Interactive Search and Exploration of Entity-entity Relationships in a Huge Document Corpus

Andreas Schmidt*† and Steffen Scholz*
* Department of Computer Science and Business Information Systems,
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Email: andreas.schmidt@hs-karlsruhe.de
† Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: andreas.schmidt@kit.edu, steffen.scholz@kit.edu

***Abstract*—This paper presents an interactive tool developed for the search and exploration of named entities and their relationships. The tool sits on top of an entity-based search engine, which previously has extracted and indexed all entities from a potentially huge document corpus. Relatedness between different entities is calculated based on entity n-tuples in the document corpus. The relatedness measure between entities is calculated during indexing time, which makes the algorithm very fast and usable for interactive application. Furthermore, the user can search for entities and their relationships to other entities using an interactive auto-completion and suggestion service. Related entities can then be filtered further by a multi-prefix search as well as based on type restrictions from an existing classification taxonomy. Another powerful feature is the merging of multiple entities into a group which allows the extraction of entities related to this group. A graphical interface is proposed with an entity or entity group as a central point, surrounded by the most n-related entities, based on some restrictions formulated by the user.**

Keywords–Interactive graph; entity-based search engine; relationship exploration; graphical representation.

I. INTRODUCTION

A. Motivation

The advent of information-driven technologies has recently initiated massive document collections, the so-called Big Data, to exist and has enabled an exponentially growing demand for collecting as much relevant data as possible. Although these data collections can give access to rich knowledge, such a large scale of gathered information could technically preclude timely and effective data processing and hence be one of the main barriers to further growth and development of Big Data technology. This issue is not only restricted to general information, which can be obtained from the Web, but also applies to specific data in different fields [1] e.g., aviation, bank and security exchanges, medicine, engineering, and technology, and many others. This has motivated researchers to address such a challenging issue with the objective of advancing relevant computing technologies.

B. Problem

Processing a large collection of documents initially requires understanding the content of the documents. This is a process allowing the relevant entities or concepts (persons, cities, organizations, materials, diseases, etc.), and the way in which they are related to each other. Subsequently, and based on this step, further cognitive processes take place, which are mostly influenced by prior background knowledge of the human reader. Due to the huge amount of data, these steps cannot be done by a human for all documents available.

C. Solution

Having an automatic tool, which extracts the entities and their relationships, can give a first insight into a given document collection. With the emergence of Named Entity Recognition (NER) [2], Named Entity Disambiguation (NED) [3], and entity-based search engines [4] now exists reliable tools to help identify and extract entities from document collections. Also, complex concepts, consisting of multiple different entities can be extracted [5].

STICS [4], for example, is a search engine, which works on entities instead of on words. In particular, rather than building an inverted index from words, STICS identifies named entities in the text and uses these entities for building the index. Additionally, STICS performs the so-called disambiguation step [3], which identifies the correct meaning of the entity. As an example, consider the word "Paris", which could be the french capital, a greek deity, the biological name of a plan, or a blonde hotel heiress. The correct meaning can be extracted from the context, by looking for other entities in the surrounding. As a result of this entity recognition and disambiguation step, we have a clear picture of which entities occur at which places in the indexed documents.

The system presented in this paper, uses the previously mentioned technologies for identification and disambiguation of entities inside a document corpus and presents the entities and their relationships in an interactive graph, which ultimately allows the search and exploration of entities and their quantitative relationships to other entities of interest. It is worth emphasizing that in contrast to the large body of the relevant reported approaches in this field, this research study is not only focused on bilateral relationships, but it is extended to allow the acummulation of entities into groups and look for further related entities. For example, we can accumulate the entities "*Emmanuel Macron*" and "*Germany*" to form a group and find out which other entities are related to this group. Figure 1 illustrates a visual representation of the user interface, which

