

conTEXT – Lightweight Text Analytics Using Linked Data

Ali Khalili¹, Sören Auer², and Axel-Cyrille Ngonga Ngomo¹

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

Abstract. The Web democratized publishing – everybody can easily publish information on a Website, Blog, in social networks or microblogging systems. The more the amount of published information grows, the more important are technologies for accessing, analysing, summarising and visualising 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. In this paper we develop a text analytics architecture of participation, which allows ordinary people to use sophisticated NLP techniques for analysing and visualizing their content, be it a Blog, Twitter feed, Website or article collection. The architecture comprises interfaces for information access, natural language processing and visualization. Different exchangeable components can be plugged into this architecture, making it easy to tailor for individual needs. We evaluate the usefulness of our approach by comparing both the effectiveness and efficiency of end users within a task-solving setting. Moreover, we evaluate the usability of our approach using a questionnaire-driven approach. Both evaluations suggest that ordinary Web users are empowered to analyse their data and perform tasks, which were previously out of reach.

Keywords: `:#eswc2014Khalili`.

1 Introduction

The Web democratized publishing – everybody can easily publish information on a website, blog, in social networks or microblogging systems. The more the amount of published information grows, the more important are technologies for accessing, analysing, summarising 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. Natural Language Processing (NLP) technologies, for example, were developed for text analysis, but are often cumbersome and difficult to use for ordinary people and it is even

more difficult to make sense of the results produced by these tools. Information visualization techniques, such as data-driven documents [3], on the other hand can provide intuitive visualizations of complex relationships.

We showcase *conTEXT*¹ – a text analytics architecture of participation, which allows 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 comprises interfaces for information access, natural language processing (currently mainly Named Entity Recognition) 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>.

Motivation. 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 principles²). 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.

Let us look at the case of a typical social network user Judy. When Judy updates her social networking page regularly over years, she should be able to discover what the main topics were she shared with her friends, what places, products or organizations are related to her posts and how these things she wrote about are interrelated. Currently, the social network Judy uses analyses her and other users data in a big data warehouse. Advertisement customers of the social networking platform, can place targeted adds to users being interested in certain topics. Judy, for example, is sneaker aficionado. She likes to wear colorful sports shoes with interesting designs, follows the latest trends and regularly shares her current favorites with her friends on the social network. Increasingly, advertisements for sportswear are placed within her posts. Being able to understand what conclusions can be drawn by analysing her posts will give Judy at least some of the power back into her hands she lost during the last years to Web giants analysing big user data.

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

¹ We choose the name conTEXT, since our approach performs analyzes *with* (Latin ‘con’) text and provides contextual visualizations for discovered entities in text.

² <http://oecdprivacy.org/#participation>

- 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.

The text analytics architecture and implementation we present in this article helps to mitigate the analytical information imbalance. With almost no effort, users can analyse the information they share and obtain similar insights as social networking sites.

Approach. 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 analysis to users in a variety of visualizations.
- Allow refinement of automatic annotations and take feedback into account.
- Provide a generic architecture where different modules for content acquisition, natural language processing and visualization can be plugged together.

Semantic Web and Linked Data is used in conTEXT in particular in the following ways:

- The linked-data aware *Natural Language Interchange format* (NIF) is used for integrating various NLP tools.
- The FOX and Spotlight Linked Data based disambiguation ensures that we work with real-world entities instead of surface forms.
- Linked Data background knowledge is used to enrich the result of the analysis and provide upper-level ontological knowledge for facilitating the exploration.
- Semantic annotations are encoded in RDFa and can be re-integrated back into the original data sources.

The article is structured as follows: We show that conTEXT fills a gap in the space of related approaches in Section 2. The general workflow and interface design is presented in Section 3. The different visualizations and views supported by conTEXT are discussed in Section 4. We show the results of a qualitative and quantitative user evaluation in Section 5 and discuss some more general aspects in Section 6 before we conclude in Section 7.

