickly identify the most prominent entities in the corpora.
- *Chordal graph view* shows the relationships among the different entities in a corpus. The relationships are extracted based on the co-occurrence of the entities and their matching to a set of predefined natural language patterns.
- *Places map* shows the locations and the corresponding articles in the corpus. This view allows users to quickly identify the spatial distribution of locations refereed to in the corpus.
- *People timeline* shows the temporal relations between people mentioned in the corpus. For that purpose, references to people found in the corpus are enriched with birth and death days found in DBpedia.
- *Trend view* shows the occurrence frequency of entities in the corpus over the times. The trend view requires a corpus with articles having a timestamp

(such as blogposts or tweets).

2.2 Annotation refinement interfaces

A lightweight text analytics as implemented by conTEXT provides direct incentives to users to adopt and revise semantic text annotations. Users will obtain more precise results as they refine annotations. On the other hand, NLP services can benefit from these manually-revised annotations to learn the right annotations. conTEXT employs the RDFa Content Editor RDFaCE within the faceted browsing view and thus enables users to edit existing annotations while browsing the data. The WYSIWYM (What-You-See-Is-What-You-Mean) interface [2] provided by RDFaCE enables integrated visualization and authoring of unstructured and semantic content (i.e. annotations encoded in RDFa). The manual annotations are collected and sent as feedback to the corresponding NLP service. In collaboration with DBpedia Spotlight and FOX team, we created feedback APIs for these services. The user feedback serves two purposes: On one hand, it directly increases the quality of the semantic annotation. On the other hand, it can serve as input for active learning techniques, which can further boost precision and recall of the semantic annotation.

2.3 Linked Data interface for search engine optimization (SEO)

The `Schema.org` initiative provides a collection of shared schemas that Web authors can use to markup their content in order to enable enhanced search and browsing features offered by major search engines. *RDFa*, *Microdata* and *JSON-LD* are currently approved formats to markup web documents based on Schema.org. There are already tools like *Google Structured Data Markup Helper*[7] which help users to generate and embed such markup into their web content. A direct feature of the Linked Data based text analytics with conTEXT is the provisioning of a *SEO* interface. conTEXT encodes the results of the content annotation (automatic and revisions by the user) in the *JSON-LD*[8] format which can be directly exposed to schema.org aware search engines. This component employs the current mapping from the DBpedia ontology to the Schema.org vocabularies[9]. Thus the conTEXT SEO interface enables end-users to benefit from better exposure in search engines (e.g. through Google's *Rich Text Snippets*) with very little effort.

2.4 Real-time semantic analysis

In addition to its normal functionality, conTEXT also supports real-time content analysis for streaming data like Twitter streams. A demo of the real-time semantic analysis for Twitter is available at `http://context.aksw.org/resa`.

[7] `https://www.google.com/webmasters/markup-helper/`
[8] JSON for Linked Data `http://json-ld.org/`
[9] `http://schema.rdfs.org/mappings.html`

This way, users can see the live progress of different analytics views on incoming data and therby can quickly follow the trends which are currently on the social media. Real-time analytics is also useful for the companies and businesses to gain competitive advantage and to improve their customer relationships by monitoring users feedback on social media websites.

