Demo: HistoryViz – Visualizing Events and Relations Extracted from Wikipedia

Ruben Sipoš[1], Abhijit Bhole[2], Blaž Fortuna[1], Marko Grobelnik[1], and Dunja Mladenić[1]

[1] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{ruben.sipos,blaz.fortuna,marko.grobelnik,dunja.mladenic}@ijs.si
[2] Indian Institute of Technology, Bombay, Mumbai - 400076, India
abhijit.bhole@cse.iitb.ac.in

Abstract. HistoryViz provides a new perspective on a certain kind of textual data, in particular the data available in the Wikipedia, where different entities are described and put in historical perspective. Instead of browsing through pages each describing a certain topic, we can look at the relations between entities and events connected with the selected entities. The presented solution implemented in HistoryViz provides user with a graphical interface allowing viewing events concerning the selected person on a timeline and viewing relations to other entities as a graph that can be dynamically expanded.

1 Introduction

The challenge we are facing with growing amount of electronic data on the Web is how to find and extract the knowledge we want form the vast amount of the available data. The most common way is searching using keywords or even just browsing using entity names and links. The main drawback of this approach, when we are looking for a broader overview of some topic and not only a specific fact, is the amount of human work required to extract and consolidate facts from a number of web pages found using a search engine.

Another approach to fact discovery and knowledge acquisition is using textual data that is already in (more or less) structured format. Instead of browsing through dozens of search engine results to find relevant facts about e.g., iron we can go to Wikipedia[1] article and find everything on a single page. Similarly, we could use RDF data from some provider. However, doing this we do not fully exploit the potential that data sources provide us with. If we used those data sources as a whole and not only the explicitly relevant bits, we could gain some additional knowledge. While this can be done manually, it is too time consuming because we have to manually combine a lot of small pieces into a larger picture. Because data sources we are considering are at least partially structured we can use text mining [3] and similar approaches to automatically extract data from multiple subsections and present it's summary to the user.

In this demo, we present HistoryViz[2], a web application allowing user to explore events (extracted from Wikipedia) connected with selected persons presented on a

[1] http://www.wikipedia.org/
[2] http://historyviz.ijs.si/

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 903–907, 2009.

timeline and to browse the network consisting of persons described on Wikipedia. HistoryViz uses data extracted using text mining and presents a new perspective on indirectly available data contained in Wikipedia that can be acquired automatically. Our main focus was on the entity network and timeline of events although using a similar approach one can also gain some other information.

1.1 Related Work

The demo presented in this paper is using some of the data extracted from Wikipedia using the approaches described in [1], where articles from Wikipedia were categorized (using binary linear SVM) into articles describing persons, places and organizations. They also show how to extract events, described in Wikipedia articles about persons, and their time frame and extract link data used for building the entity network and determining the importance of nodes. In our work we have extended the existing approaches by using and incorporating other data sources, such as Freebase[3] and performing some additional steps (e.g., associating keywords with events) in processing the data. Due to a space limitation of this paper, only the closely related work is discussed here, more can be found in [1].

1.2 Motivation

The advent of web sources providing semi-organized data has drastically improved user's experience when using the web to gain new knowledge. Still, if user wants some information that can be obtained by gathering facts from different articles or entries, there is usually a lot of manual work involved. Because the data is not strictly structured we can not use most of existing tools. However, we can use text mining on this data and provide new ways to view data to the user.

This demo shows how to use already existing data in Wikipedia to provide user with an interface displaying temporal data and social network based on data obtained by combining pieces of information gathered from many articles. For example, user writing a report on some important historical person might have to read through many articles to gather enough biographical data and discover relations with other persons. On the other hand, by using our interface as implemented in HistoryViz it is easy to see all important relations (presented as an entity network) and timeline of the events which happened in a life of the selected person.

Moreover, we wanted to show that it is possible to simplify certain user tasks and provide additional views on already existing data using currently available technologies. Many data sources are still not fully exploited because it would require too much manual effort and because the structure is too lax to allow use of current tools. Nevertheless, we can now take advantage of that still unused potential.