2 Related Work

Analytics (i.e. the discovery and communication of meaningful patterns in data) is a broad area of research and technology. Involving research ranging from *NLP* and *Machine Learning* to *Semantic Web*, this area has been very vibrant in recent years. Related work in the domain of analytics can be roughly categorized according to the following dimensions:

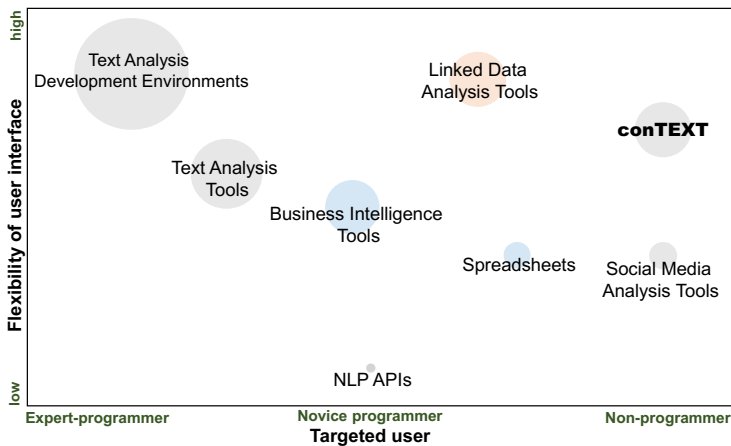


Fig. 1. Flexibility of user interfaces and targeted user groups as well as genericity (circle size) and degree of structure (circle color) for various analytics platforms

- *Degree of structure.* Typically, an analytics system extracts patterns from a certain type of input data. The type of input data can vary between *unstructured* (e.g. text, audio, videos), *semi-structured* (e.g. text formats, shallow XML, CSV) and *structured* data (e.g. databases, RDF, richly structured XML).
- *Flexibility of user interface.* Analytics systems provide different types of interfaces to communicate the found patterns to users. A flexible UI should support techniques for *exploration*, *visualization* as well as even *feedback and refinement* of the discovered patterns. This dimension also evaluates the *interactivity* of UIs, *diversity* of analytical views as well as the capability to *mix* results.
- *Targeted user.* An analytics system might be used by different types of users including *non-programmer*, *novice-programmer* and *expert-programmer*.
- *Genericity.* This dimension assesses an analytics system in terms of *genericity of architecture* and *scalability*. These features enable reuse of components as well as adding new functionality and data at minimal effort.

Figure 1 provides an abstract view of the state-of-the-art in analytics according to these dimensions.

Text analysis development environments usually provide comprehensive support for developing customized text analytics workflows for extracting, transforming and visualizing data. Typically they provide a high degree of genericity and interface flexibility, but require users to be expert-programmers. Examples include the *IBM Content Analytics platform* [1], *GATE* [4], *Apache UIMA* [7].

Text analysis tools provide a higher level of abstraction (thus catering more novice users) at the cost of genericity. Yang et al. [20] recently published an extensive text analytics survey from the viewpoint of the targeted user and

introduced a tool called *WizIE* which enables novice programmers to perform different tasks of text analysis. Examples include *Attensity*³, *Thomson Data Analyzer*⁴ *Trendminer* [19] and *MashMaker* [6].

Business intelligence (BI) tools are applications designed to retrieve, analyse and report mainly highly-structured data for facilitating business decision making. BI tools usually require some form of programming or at least proficiency in query construction and report designing. Examples include *Zoho Reports*⁵, *SAP NetWeaver*⁶, *Jackbe*⁷, and *RapidMiner* [12].

Spreadsheet-based tools are interactive applications for organization and analysis of data in tabular form. They can be used without much programming skills, are relatively generically applicable and provide flexible visualizations. However, spreadsheet-based tools are limited to structured tabular, data and can not be applied to semi-structured or text data. Examples include *Excel*, *DataWrangler* [13], *Google Docs Spreadsheets* and *Google Refine*.

NLP APIs are web services providing natural language processing (e.g. named entity recognition and relation extraction) for analysing web pages and documents. The use of these APIs requires some form of programming and flexible interfaces are usually not provided. Examples include *Alchemy*, *OpenCalais*, *Apache OpenNLP*.⁸

Linked Data analysis tools support the exploration and visualization of Linked Data. Examples include *Facete*⁹ for spatial and *CubeViz*¹⁰ for statistical data. Dadzie and Rowe [5] present a comprehensive survey of approaches for visualising and exploring Linked Data. They conclude that most of the tools are designed only for tech-users and do not provide overviews on the data.

