2015 IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2 2015

Social Media Story Telling

Patrick Hennig*, Philipp Berger*, Christian Dullweber†, Moritz Finke†,
Fabian Maschler†, Julian Risch†and Christoph Meinel‡
Hasso-Plattner-Institute
University of Potsdam, Germany
*{patrick.hennig, philipp.berger}@hpi.de
†{christian.dullweber, moritz.finke, fabian.maschler, julian.risch}@student.hpi.uni-potsdam.de
‡office-meinel@hpi.de

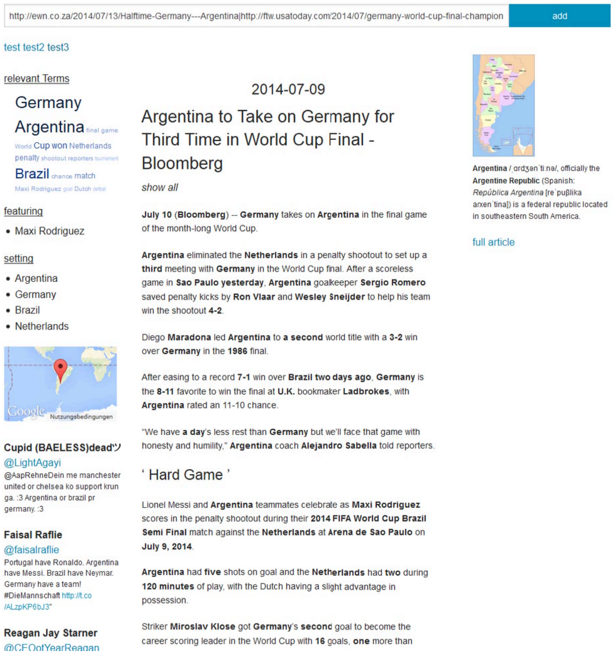
***Abstract*—The number of documents on the web increases rapidly and often there is an enormous information overlap between different sources covering the same topic. Since it is impractical to read through all posts regarding a subject, there is a need for summaries combining the most relevant facts. In this context combining information from different sources in form of stories is an important method to provide perspective, while presenting and enriching the existing content in an interesting, natural and narrative way.**

Today, stories are often not available or they have been elaborately written and selected by journalists. Thus, we present an automated approach to create stories from multiple input documents. Furthermore the developed framework implements strategies to visualize stories and link content to related sources of information, such as images, tweets and encyclopedia records ready to be explored by the reader. Our approach combines deriving a story line from a graph of interlinked sources with a story-centric multi-document summarization.

I. Introduction

The Internet in its current shape is highly driven by user-generated content, for example, blogs. Even as early as 2011 there was already an estimated number of 181 million blogs[1] covering a broad range of topics. The blogosphere as the collective of all blogs is a great source of information consisting not only of text, but also of multimedia content such as pictures, music or videos. Since the authors have no need to be objective (apart from news-blogs), the reader is confronted with an individual and subjective view of the topic. This could be seen as a disadvantage, but there are also many blogs about topics that are entirely subjective by nature, like those covering travel experiences. These Blogs are meaningful as a source of personal information for someone, who identifies with the author and possibly aims to take the same trip. Although most of the blogs are written in a subjective manner, often the provided information is highly relevant precisely because of the unique perspective that is reflected.

In general several documents with overlapping information need to be considered and compared in order to get an balanced overview about the coverage of an issue that possibly stretches over a long period of time and often relates to multiple events. The motivation of this work was derived by approach of analyzing Events and Trends introduced by Hennig et. al [1].



http://ewn.co.za/2014/07/13/Halftime-Germany---Argentina|http://ftw.usatoday.com/2014/07/germany-world-cup-final-champion

add

test test2 test3

relevant Terms

Germany
Argentina final game
World Cup won Netherlands
penalty shootout reporters tournament
Brazil chance match
Maxi Rodriguez goal Dutch defeat

featuring

- Maxi Rodriguez

setting

- Argentina
- Germany
- Brazil
- Netherlands

Cupid (BAELESS)dead'ツ
@LightAgayi
@AapRehneDein me manchester united or chelsea ko support krun ga. :3 Argentina or brazil pr germany. :3

