Quelle:
Fabian, Odoni; Semar, Wolfgang; Elena, Mastrandrea: Development of a Benchmark System for Collaborative Online Knowledge Management Systems. In: Maria, Gäde; Violeta, Trkulja; Vivan, Petras (Eds.): Everything Changes, Everything Stays the Same? Understanding Information Spaces.Proceedings of the 15. International Symposium of Information Science (ISI 2017), Berlin, 13.-15. March 2017. Glückstadt: Verlag Werner Hülsbusch, pp. 386-388, 2017

VICI

Visualisation of Collaboration in Social Collaborative Knowledge Management Systems

Fabian Odoni
Swiss Institute for Information Research (HTW Chur), Switzerland
fabian.odoni@htwchur.ch

Wolfgang Semar
Swiss Institute for Information Research (HTW Chur), Switzerland
wolfgang.semar@htwchur.ch

Elena Mastrandrea
Swiss Institute for Information Research (HTW Chur), Switzerland
elena.mastrandrea@htwchur.ch

Abstract

This paper describes the relationship visualisation framework VICI. VICI was developed to extract co-author relationships found on an Atlassian Confluence installation and visualise them as network graphs. Therefore wiki-entries are extracted directly from the Confluence database using sql-queries. This data is used to calculate co-authorship bigrams and build a relationship graph. The graph is then visualised as force-directed network graph using D3.js in a webapp. Node size and colour as well as path width and colour are used to encode relationship degrees and author activity. New visualisations for different aspects of the wikis and wiki-entries as well as topic relationships and expert identification will be implemented in the future.

Keywords: information retrieval; data visualisation; knowledge management; software development

1 Introduction

Globalisation and the therein resulting increasing interconnectivity is forcing companies to stay competitive by treating knowledge as a resource and managing and maintaining it accordingly (Hopfenbeck et al. 2001; Nohr 2001).

This can be done by using Social Collaborative Knowledge Management Systems (SCKMS) like Atlassian Confluence[1], IBM Connections[2] or Humhub[3]. These software based knowledge management products are conceived as Facebook-like social media applications combined with knowledge management functionalities similar to Wikipedia and features such as Q&A implementations, file and document management, and more.

Data visualisation can facilitate the comprehension of different relationships between users and content found in SCKMS. This allows a company to make better informed decisions and therefore increase effectivity and efficiency (Meyer 1999).

VICI is a framework designed to extract data from Atlassian Confluence - a SCKMS focusing on wiki based knowledge management - and calculate and display co-author relationships therein.

2 Architecture and Methods

VICI is composed of a Python[4] based backend system that extracts the data and calculates the relationships needed for the visualisations, and a web based frontend that displays the relationship graph and allows some user interaction with the graph as well as the backend system.

2.1 Data Extraction

At the moment VICI focuses on the wiki-entries written in the Confluence wikis. The data extraction has to take into account that Confluence does not only store the latest version of a wiki-entry but also all its revisions. These revisions are linked by id numbers.

[1] https://www.atlassian.com/software/confluence
[2] http://www-03.ibm.com/software/products/en/conn
[3] https://www.humhub.org
[4] https://www.python.org

In a first step all database entries are extracted directly from the source database (at the moment a PostgreSQL[5] database with a copy of an Atlassian Confluence database on it) using sql-queries. The wiki-entry versions are then grouped by their original id - the first version of the entry - and all the authors of the wiki-entry are aggregated and linked to the respective wiki-entry. This data is then used to calculate the needed relationships.

2.2 Calculating the Relationship Graph

Since every author of a wiki-entry is a co-author of every other author of the same wiki entry, a combination *([A, B, C] -> [[A, B], [A, C], [B, C]])* of all the authors per wiki-entry is calculated. These bigrams are then entered into a NetworkX[6] graph.

A weight attribute per link, representing the number of different wiki-entries two linked authors have collaborated on, is set as well as an author weight attribute, calculated by counting the number of different wiki-entries an author has worked on. This NetworkX graph is then converted into node-link JSON data and transmitted to the visualisation module of VICI.

2.3 Visualising the Graph

The frontend graph visualisation uses the D3.js[7] library to build a force-directed graph. Authors are displayed as nodes, and the relationship between them are shown as links between these nodes. For every node, the author weight is logarithmically scaled and used to define the node's size and colour. Using the weight value of the links between the nodes, the stroke width and colour of the link paths as well as the distances between the nodes are calculated. Additionally a mouse-over function was implemented to highlight a node and all of its first degree links.