Social Media analysis tools such as *SRSR*, *TweetDeck*¹¹, *Topsy*¹², *Flumes*¹³, and *Trendsmap*¹⁴ focus in comparison to conTEXT primarily on the content aggregation across large repositories (e.g. Twitter as a whole) and perform popularity and trend analysis. conTEXT on the other hand aims at providing different exploration and visualization means for more specific types of content exploiting the extracted semantics.

When comparing these different analytics tool categories according to the dimensions genericity, UI flexibility, target users and degree of structure we discovered a lack of tools dealing with unstructured content, catering non-expert

³ <http://www.attensity.com>

⁴ <http://thomsonreuters.com/thomson-data-analyzer/>

⁵ <http://www.zoho.com/reports/>

⁶ <http://sap.com/netweaver>

⁷ <http://jackbe.com/>

⁸ A complete list of NLP APIs is available at <http://nerd.eurecom.fr/>

⁹ <http://aksw.org/Projects/Facete>

¹⁰ <http://aksw.org/Projects/CubeViz>

¹¹ <http://tweetdeck.com/>

¹² <http://topsy.com/>

¹³ <http://www.flumes.com/>

¹⁴ <http://trendsmap.com/>

users and providing flexible analytics interfaces. The aim of developing the text analytics tool conTEXT is to fill this gap.

3 Workflow and Interface Design

Workflow. Figure 2 shows 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.

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* [17] and the *Federated knOwledge eXtraction Framework* (FOX) [18] 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 [18] for details).

The processed corpus is then further enriched by two 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) [8] in order to extract possible relationships between the entities.

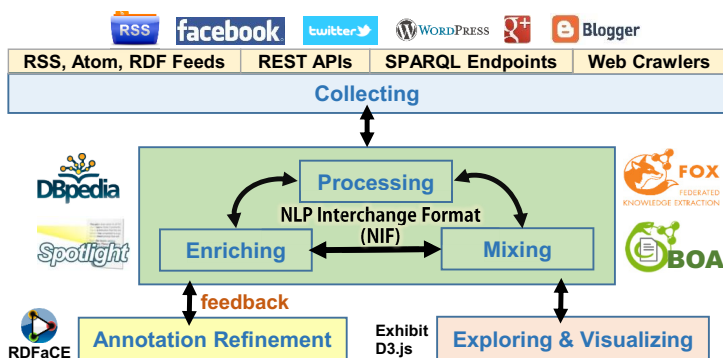


Fig. 2. Text analytics workflow in conTEXT

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 comparative 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) [10] 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* [11]¹⁵ (structured data publishing) and *D3.js* [3]¹⁶ (data-driven documents) are employed for realizing a dynamic exploration and visualization experience. Additionally, conTEXT provides an annotation refinement user interface based on the *RDFa Content Editor* (RDFaCE) [15] 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.

Progressive crawling and annotation. The process of collecting and annotating a large text corpus can be time-consuming. Therefore it is very important to provide users with immediate results and inform them about the progress of the crawling and annotation task. For this purpose, we have designed special user interface elements to keep users informed until the complete results are available. The first indicator interface is an animated progress bar which shows the percentage of the collected/annotated results as well as the currently downloaded and processed item (e.g. the title of the blog post). The second indicator interface is a real-time tag cloud which is updated while the annotation is in progress. We logged all crawling and processing timings during our evaluation period. Based on these records, the processing of a Twitter feed with 300 tweets takes on average 30 seconds and the processing of 100 blog posts approx. 3-4 minutes on standard server with i7 Intel CPU (with parallelization and hardware optimizations further significant acceleration is possible). This shows, that for typical crawling and annotation tasks the conTEXT processing can be performed in almost real-time thus providing instant results to the users.