Faisal Raflie
@faisalraflie
Portugal have Ronaldo. Argentina have Messi. Brazil have Neymar. Germany have a team! #DieMannschaft http://t.co/ALzpKP6bJ3"

Reagan Jay Starner
@CFOofYearReagan

2014-07-09

Argentina to Take on Germany for Third Time in World Cup Final - Bloomberg

show all

July 10 (**Bloomberg**) -- **Germany** takes on **Argentina** in the final game of the month-long World Cup.

Argentina eliminated the **Netherlands** in a penalty shootout to set up a **third** meeting with **Germany** in the World Cup final. After a scoreless game in **Sao Paulo yesterday**, **Argentina** goalkeeper **Sergio Romero** saved penalty kicks by **Ron Vlaar** and **Wesley Sneijder** to help his team win the shootout **4-2**.

Diego **Maradona** led **Argentina** to **a second** world title with a **3-2** win over **Germany** in the **1986** final.

After easing to a record **7-1** win over **Brazil two days ago**, **Germany** is the **8-11** favorite to win the final at **U.K.** bookmaker **Ladbrokes**, with **Argentina** rated an 11-10 chance.

"We have **a day**'s less rest than **Germany** but we'll face that game with honesty and humility," **Argentina** coach **Alejandro Sabella** told reporters.

' Hard Game '

Lionel Messi and **Argentina** teammates celebrate as **Maxi Rodriguez** scores in the penalty shootout during their **2014 FIFA World Cup Brazil Semi Final** match against the **Netherlands** at **Arena de Sao Paulo** on **July 9, 2014**.

Argentina had **five** shots on goal and the **Netherlands** had **two** during **120 minutes** of play, with the Dutch having a slight advantage in possession.

Striker **Miroslav Klose** got **Germany**'s **second** goal to become the career scoring leader in the World Cup with **16** goals, **one** more than

Argentina /ˌɑrdʒənˈtiːnə/, officially the **Argentine Republic** (Spanish: *República Argentina* [reˈpuβlika arxenˈtina]) is a federal republic located in southeastern South America.

full article

Figure 1: Visualisation of a story

We propose an automated approach for creating a digital story, based on one or multiple blog posts or similar types of web documents as input. To handle the potentially huge amount of information, we summarize the content of one blog post to focus on important insights and include complementary information from other blogs and sources that might be of interest to the user. This is achieved by deriving a linear ordering from the graph of interlinked posts, extracting the central topics and linking to related sources. This way we combine and enrich summaries of individual documents to create a vivid story that makes use of narrative elements. Figure 1 shows such a story generated from three blog posts about the FIFA world cup soccer final 2014.

In the following, we present our approach to create a digital story based on blog posts. section II is about related techniques that we used to build the software. In **??**, we explain in detail how the information is extracted from the initial post. Further, the gathering of additional data is shown. Based on this, the creation of the story is shown in section V and the developed tool is discussed in section VI with an example story. Finally, we conclude with an outlook on imaginable improvements and

[1] http://www.nielsen.com/us/en/insights/news/2012/buzz-in-the-blogosphere-millions-more-bloggers-and-blog-readers.html at 09.11.2014

978-1-5090-1893-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SmartCity.2015.84

future challenges in section VII.

A. Project Scope

Weblogs offer access to latest information discussed in the real world. Since writing posts in weblogs goes along with a high editorial effort, the available information is of major interest. However, for a user it is becoming harder and harder to gain an overview of all discussions in the blogosphere. It is almost impossible for a user to extract information from the web, especially from the blogosphere. Hence, a system that collects information from the blogosphere and presents it to the user in a very meaningful way would be of great use.