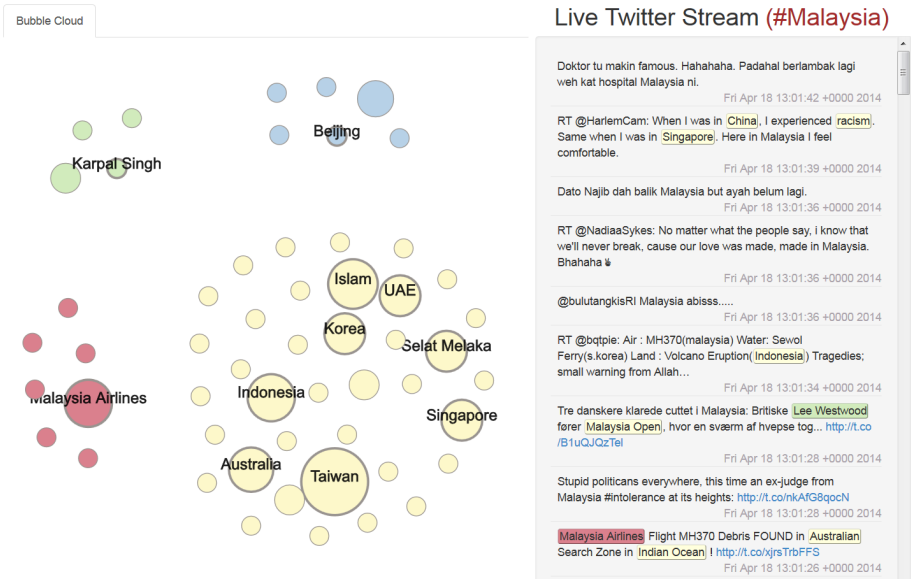

Fig. 3. Example of realtime semantic analysis monitoring #Malaysia

2.5 Implementation

conTEXT is a Web application implemented in *PHP* and *JavaScript* using a relational database backend (MySQL). The application makes extensive use of the model-view-controller (MVC) architecture pattern and relies heavily on *JSON* format as input for the dynamic client-side visualization and exploration functionality.

Figure 4 shows the conTEXT data model, which comprises `Corpus`, `Article`, `Entity` and `Entity_Type` tables to represent and persist the data for text analytics. A corpus is composed of a set of articles or a set of other corpora (in case of a mixed corpus). Each article includes a set of entities represented by URIs and an annotation score. The `Entity_type` table stores the type(s) for each entity. As described in Section 2, conTEXT employs NIF for interoperability between different NLP services as well as different corpora. Code 1.1 shows a sample NIF annotation stored for an article. In order to create the required input data structures for different visualization views supported by D3.js and

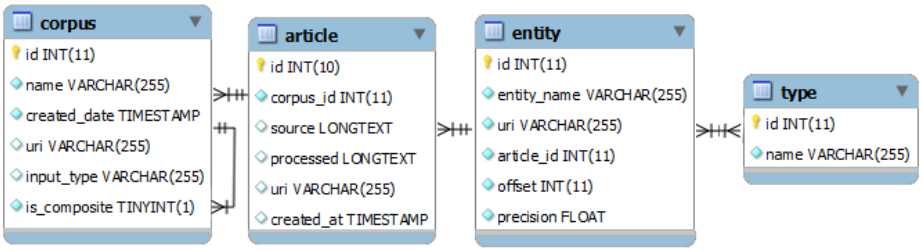

Fig. 4. conTEXT data model.

Exhibit, we implemented a *data transformer* component. This component processes, merges and converts the stored NIF formats into the appropriate input formats for visualization layouts (e.g. D3 Matrix layout or Exhibit Map layout). After the transformation, the converted visualization input representations are cached on the server-side as JSON files to increase the performance of the system in subsequent runs.

One of the main design goals during the development of conTEXT was modularity and extensibility. Consequently, we realized several points of extensibility for implementation. For example, additional visual analysis views can be easily added. Additional NLP APIs and data collectors can be registered. The faceted browser based on Exhibit can be extended in order to synchronize it with other graphical views implemented by D3.js and to improve the scalability of the system. Support for localization and internationalization can be added into the user interface as well as to the data processing components.

Code 1.1. Generated semantic annotations represented in NIF/JSON.

```
{
  "@context": {
    "nif":
        "http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core",
  },
  "@id": "http://blog.aksw.org/2013/dbpedia-swj",
  "resources":[{
    "@id": "http://dbpedia.org/resource/DBpedia",
  "anchorOf": "DBpedia",
  "beginIndex": "1144",
  "endIndex": "1151",
  "confidence": "0.9",
  "type": "DBpedia:Software"
  }, {
    "@id": "http://dbpedia.org/resource/Freebase_(database)",
  "anchorOf": "Freebase",
  "beginIndex": "973",
  "endIndex": "981",
  "confidence": "0.9",
  "type": "DBpedia:Misc"
  }, ... ]
}
```

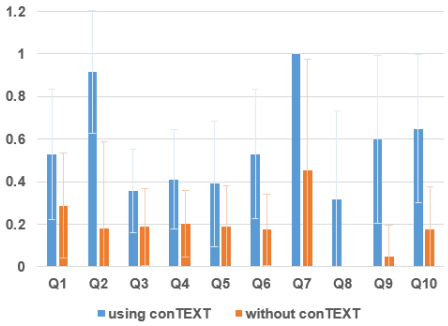

Fig. 5. Avg. Jaccard similarity index for answers using & without the conTEXT.

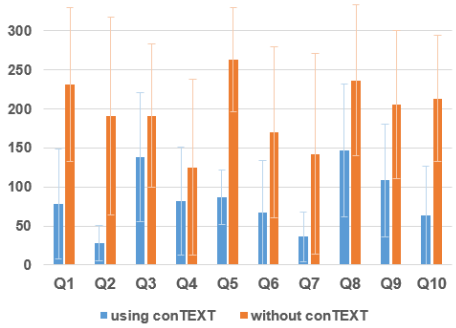

Fig. 6. Avg. time spent (in second) for finding answers using & without the conTEXT.