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

¹⁵ <http://simile-widgets.org/exhibit3/>

¹⁶ <http://d3js.org/>

Table 1. NLP Feedback parameters

Parameter	Description
<i>text</i>	annotated text.
<i>entityUri</i>	the identifier of the annotated entity.
<i>surfaceForm</i>	the name of the annotated entity.
<i>offset</i>	position of the first letter of the entity.
<i>feedback</i>	indicates whether the annotation is correct or incorrect.
<i>context</i>	indicates the context of the annotated corpus.
<i>isManual</i>	indicates whether the feedback is sent by user or by other NLP services.
<i>senderIDs</i>	identifier(s) of the feedback sender.

to edit existing annotations while browsing the data. The WYSIWYM (What-You-See-Is-What-You-Mean) interface [14] 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. The feedback encompasses the parameters specified in Table 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.

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*¹⁷ 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

¹⁷ <https://www.google.com/webmasters/markup-helper/>

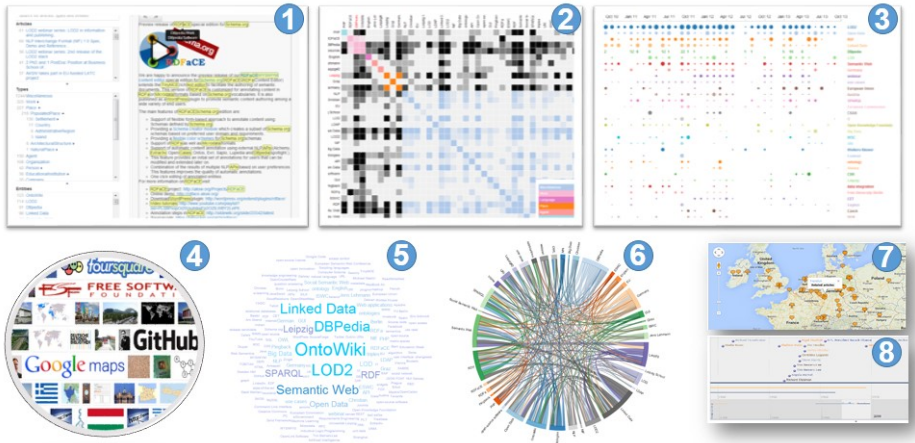


Fig. 3. Different views for exploration and visualization of an analysed corpus: 1) faceted browser, 2) matrix view, 3) trend view, 4) image view, 5) tag cloud, 6) chordal graph view, 7) map view, 8) timeline.

and revisions by the user) in the *JSON-LD*¹⁸ 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 vocabularies¹⁹. 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.

4 Views

A key aspect of conTEXT is to provide intuitive exploration and visualization options for the annotated corpora. For that purpose, 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:

- *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).
- *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).

¹⁸ JSON for Linked Data <http://json-ld.org/>

¹⁹ <http://schema.rdfs.org/mappings.html>

- *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 referred 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.

Modularity and extensibility is not only limited to views in conTEXT. For example, additional NLP APIs and data collection interfaces can be registered as well.

5 Evaluation

The goal of our evaluation was two-fold. First, we wanted to provide quantitative insights in the usefulness of conTEXT. To this end, we carried out a task-driven usefulness study where we measured the improvement in efficiency and effectiveness that results from using conTEXT. Second, we aim to evaluate the usability of our approach.

5.1 Usefulness Study

Experimental Setup. To achieve the first goal of our evaluation, we carried out controlled experiments with **25** users (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. For example, we asked users the following question: “What are the five most mentioned countries by Bill Gates tweets?”. The 10 questions were determined as follows: We collected a set of 61 questions from 12 researchers of the University of Leipzig. These questions were regarded as a corpus and analysed using conTEXT. After removing questions that were quasi-duplicates manually, we chose 10 questions that we subdivided into 2 sets of 5 questions. Each of users involved in the evaluation was then asked to solve one set of questions with conTEXT and the other one without the tool. In all cases, the users were given access to the corpus from which the question was extracted. While answering the questions with conTEXT, the users used the analysis abilities of conTEXT. Else, they were allowed to use all

digital search media of their choice except conTEXT. To ensure that we did not introduce any bias in the results due to distribution of hard questions across the two sets, one half of the users was asked to solve the first set of questions with conTEXT while the others did the same with the second set and vice-versa. We evaluated the users' efficiency by measuring the time that they required to answer the questions. Note that the users were asked to terminate any task that required more than 5 minutes to solve. In addition, we measured the users' effectiveness by comparing the answers of each user to a gold standard which was created manually by the authors. Given that the answers to the questions were sets, we measured the similarity of the answers A provided by the each user and the gold standard G by using the *Jaccard* similarity of the two sets, i.e., $\frac{|A \cap G|}{|A \cup G|}$. The platform²⁰ provided users with a short tutorial on how to perform the tasks using conTEXT and how to add their responses for the questions.