Therefore, mining, analyzing, modeling and presenting this enormous amount of data is the overall aim of the project the presented work is integrated in. This enables the user to detect technical trends, political climates or news articles about a specific topic. Most approaches to mining and analyzing such a huge amount of data focus on offline algorithms which use pre-aggregated results. This is in contrast to the continuously growing nature of the World Wide Web. As a result, including the latest data is one of the key aspects of data mining on the web. This is exactly the topic covered by the *BlogIntelligence* project. The presented work in this paper is integrated into the *BlogIntelligence* project. There are three main steps involved to visualize blogs in the BlogIntelligence project:

1) Extraction: In the extraction step the blogs are basically crawled. In order to achieve this a, purpose-built crawler needs to be used as traditional crawlers do not fully meet the particularities of blogs as opposed to conventional websites.

2) Analysis: The analysis step prepares the crawled data for visualization. Each blog is analyzed by multiple *Analyzers*, that process its details in certain ways. Among potentially others, there are *data analyzers* that store the meta information about the blogs into the database, *content analyzers* that store information about the content which allow content-related analyses and there are *network analyzers* that store information on the relationships and links between blogs or other communities.

3) Visualization: The last step within the BlogIntelligence framework is the visualization of the analyzed information. The Blog IntelliTrends solution is part of this last step as it provides the stored data via an interface and visualizes them in client applications.

II. Related Work

The following section is split into two parts of related work. The first is about digital story telling, the second deals with summarization of social media content and analysis of blogs. In order to get a common understanding of a blog, we refer to Meinel et al.: “A blog is a journal like website that consists of reverse-chronological ordered articles called posts.” [2]. There are many different types of blogs considering topics, writing style, up-to-dateness, such as news websites to cooking blogs. The challenge is to create stories out of these heterogeneous information sources.

A. Digital Story Telling

The “Center for Digital Storytelling” in Berkeley has done research about digital story telling, they preserve the culture of story telling in the current times, offer books, workshops and projects to do so and provide example stories. We use their seven step approach to define what a story is and build our workflow to create a story upon these steps. The steps are the following [3]: 1. Point of view, 2. A dramatic question, 3. Emotional content, 4. The gift of your voice, 5. The power of the soundtrack, 6. Economy, 7. Pacing.

We identified the parts of a story in some examples (e.g. a project from BBC where ordinary people can tell video-stories [4]). The text of our story isn’t written by the tool, since we summarize the original author’s content. Thus the point of view changes only across multiple posts. The same applies for the seventh point, whereby the progression and rhythm of the story might stay similar for one post, but can change across multiple posts. If we find a dramatic or guiding question in the post, the author states important issues and may solve this afterwards. This stylistic mean keeps the readers attention and are important in a story. To add emotional content and tell the story in a personal way, we focus on sentiments extracted from the blog and Twitter as described in subsection IV-H. We combine the fourth and fifth point by adding context information from Wikidata and include multimedia content to. Finally, we condense all the content into a reasonable amount that does not overload the reader with too much information. This amount is first set to 10 sentences but can be expanded as well.

B. Summarization

One essential part of a story is the summarization of the text in a blog. In this area, we build upon existing work, since especially with Twitter, there are several approaches to summarize the content automatically [5], [6]. It is common in summarization to rank sentences according to their relevance or similarity and leave out the unimportant ones (extractive summarization). This allows to condense the text into the most important facts, but avoids writing them completely new in your own words which would raise grammatical and structural problems. In contrast to Tweets, blogs contain more textual information which allows building a sufficient corpus and analyze dependencies between sentences. Therefore, we use the LexRank algorithm to rank sentences [7]. LexRank is a stochastic graph-based approach to define a relative importance of sentences based on their entropy and similarity. It assesses the centrality of sentences with TF-IDF vectors and is used for single-document summarization. As one of our main contributions, we propose an adaption of LexRank for multi-document summarization. We combine this with approaches from social media summarization, such as language style and redundancy scores [5]. See subsection IV-G for more details on this.