 ISBN: 978-1-61208-637-8

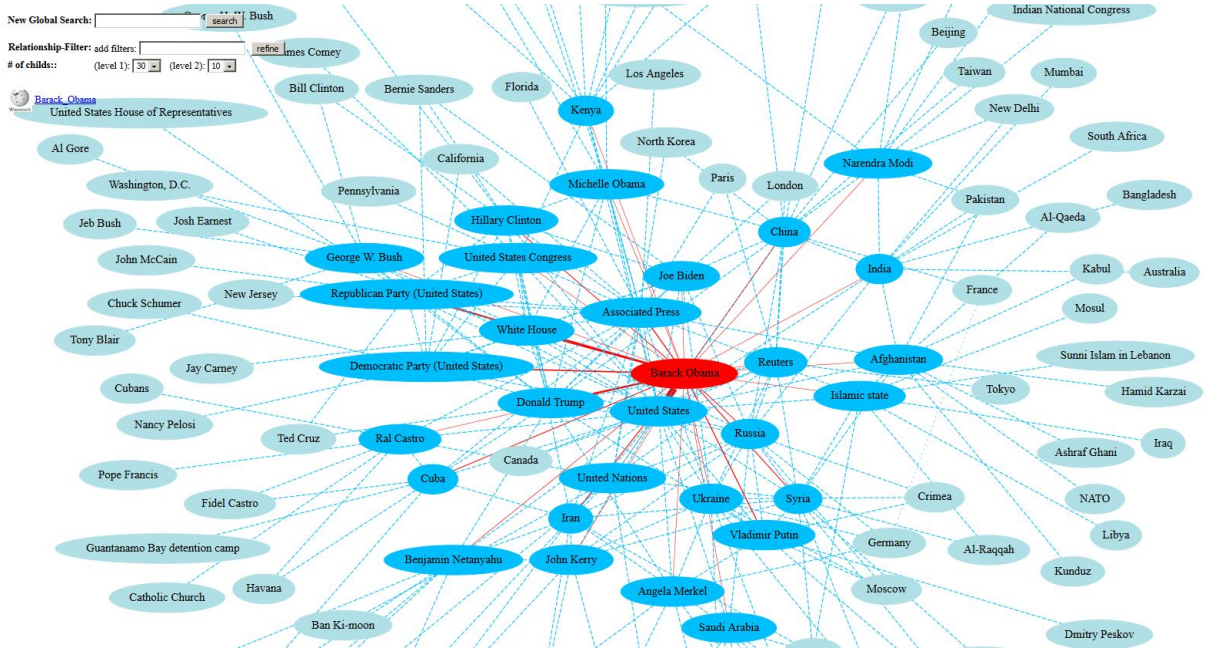

Figure 1. Screenshot of the graph-based interface.

allow navigation along entities and their relationships by just clicking on the nodes and edges.

The main contribution of this papers are the following: (1) Visual representation of the relationships of fully automatically extracted entities from a large document corpus. (2) Introduction of the concept of entity groups to formulate more complex concepts consisting of several entities and integrating them into the relationship graph.

The structure of the paper is as follows: In Section II we discuss what "relatedness" between entities mean. Then, in Section III we introduce the concept of our graph-based GUI. Section IV gives a short overview over related work and the last section finishs the paper with a conclusion and an outlook for further research avenues.

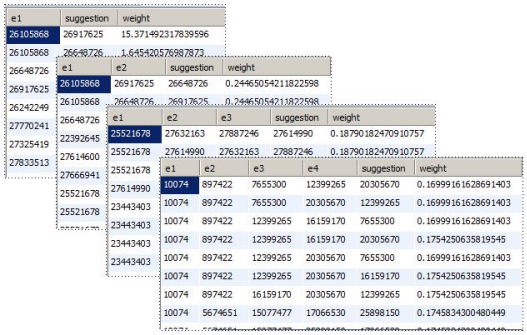

Figure 2. Precalculated materialized views for tuples, triples, quadruples and quintuples.

II. Relatedness Measure

The relatedness between two or more entities is based on the occurrence of the entities inside a sliding window of a predefined size. By shifting the window over the text of the documents, tuples, triples, and quadruples, along with other sequences of different entities can be extracted. The distance between different entities, as well as the number of times when a combination of entities appears inside the document corpus is used to calculate a co-occurrence measure. Please note that details about the exact calculation can be found in [6].

A. Realtime Aspects

Since the tool should allow of an interactive exploration, we need an adequate data structure to support our queries. Besides an inverted index for fast retrieval of entities, based on prefixes, the information about the co-occurrence measure must be provided. This is done by precalculating the co-occurrences of all possible combinations. Figure 2 shows the result of this process. Although, this sounds very expensive with respect to space requirements and computational effort, the number of combinations is much smaller than the theoretical maximum value, based on all possible combinations, which was already approved in [6], Accordingly, one can argue that the chosen data structure can also be used for incremental updates.

III. Interactive Exploration

A. Search for Entities

The starting point for an interactive exploration is the selection of an initial entity. To rapidly identify an entity or category, a multi-prefix search is implemented. The search is combined with an auto-suggestion mode, as shown in Figure 3. The disambiguation tool used to identify the entities in the text is AIDA [7], which itself utilizes YAGO [8] as a knowledge base. This means that about 4.3 million entities can be identified. Additionally, about 660 thousand categories are available, which can be selected also.