Results. The results of our first series of evaluations are shown in Figures 4 and 5. On average, the users required 136.4% more time without conTEXT than when using the tool. A fine-grained inspection of the results suggests that our approach clearly enables users to perform tasks akin to the ones provided in the evaluation in less time. Especially complex tasks such as “Name a middle-eastern country that has never been spoken of in the AKSW blog” are carried out more than three times faster using conTEXT. In some cases, conTEXT even enables users to carry out tasks that seemed out of reach before. For example, the question “What are the five most mentioned countries by Bill Gates’ tweets?” (Q10) was deemed impossible to answer in reasonable time by using normal search tools by several users. A look at the effectiveness results 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. The overall Jaccard score with conTEXT lies around 0.57, which suggests that the tasks in our evaluation were non-trivial. This is confirmed by the overall score of 0.19 without conTEXT. Interestingly, the average effectiveness results achieve by users with conTEXT are always superior to those achieved without conTEXT, especially on task Q8, where users without conTEXT never found the right answer. Moreover, in all cases, the users are more time-efficient when using conTEXT than without the tool.

5.2 Usability Study

Experimental Setup. The goal of the second part of our evaluation was to assess the usability of conTEXT. To achieve this objective, we used the standardized, ten-item Likert scale-based *System Usability Scale* (SUS) [16] questionnaire and asked each person who partook in our usefulness evaluation to partake in the usability evaluation. The questions were part of a Google questionnaire and can be found at <http://goo.gl/JKzgdK>.

²⁰ available at <http://context.aksw.org/app/evaluation>

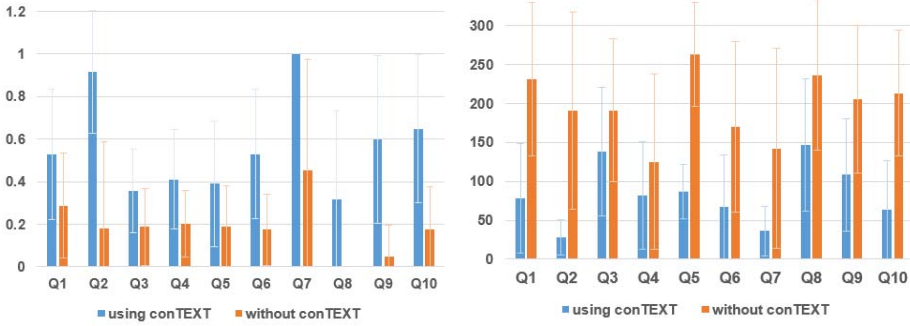


Fig. 4. Avg. Jaccard similarity index for **Fig. 5.** Avg. time spent (in second) for finding answers using & without the conTEXT

Results. The results of our study (cf. Figure 6) showed a mean usability score of **82** indicating a high level of usability according to the SUS score. The responses to question 1 suggests that our system is adequate for frequent use (average score to question 1 = 4.23 ± 0.83) by users all of type (4.29 ± 0.68 average score for question 7). While a small fraction of the functionality is deemed unnecessary by some users (average score of 1.7 ± 0.92 to question 2, 1.88 ± 1.05 to question 6 and 1.76 ± 1.09 to question 8), the users deem the system easy to use (average score of 4.3 ± 0.59 to question 3). Only one user suggested that he/she would need a