Summarizations can be rated with respect to conciseness, readability and completeness. These criteria are also supported by NIST[2], which conducts the annually Document Understanding Conferences. *Conciseness* means, the summarization is as short as possible under the other presumptions. In other words no irrelevant information is included, all included information is relevant (high precision). *Readability* means that sentences form a coherent text that is as easy as possible to read. *Completeness* means every relevant information is included in the summarization (high recall).

[2] http://www-nlpir.nist.gov/projects/duc/index.html at 18.02.2015

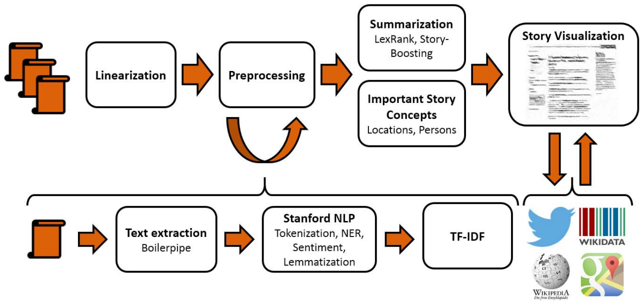


Figure 2: Processing Steps

Search engines, such as Google[3], provide short summarizations for each search result record. This summarization is context-sensitive, which means the summarizations for each search result is dependent on the search query. It puts emphasis on sentences which include the search keywords and displays parts of these sentences. In our context of multi-document summarization, we deal also with context-sensitive summarization. Here, the context consists of the relevant topics that all relevant documents have in common. So the challenge is to enrich the information gained from one single summarization with information from the other documents, while keeping the entire summarization as short as possible.

Multi-document summarization is already done by the Ultimate Research Assistant [8]. For a given query this search engine creates a summarization of publicly available online sources including tag clouds or histograms. The difference to our work lies in the idea of telling a story, instead of only gathering information and that the number of information sources is smaller for us. Our goal is more to interlink relevant parts of blog posts than finding these relevant posts in the web with information retrieval approaches.

III. Process

In this section, we give a short overview about our processing pipeline. The individual steps will be explained in the later sections.

As shown in Figure 2 we start with a graph of blog posts. This graph is evaluated to find a linear order for these posts. Each blog post is preprocessed to extract the content, find entities and assign TF-IDF values to each word. Afterwards we process the text to form a summarization. The summarization algorithm identifies which parts of each blogposts contribute to the story and avoids repeating information. We also identify the most important story concepts. Finally the story will be visualized enriched with further information from twitter, wikipedia, wikidata and google maps.

IV. Information extraction

A. Text extraction

First of all, we use *boilerpipe*[4] to remove unnecessary content like advertisement, the navigation, HTML-tags and frames. The initial HTML-code includes more semantic information than the plain text, for instance paragraphs group the text in parts, headings are given in h-tags and other tags contain more specific semantics (e.g. quote, acronym, address). To get this information, we use a boilerpipe extractor that returns titles and textblocks. This works reliably and the resulting text is analyzed for entities in the next step.

B. Named Entity Recognition

After extracting the content of the blog post, we extract the topics. The Stanford "NLP Group" has done research in natural language processing and published implementations of many NLP algorithms in a toolkit. We use the Java Library *CoreNLP*[5] to detect entities in the text of the post. The plain text from the boilerpipe pass is analyzed and a category is assigned for each recognized entity. We are using the 7 class model "english.muc.7class.distsim.crf.ser.gz" that is distributed with the CoreNLP toolkit. It is able to tag entities as persons, locations, organizations, dates, times or money and percent values. For further improvements it might be useful to add models for other languages too. The CoreNLP website offers models which were trained for German, Spanish and Chinese texts [6]. This way, entities which are mentioned in the post are identified. During later processing steps the included type information is used to highlight important story elements, such as important actors or the main location. The entities are also used to detect important new pieces of information that should be part of the summary of a given article.