2.4 Web Framework

In order to run the whole pipeline, a small web application was built using the Flask[8] microframework. This application fetches and calculates the needed data, provides a web interface to display the visualisation and offer the option to switch between different databases.

---

[5] https://www.postgresql.org
[6] https://networkx.github.io
[7] https://d3js.org
[8] http://flask.pocoo.org

The web framework and a PostgreSQL database where packed in two separate Docker[9] containers linked together using Docker-Compose. The web app container was configured to run the web app from a host directory mounted into the container. The PostgreSQL container is configured to run a startup script that checks, if database dumps are located in a specified folder and if so, loads them as separate tables into the database, in case the tables do not exist yet. The frontend can be accessed using a browser and the url pointing to the web app container (eg. http://localhost:5000/).

3 Initial Tests and Results

Initial tests of the framework were done using three different sized Confluence database dumps (117.1MiB, 187.7MiB and 354.2MiB). While small datasets showed an easily readable graph, bigger datasets became increasingly more difficult to comprehend. Using the mouse-over highlighting feature this could be partially counteracted, but additional implementations of visual as well as algorithmic features are needed to optimise the graph-based visualisation of larger datasets.

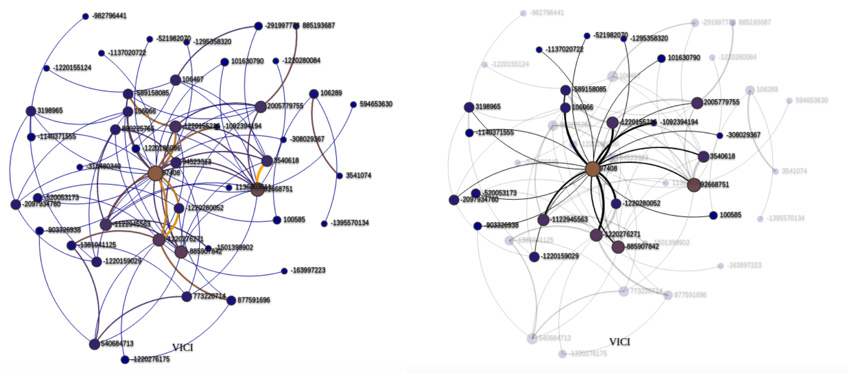

FIG. 1: VICI co-author visualisation of a 117.1 MiB Confluence database dump and highlighted nodes and links on mouse-over (displaying anonymized user names)

4 Future Development

Future development of VICI will include graphic and algorithmic improvements to the existing framework as well as the development of algorithms to display new relationships. Following new relations are considered at the moment.

[9] https://www.docker.com

Wiki-entry relationships: The relationships between wiki-entries calculated by analyzing the number of mutual authors between different wiki-entries can be used to generate a graph based visualisation that focuses on the connection between wiki-entries based on their authors.

Group relationships: Atlassian Confluence allows the users to create groups that focus on a specific topic or project. These groups are essentially collections of wiki-entries that "belong together". This visualisation will be similar to the wiki-entry relationship visualisation but offers a broader view on the whole content in a SCKMS.

Wiki links: Wiki-entries allow the authors to create links to other wiki-entries. These entries will be extracted and used to link two entries that are linked together visualising how the authors themselves define the connections between the content found on the SCKMS.

Topic links: This visualisation will aim to show how wiki-entries or whole wikis are topically linked together, ignoring co-authorship or wiki links, but by comparing the tags assigned to the different wiki-entries and by using automated keyword extraction methods. The wikis or wiki-entries will be linked by the number of mutual tags and keywords used to classify their content.

Expert identification: Confluence Questions[10] adds stackoverflow.com like Q&A functionality to Attlasian Confluence. It is planned to use this feature in combination with topic/keyword extraction methods to identify users that successfully answer questions related to certain topics. A visualisation, made by linking users to certain topics they seem to be knowledgeable about, will produce a knowledge map of the organisation, allowing the identification of existing knowledge as well as knowledge gaps.

References

Hopfenbeck, Waldemar; Müller, Manuela; Peisl, Thomas (2001): Wissensbasiertes Management: Ansätze und Strategien zur Unternehmensführung in der Internet-Ökonomie. Landsberg/Lech: Verlag Moderne Industrie.

Nohr, Holger (2001): Wissensmanagement: Wissen wird zum Fokus betrieblichen Managements. In: A. Blum (Hrsg.): Bibliothek in der Wissensgesellschaft (S. 413–421): De Gruyter.

[10] https://www.atlassian.com/software/confluence/questions

Meyer, Jörn-Axel (1999): Visualisierung von Informationen: verhaltenswissenschaftliche Grundregeln für das Management. Wiesbaden: Gabler.