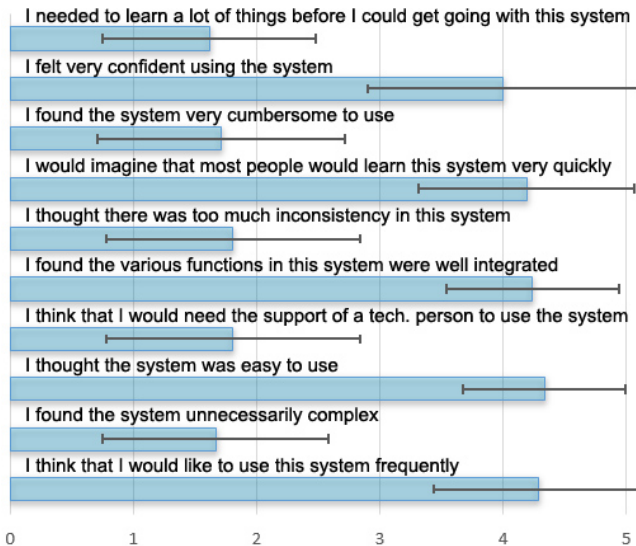


Fig. 6. Result of usability evaluation using SUS questionnaire

technical person to use the system, while all other users were fine without one. The modules of the system in itself were deemed to be well integrated (4.23 ± 0.66 average score to question 5). Overall, the output of the system seems to be easy to understand (4.11 ± 1.05 score to question 9) while users even without training assume themselves capable of using the system (1.52 ± 0.72 to question 10). These results corroborate the results of the first part of our evaluation as they suggest that conTEXT is not only easy to use but provides also useful functionality.

6 Discussion

The incentive for each author to use conTEXT is the ease of analysing unstructured and semi-structured data and the resulting sophisticated user interfaces. While this motivation is personal and the immediately perceptible benefit is local, there are far reaching effects as a result of the semantically annotated information being entirely publicly accessible in structured form. We now discuss how conTEXT, by design, is helping to democratize the use of NLP technology, helps alleviating the Semantic Web's chicken-and-egg problem and harnesses the power of feedback loops.

Democratizing the NLP usage. With conTEXT natural language processing technology is made more accessible, so that sophisticated text analytics can be used with just a few clicks by ordinary users. This was achieved by abstracting from a particular technology (e.g. by using the NIF format) and by supporting a) typical input formats for corpus generation (such as social networking feeds) and b) sophisticated visualizations employing the data-driven document metaphor. As a result, ordinary users can observe the power of natural language processing and semantic technologies with minimal effort. By directly showing the effect of semantic annotations and demonstrating the benefits for improved navigation, exploration and search, users will gain a better understanding of recent technology advances. On the other hand, users will regain control and command of their information, since they are empowered to perform similar analyses as major social and advertising networks do. If users discover using conTEXT, that the patterns of their communication habits do not correspond to what they would like others to observe, they can delete certain information and alter their blogging habits (e.g. publish more post on professional than leisure activities).

In addition to gaining insights into their own communication habits, users can also more easily discover communication habits of people in charge (e.g. politicians) or do quicker fact checking. Answering questions, such as 'What has Angela Merkel said about the Syria conflict?' or 'What are commercial property development areas discussed in the last two years by the city council?' become dramatically easier to answer, even when the information source is a large unstructured text corpus.

Alleviating the Semantic Web's chicken-and-egg problem. Recently we could observe a significant increase of the amount of structured data publishing on

the Web. However, this increase can be attributed primarily to article metadata being made available and already to a much lesser extent to just a few entity types (people, organizations, products) being prevalent [2]. As a consequence, we still face the chicken-and-egg problem to truly realize the vision of a Web, where large parts of the information are available in structured formats and semantically annotated. Before no substantial amount of content is available in semantic representations, search engines will not pick up this information and without better search capabilities publishers are not inclined to make additional effort to provide semantic annotations for their content. The latter is particularly true for unstructured and semi-structured content, which is much more difficult to annotate than structured content from relational databases (where merely some templates have to be adopted in order to provide e.g. RDFa).

conTEXT can help to overcome this problem, since it provides instant benefits to users for creating comprehensive semantic annotations. The result of an annotation with conTEXT can easily be exported, re-integrated or published along the original content. Also, we plan to provide conTEXT as a service, where a user's content is continuously ingested and processed, the user is informed about updates and thus the semantic representations of the content evolve along with the content itself.