C. Entity Linking using Wikidata

Additionally, detected entities are linked to Wikidata-Items thereby enriching the existing content. The type information described in subsection IV-B is essential in the linking process. First a search query containing the name of an entity is send to the Wikidata API. In most cases many different Wikidata-Items are returned as potential matches. In the next step the type information of the processed entity is used to disambiguate between the Wikidata-Items in the result by filtering out likely mismatches based on a rule system. This rule systems maps properties that are frequently used in the context of Wikidata to corresponding types recognized by the NER system. Consequently, an item with the property *Birthdate* is considered a potential match for an entity of type *Person*, while the property *Postal Code* indicates a *Location*. In case the described filtering process leaves more than one item, the disambiguation system relies on the internal ranking of the Wikidata API, which aims to favor items that are of general interest and importance.

Once an item is determined as a match it can serve as a central hub for further linking to additional sources: Items on Wikidata are connected to their corresponding Wikipedia articles, while the Wikipedia API allows to extract images and a summary for a given article.

The described approach was implemented in an independent module that relies on the name and type of an entity to find representations in open data like Wikidata and Wikipedia. Apart from the previously mentioned application in enriching the content of stories, the module was also successfully used and tested on a large amount of entity information, which had

[3] http://www.google.com

[4] version 1.2.2 https://code.google.com/p/boilerpipe/

[5] version 3.4.1 http://nlp.stanford.edu/software/

[6] Stanford NER http://nlp.stanford.edu/software/CRF-NER.shtml#Models

been previously detected during a wide-scale analysis of the blogosphere (see subsection VI-B).

D. Text Processing

In addition to the entity recognition, we also use the CoreNLP toolkit to do sentence splitting, tokenization, lemmatization and part-of-speech tagging. We create a data structure that represents the article and its structure of text blocks, sentences and fragments. Fragments are either entities or words that are not detected as an entity. These steps are very useful for our further processing steps like TF-IDF in subsection IV-E and summarization in subsection IV-G.

E. TF-IDF

In order to find and rank the concepts that are most important for the story, we make use of the TF-IDF measure defined by Manning [9], which is often used to judge the relevancy of a term for a document in a collection of documents. The term frequency is the quotient of the frequency of a term and the maximum frequency of any term. The document frequency is the fraction of documents that contain a term. For the TF-IDF measure the logarithm of the inverse of this value is used. We decided to use a word list with the frequency of common English words to get an approximation of the document frequency. The word list contains the 5000 most frequent words from the Corpus of Contemporary American English[7] For words that were not part of the word list we assumed that they were rare and assigned a frequency below the rarest term of the word list.

F. Graph Linearization

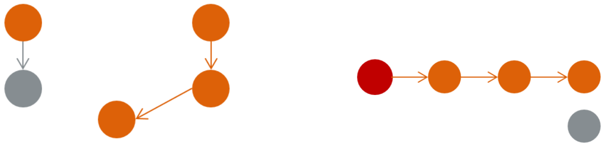
When we build a story out of several blog posts we have to arrange them properly. The first goal doing this is to preserve the chronological order of the posts to allow the reader an easy understanding. Moreover, we consider the links between posts to convey their relationship in the story and remove unrelated or duplicate posts.

The input of our system is a graph of posts, each annotated with the publication date, connected via edges if posts are linked. This graph can be created by the user or extracted from *Blog Intelligence*. We determine the date of posts by analyzing the URL with regular expressions for a date pattern. Alternatively, we use HTML meta-tag values as proposed by Google[8] like pubdate, DCTERMS.issued, datePublished, DC.date.issued.

We classify the posts as relevant for the story or not if on one hand a post is about a different topic or on the other hand two posts are quite equal. The topic of the story is given by the earliest post in the graph. By comparing the most important entities of two posts we reject a post as irrelevant, if they have all concepts in common (duplicate) or less than two (unrelated). The most important concepts are the top twenty entities according to their respective TF-IDF ranking as described in subsection IV-B.