2 HistoryViz

HistoryViz is a web based application that allows the user to view events related to selected person's life on a timeline and to explore social network based on relations found in Wikipedia. There are numerous browsers which simplify exploring of

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

Wikipedia. However, they usually present hyperlinks between articles in a graphical manner. Most of them directly use links whereas we consider the whole entity network based on relations found by examining content of the articles. Moreover, we provide timeline view of data automatically collected from separate articles, while other solutions take already consolidated data or gather data from strictly structured sources such as RDF.

In timeline view shown on Figure 1 the user selects person by entering full or partial name into query box. System finds the corresponding article in Wikipedia describing that person. Events (extracted from Wikipedia) are then shown on the timeline, which consists of two parts. The top part shows bars representing the life from birth to death of the found persons. Also, the most closely related persons to the one user selected are shown. They are chosen by exploring local neighborhood in the entity network and using heuristics mostly based on PageRank [2] and cosine similarity of the link structure. This enables the user to easily explore correlation of events in lives of closely related persons.

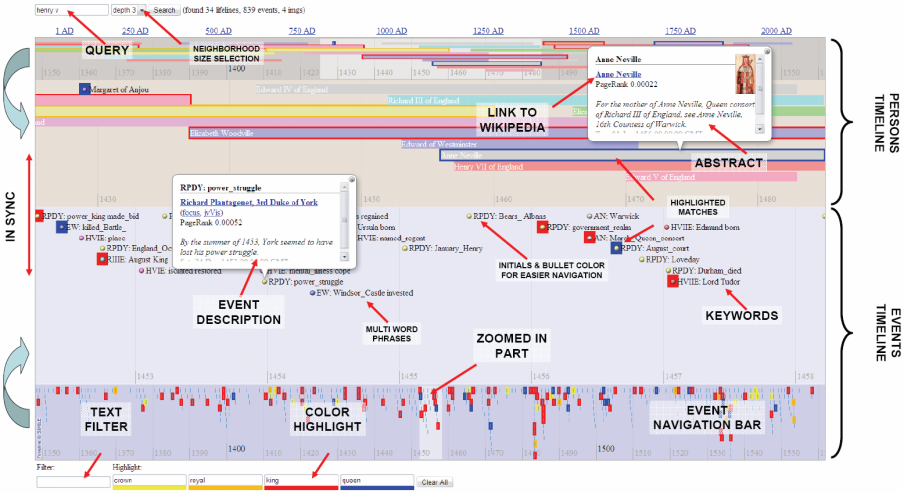

Fig. 1. Detailed description of user interface in the timeline view

For each displayed person, the user can also view a picture (taken from Freebase) and a short abstract of the person's biography (the first paragraph of the corresponding Wikipedia article). The bottom part of the timeline shows events described with a few keywords. The keywords are calculated [5] from the text describing events. The user can also read the event description after selecting the desired event.

The user can explore network of entities using a java applet (based on Prefuse[4] [3]) shown in Figure 2. The starting node is selected by searching for article corresponding to the user's query. The graph is dynamically expanded as the user traverses over it.

[4] http://prefuse.org/

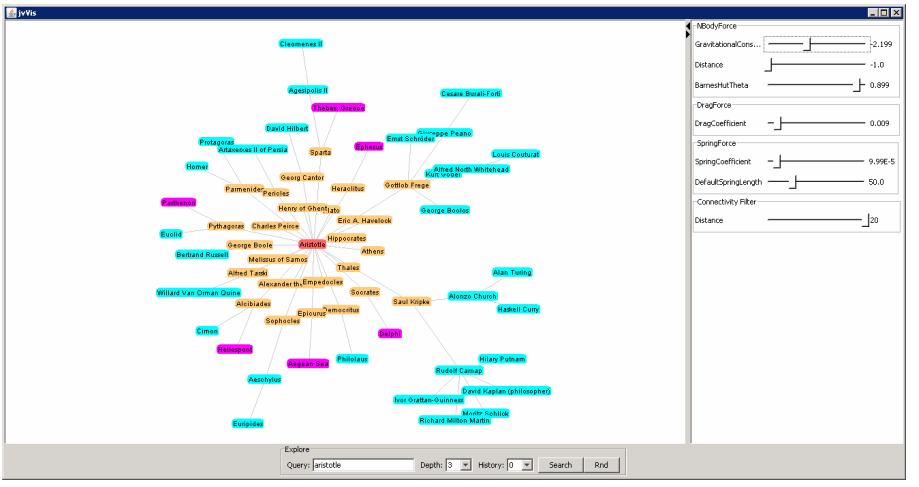

Fig. 2. Example of a part of the entity network using our java applet. The starting point was query "Aristotle". Colors represent entity categories (persons, places and organizations).