2.6 Evaluation

In order to evaluate the usefulness and usability of conTEXT, we performed a user study with 25 subjects (20 PhD students having different backgrounds from computer software to life sciences, 2 MSc students and 3 BSc students with good command of English) on a set of 10 questions pertaining to knowledge discovery in corpora of unstructured data. To provide quantitative insights in the usefulness of conTEXT, we carried out a task-driven usefulness study where we measured the improvement in efficiency and effectiveness that results from using conTEXT. The evaluation platform provided users with a short tutorial on how to perform the tasks using conTEXT and how to add their responses for the questions. A look at the effectiveness results (Figure 5) suggests that those users who tried to carry out these task without conTEXT failed as they achieve an average Jaccard score of 0.17 on this particular task while users relying on conTEXT achieve 0.65. Moreover, as shown in Figure 6, in all cases, the users are more time-efficient when using conTEXT than without the tool.

To assess the usability of conTEXT, we used the standardized, ten-item Likert scale-based *System Usability Scale* (SUS) questionnaire and asked each person who partook in our usefulness evaluation to partake in the usability evaluation. The results of our study (cf. Figure 7) showed a mean usability score of **82** indicating a high level of usability according to the SUS score.

3 Conclusion and Future Work

With conTEXT, we showcased an innovative text analytics Mashup platform for end-users, which integrates a number of previously disconnected technologies. In this way, conTEXT is making NLP technologies more accessible, so they can be easily and beneficially used by arbitrary end-users. conTEXT provides instant benefits for annotation and empowers users to gain novel insights and complete tasks, which previously required substantial development.

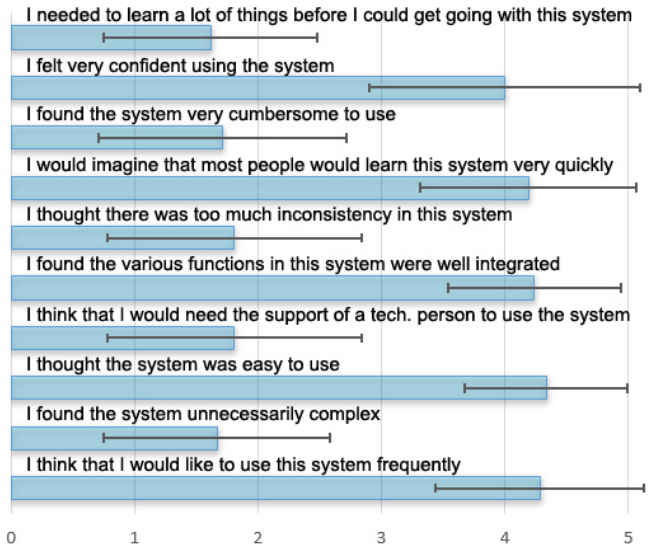

Fig. 7. Result of usability evaluation using SUS questionnaire.

In future, we plan to investigate, how user feedback can be used across different corpora. We consider the harnessing of user feedback by NLP services an area with great potential to attain further boosts in annotation quality. We plan to integrate revisioning functionality, where users can manipulate complete sets of semantic annotations instead of just individual ones. In that regard, we envision that conTEXT can assume a similar position for text corpora as have data cleansing tools such as OpenRefine for structure data.

References

1. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating nlp using linked data. In: 12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia (2013)
2. Khalili, A., Auer, S.: Wysiwym authoring of structured content based on schema.org. In: WISE 2013. pp. 425–438. Springer Berlin Heidelberg (2013)
3. Khalili, A., Auer, S., Ngomo, A.C.N.: context – lightweight text analytics using linked data. In: 11th Extended Semantic Web Conference (ESWC2014) (2014), `http://svn.aksw.org/papers/2014/ESWC_conTEXT/public.pdf`
4. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. I-Semantics '11, ACM, New York, USA (2011)
5. Narayanan, V., Arora, I., Bhatia, A.: Fast and accurate sentiment classification using an enhanced naive bayes model. CoRR abs/1305.6143 (2013)
6. Ngomo, A.C.N., Heino, N., Lyko, K., Speck, R., Kaltenböck, M.: Scms - semantifying content management systems. In: ISWC. pp. 189–204 (2011)