The linear order of the posts is determined in several steps:

1) Order all posts with a date chronologically.
2) Consider remaining posts without recognized date but outlinks to other posts in the graph. Put these right behind the linked post, since they were published afterwards.
3) Consider remaining posts with inlinks from other posts in the graph. Put these right before the linking post, since they were published before.
4) Append all reaming posts at the end.



(a) Graph of linked posts

(b) Linear storyline after processing

Figure 3: Graph linearization

In Figure 3 an example graph is shown and how the different passes are applied. With this procedure, we create a chronological storyline which also considers the relations between posts and allows the reader an easy understanding of the topic.

G. Summarization

With respect to the sixth point of story telling, Economy, we have to reduce all the collected information and show only the most important facts to the reader. To condense the text from the main post and all the related ones, we use the LexRank algorithm [7]. With the help of the CoreNLP library, we get a list of lemmata (distinct words) and generate a TF-IDF vector for each word with the text corpus. The term frequency and inverse document frequency measures the importance of words in the text in relation to the general corpus. Based on this, we create a similarity-graph for each sentence which also includes the weight of single words as TF-IDF values. With a language style score considering the spelling, and redundancy, we enhance this weight. We take the highest ranked sentences from the graph and all the similar sentences, since they deal with the same topic and help the reader to understand the text. The threshold of importance is adjusted with the amount of text the reader wants to see.

We build one part of our summarization algorithm based on the idea of LexRank. LexRank is a stochastic graph-based approach to rank sentences. But this approach cannot deal with the challenges of multi-document summarization. We therefore extend LexRank to multiple documents in order to fit our demand. First, we use one document as the main document for the entire story. From this main document, we extract the main concepts by their TF-IDF values. We use the main concepts to classify additional documents as unrelated or duplicate content. According to global and local story concepts we then boost sentences so that they are more likely to be included in the summarization. When a sentence adds new information to the story, for example by mentioning an entity that not occurred before, it is boosted. This boosting can spread in the graph of sentences of the LexRank algorithm.

The result is a ranking of sentences by LexRank which form a multi-document summarization with respect to concise-

[7] http://www.wordfrequency.info/free.asp

[8] https://developers.google.com/custom-search/docs/structured_data#metatags

ness, readability and completeness. Completeness is provided because sentences that include entities that are not included in other sentences are boosted. Conciseness is reached by excluding duplicate and unrelated content and by choosing only the highest-ranked sentences for the summarization. Readability is covered by carrying over the original sentences. We assume that consecutive sentences from the original texts are easier to read, than a rearranged version of these sentences. That is why we do not change the order of sentences in our summarization. It also sticks to the chronological order of the documents as previously described in subsection IV-F.

H. Sentiment Analysis

With respect to the third point “Emotional Content” in the seven-step approach of story telling described in section II, we analyze the sentiments of sentences. As shown in subsection IV-B, we use the *CoreNLP* library for entity recognition. Simultaneously the library allows the detection of sentiments by using machine learning with neural networks and a model trained on labeled sentences. Each sentence is categorized in one of the following sentiments: strong positive, positive, neutral, negative or strong negative. Currently, we only use both of the strong sentiments for the visualization assuming a higher probability of a correct detection with these.The positive sentences are highlighted in green, the negative ones in red.

V. Story Composition

Another challenge of the project is the creation of a story from the gathered content. The story itself is a website in HTML with interactive elements in JavaScript, such as additional information displayed on clicks, sentiment highlighting or automatic expansion.