Harnessing the power of feedback loops. Thomas Goetz states in his influential WIRED Magazin article [9]: 'Provide people with information about their actions in real time, then give them a chance to change those actions, pushing them toward better behaviors.' With conTEXT, we want to give users direct feedback on what information can be extracted from their works. At the same time we want to incorporate their feedback and revisions of the semantic annotations back in the NLP processing loop. Incorporating user feedback was so far not much in the focus of the NLP community. With conTEXT, we aim to contribute to changing this. We argue, that NLP technology achieving, for example, 90% precision, recall or f-measure, might not fulfill the requirements of a number of potential use cases. When we can increase the quality of the NLP through user feedback, we might be able to substantially extend the range of potential NLP applications. The user feedback here serves two purposes: One the 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.

7 Conclusion

With conTEXT, we showcased an innovative text analytics application 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.

In future, we plan extend work on conTEXT along several directions. We aim 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. On a related angle, 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. Finally, we aim to extend conTEXT with REST interfaces to ease its integration into other applications such as content management systems.

References

1. Solution brief: Ibm content analytics with enterprise search, version 3.0, <http://www-03.ibm.com/software/products/us/en/contentanalyticssearch>
2. Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., Völker, J.: Deployment of rDfA, microdata, and microformats on the web – A quantitative analysis. In: Alani, H., et al. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 17–32. Springer, Heidelberg (2013)
3. Bostock, M., Ogievetsky, V., Heer, J.: D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17(12), 2301–2309 (2011)
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011), <http://tinyurl.com/gatebook>
5. Dadzie, A.S., Rowe, M.: Approaches to visualising linked data. *Semantic Web* 2(2), 89–124 (2011)
6. Ennals, R., Brewer, E.A., Garofalakis, M.N., Shadle, M., Gandhi, P.: Intel mash maker: join the web. *SIGMOD Record* 36(4), 27–33 (2007)
7. Ferrucci, D., Lally, A.: Uima: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.* 10(3-4), 327–348 (2004), <http://dx.doi.org/10.1017/S1351324904003523>
8. Gerber, D., Ngonga Ngomo, A.C.: Bootstrapping the linked data web. In: 1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011 (2011)
9. Goetz, T.: Harnessing the power of feedback loops. *WIRED Magazine* (2011), http://www.wired.com/magazine/2011/06/ff_feedbackloop/
10. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating NLP using linked data. In: Alani, H., et al. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 98–113. Springer, Heidelberg (2013)
11. Huynh, D.F., Karger, D.R., Miller, R.C.: Exhibit: lightweight structured data publishing. In: WWW 2007, pp. 737–746. ACM, New York (2007)
12. Jungermann, F.: Information Extraction with RapidMiner, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.5586>
13. Kandel, S., Paepcke, A., Hellerstein, J., Heer, J.: Wrangler: interactive visual specification of data transformation scripts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011, pp. 3363–3372. ACM (2011)
14. Khalili, A., Auer, S.: Wysiwym authoring of structured content based on schema.org. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (eds.) WISE 2013, Part II. LNCS, vol. 8181, pp. 425–438. Springer, Heidelberg (2013)

15. Khalili, A., Auer, S., Hladky, D.: The rdfa content editor. In: IEEE COMPSAC 2012, pp. 531–540. IEEE Computer Society (2012)
16. Lewis, J., Sauro, J.: The Factor Structure of the System Usability Scale. In: Kurosu, M. (ed.) HCD 2009. LNCS, vol. 5619, pp. 94–103. Springer, Heidelberg (2009)
17. 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, I-Semantics 2011, pp. 1–8. ACM, New York (2011)
18. Ngonga Ngomo, A.-C., Heino, N., Lyko, K., Speck, R., Kaltenböck, M.: SCMS – semantifying content management systems. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part II. LNCS, vol. 7032, pp. 189–204. Springer, Heidelberg (2011)
19. Preotiuc-Pietro, D., Samangooei, S., Cohn, T., Gibbins, N., Niranjana, M.: Trendminer: an architecture for real time analysis of social media text (June 2012)
20. Yang, H., Pupons-Wickham, D., Chiticariu, L., Li, Y., Nguyen, B., Carreno-Fuentes, A.: I can do text analytics!: designing development tools for novice developers. In: CHI 2013, pp. 1599–1608. ACM, New York (2013)