Both views are interconnected. The user can always switch from one to another. By doing this the currently displayed items in one view are transferred to the other to help user keep the current context.

2.1 Implementation

HistoryViz is implemented as client/server application. Server is written in C++ and has all the needed data stored in binary format to lower access time. The pipeline for data processing (from raw dumps to internal indexed binary format) is streamlined and therefore any updates to the data can be easily performed. Clients are Ajax web application displaying timeline and java applet for browsing the entity network. Both communicate with the server using HTTP and exchanging data in XML format. This architecture allows developers to easily add new clients with different functionalities and keeps a few gigabytes of data on the server side while exchanging only small portions of it with the client. In this demo we have 3.2GB on the server side and typically exchange 200-400kB with the client (most of it being textual descriptions of events).

3 Demo

During our demo, visitors will have the opportunity to try out our application, hear detailed description of all the implemented features and if interested see some parts of the application that are hidden from the end user (such as, intermediate data files, details about server implementation, contents of client/server communication and how we use data from other sources e.g., Freebase). We will provide sample tasks and show how our application simplifies them. Also, we will demonstrate how users can discover new and interesting facts using our application because the data is presented in a different way.

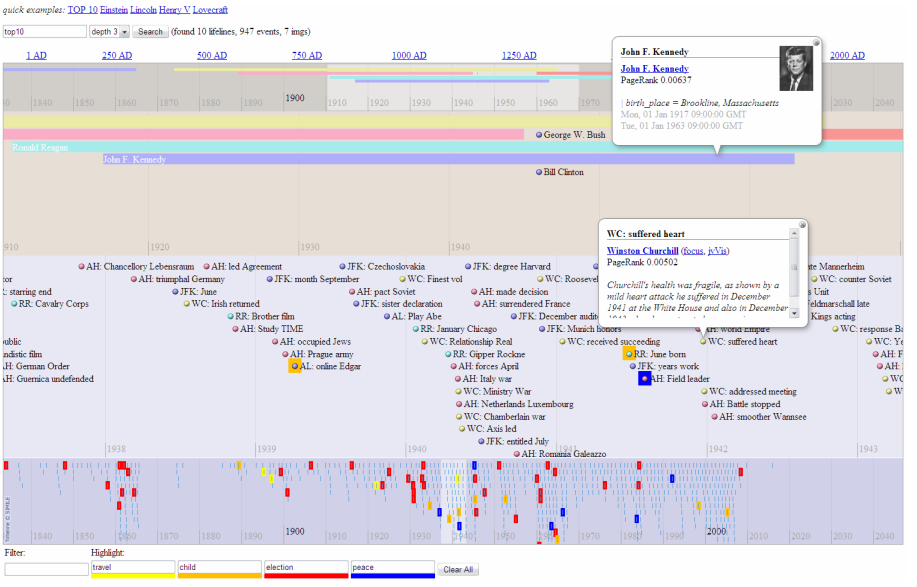

Fig. 3. Persons displayed on this timeline were selected as the most important nodes given by PageRank. This can give some additional insight into Wikipedia's content and world history.

References

1. Bhole, A., Fortuna, B., Grobelnik, M., Mladenic, D.: Extracting Named Entities and Relating Them over Time Based on Wikipedia. Informatica, Part 4 31, 463–468 (2007)
2. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Stanford Digital Libraries Working Paper (1998)
3. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
4. Heer, J., Card, S.K., Landay, J.A.: Prefuse: a toolkit for interactive information visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Portland, Oregon, USA, April 2-7 (2005)
5. Grobelnik, M., Mladenic, D.: Text Mining Recipes. Springer, Heidelberg (2009), `http://www.textmining.net`