The main content of the story is the text from different blog posts, which is arranged in chronological order from top of the page. The earliest post serves as the main content of the story. Each blog post is condensed to the basic information as described in subsection IV-G and consists of only up to 20 sentences. The text of different posts is grouped in paragraphs so that the reader is able to distinguish between the sources. The date of the post is shown if it could be extracted from the URL or the text. Structural elements, such as headlines or quotations are highlighted in other font size and style, since they make the story vivid and interesting. Information about sentiments and opinions is added in terms of Tweets about the topic on the left side. It provides more subjective information with different points of view. The overview of an example story is given in Figure 1 on page 1.

With interactive extensions of the story, we try to support the reader in understanding the topic. For instance, the detected entities are highlighted in bold and show different subjects of the story. If the entities are unknown to the reader he can hover above and an explanation from Wikidata is shown on the right.

Since the blog texts often contain subjective elements, we analyze the sentiments as described in subsection IV-H. The result of this is activated with one click and all positive sentences are highlighted with a green background, the negative ones in red. This feature is currently turned of, as it is computationally expensive and we did not focus on improving the run-time. Without this feature the story can be generated much faster with better feedback for the user.

With the help of GoogleMaps[9], we can show the setting of the story in a map. The most relevant terms of the story are shown in a tag cloud and the most relevant person is emphasized as well.

VI. Discussion

We evaluated the quality of the generated story from two different aspects. Most important one is the summarization of the textual content. Further, we assess the quality of the entity matching.

A. Summarization with LexRank Adaption

For our discussion, we compare the original summarization approach of LexRank on single documents with LexRank on all documents combined and with our adaption on boosted entities.

We found out, that summarizing single documents with LexRank and then combining these summarizations leads to many redundant sentences stating the same facts. Summarizations of unrelated documents are combined to a story, which makes this approach impracticable. Performing the LexRank algorithm on a combination of all documents performs better, but again the problem is, that LexRank is not able to deal with redundant sentences correctly. In contrast these redundant sentences occur often in multi-document summarization. Therefore, we boost sentences that introduce new facts in the form of new entities. This boosting gives comprehensible summarizations for our example sets, such as news posts or website pages. For demonstration, we build a story based on three posts: one web page about studies at HPI in general[10] and two times the same web page about bachelor studies applications[11]. As the last two posts have the same content, one is discarded and the we bpage is processed only once. Duplicate sentences of the general web page and the application web page, such as “The Hasso Plattner Institute offers degree programs in ’IT Systems Engineering’ that are unique Germany-wide.” are included in the created story at most once. Using the general web page as the main post for the story, sentences from the application web page are boosted when they introduce new entities or if the accumulated TF-IDF values of words in a sentence suggest a high relevance. This holds for the words *application*, *bachelor* and *University of Potsdam*. Also the contact information of the study advisor for the Bachelor’s Program is included in the story. Due to the name and the phone number being recognized as new entities, it is assigned a high relevance. Sentences that do not include any entities, such as enumerations of adjectives, get low relevance values, low ranks and are often discarded, for example “It is distinguished by its high scientific standard, practical approach and close cooperation...”.

As a difficult task for our approach, we identify texts where most or all entities occur only once in the text. This can happen

[9] http://maps.google.com

[10] http://hpi.de/en/channel-teaser/studium/it-systems-engineering... at 09.11.2014

[11] http://hpi.de/en/studies/application/application-bachelors... at 09.11.2014

Table 1: Example categories and their number of matchings

Category	match	missmatch	percentage
Persons	92171	231132	29%
Commercial Organizations	11224	46657	19%
Unclassified Organizations	6278	20473	23%
Total	135021	336519	29%

with already summarized texts or when synonyms are used for the same entity and when our approach is unable to map these synonyms to the same entity. In this case it is harder to decide, which entities are more relevant and thus should be part of the story.

Although, this discussion is far from complete, we can show that the boosting combined with LexRank delivers a more concise and easier readable multi-document summarization than LexRank on representative examples. The criterion of completeness is influenced by the length of the summarization. Our goal is to find the most relevant facts for a story and not to achieve completeness. As we leave out the less relevant sentences and show the more relevant ones in the sumarization, we can guarantee that every left out sentence is less relevant than the shown sentences. To make the summarization more flexible for the user, the length of the summarization can be adjusted. So it can be a short overview or a longer story.

B. *Entity Matching*

Our story telling approach is ready to be integrated in the *BlogIntelligence*[12] project. Therefore, we crawled Wikipedia information for every recognized entity occuring in blogposts in the database. After having crawled 471540 entities in total, we can state, that approximately 26% find a valid matching. 336519 entities did not find a matching in Wikipedia at all. This is most often caused by persons that have no wikipedia page and therefore have no significant influence on the recall of the matching procedure, because it is by design a necessary condition for the entities to be represented in open data. Table 1 shows the number and the percentage of matchings for persons, commercial organizations, and organizations that have no further classification.

We evaluated a random sample of 100 matchings and found out that 8 of them are false matchings, resulting in a precision of around 90% for the matching procedure. We are confident that the accuracy could be increased by further preprocessing the recognized entities. Nevertheless, the improvement of this was not the scope of this work, since the main challenge of generating stories automatically from multiple blog posts is met with our approach as demonstrated by the examples. Without doubt, this approach can be improved and extended to work for additional types of entities and to provide the user with an even wider range of information.

VII. Conclusion and Future Work

We presented an automated approach to create stories from multiple input documents. The developed framework implements strategies to visualize stories and link content to related sources of information, such as images, tweets and encyclopedia records ready to be explored by the reader. Our approach combines deriving a story line from a graph of interlinked sources with a story-centric multi-document summarization.

We defined important elements and building blocks of a story and introduced an adaption of the single-document sentence rank algorithm LexRank in the context of a strategy for multi-document summarization, which explicitly incorporates the previously determined story elements. The sub-module that performs entity matching achieves a precision of approximately 90%, so that only a small amount of entities is linked incorrectly even though we rely on a very generic approach that leaves room for tuning the algorithm for specific entity types. Although there are available gold standards for evaluating summarizations, the assessment of stories and the various aspects involved is a new field. Because perspective and therefore subjectivity plays an important role, an objective evaluation is difficult. Future work could be done in detecting turning points in stories by integrating information about shifts in sentiments and the perception of important story elements. Additionally more sources of information could be considered, such as weather information for outdoor events. Domain-specific extensions for sports events or political events are imaginable as well. Furthermore the visualisation of a story could be improved to dynamically adapt to different use cases and story types. We conclude that, even though we successfully developed a framework that automatically generates a story based on multiple blog posts, this is very broad task, which is far from being solved and remains a challenging research field.

References

[1] P. Hennig, P. Berger, D. Kurzynski, H. Rantzsch, and C. Meinel, “Efficient event detection for the blogosphere,” in *The 7th IEEE International Conference on Social Computing and Networking (SocialCom 2014)*, 12 2014.

[2] P. Berger, P. Hennig, and S. Detje, “Blogsphere-a topical map of the blogosphere,” 2014.

[3] B. R. Robin, “Digital storytelling: A powerful technology tool for the 21st century classroom,” *Theory Into Practice*, vol. 47, no. 3, pp. 220–228, 2008.

[4] D. Meadows, “Digital storytelling: Research-based practice in new media,” *Visual Communication*, vol. 2, no. 2, pp. 189–193, 2003.

[5] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, “Entity-centric topic-oriented opinion summarization in twitter,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’12. New York, NY, USA: ACM, 2012, pp. 379–387.

[6] F. C. T. Chua and S. Asur, “Automatic summarization of events from social media.” in *ICWSM*, 2013.

[7] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[8] A. Hoskinson, “Creating the ultimate research assistant.” *IEEE Computer*, vol. 38, no. 11, pp. 97–99, 2005. [Online]. Available: http://dblp.uni-trier.de/db/journals/computer/computer38.html#Hoskinson05

[9] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press., 2008.

[12] http://www.blog-intelligence.de/