1) Entity Groups: An entity group is a combination of two and more entities. Semantically, when we search for related entities of an entity group, the combination of all entities in the entity group and a further related entity appear at least

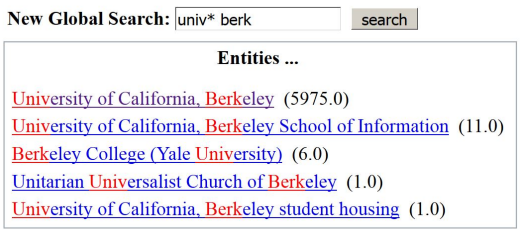

Figure 3. Auto-Suggestion, based on multiple prefixes.

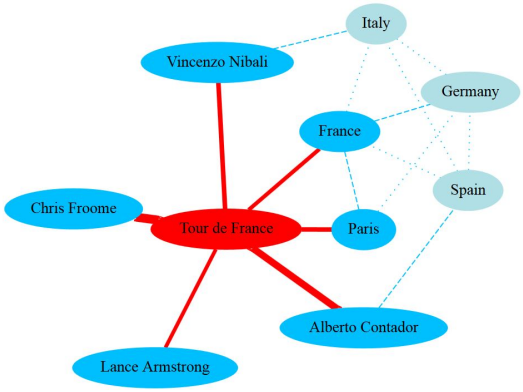

Figure 4. Tour de France with most related entities.

somewhere in the document corpus within k-words. (k represents the window size from Section II). The maximum number of entities in a group is only limited by the precalculation step, where we collect n-tuples of entities. For a given n, groups up to $n-1$ entities can be build. A meaningful value of n depends on the window size. The smaller the window size, the smaller n is, since only a limited number of entities appear within a certain window size. In our current setting the maximum number of entities in a group is 4.

2) Category Taxonomy: Every entity belongs to one or more categories. In our system we use a modified version of the Wikipedia categories. In contrast to the original category system, our approach encompasses a proper tree, where the categories are used to filter related entities.

B. Navigation

After having been chosen, the entity is displayed as a central node in a graph (shown in red). Figure 4 shows this situation around the central entity *Tour de France*. Grouped entities around (shown in dark blue), are the most related n-entities, where n is a choosable parameter between 5 and 50. The widths of the red edges represent the strengths of the relationships. Optionally, a percentile value along the edge, can quantify the relative strength of the relationship compared to the other related entities. In addition, for each of the related entities, another m-entities (light blue) can be displayed (m choosable between 0 and 5). This offers additional information about the context of the central entity.

Hence, you can select every entity in the graph as a next central entity, by simply clicking on it. Alternatively, you can click onto one of the edges, which merges the two related entities into a group and makes them the next central node in the graph. Figure 5 shows the result, after clicking on the edge

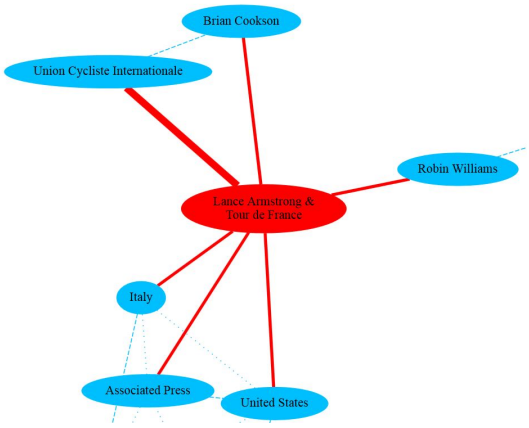

Figure 5. *Tour de France* & *Lance Armstrong* as entity group.

between the entities "*Tour de France*" and "*Lance Armstrong*", merging them into a single node and showing entities around, which are most related to the combination of these two entities.

C. Entity & Type Suggestions

A graph can also be refined, by applying filters to the related entities. Hence, for example, if we want to know which are the most famous passes along the route through France, we start entering the prefix "col" (French word for "pass") in the relationship filter field. While typing, all entities and categories, matching the prefix (or prefixes) are diplayed, making it easy to select the right one or further restrict the actual selection. It must be noted that only entities and categories, which actually are related to the central entity (here: *Tour de France*) are displayed and, thus, no suggestion would lead to a non-existing relationship. Additionally, the order of the entities and categories is context-sensitive to the specified central entity starting with the most relevant entity and category. Figure 6 shows the situation after the three characters "col" have been typed. In this situation, possible related entities, as well as categories are displayed.

At this point, there are three possibilities:

1) If we find a prefix or combination of prefixes covering all relevant entities (i.e., a family name) we can simply press the `refine` button and only the displayed entities will be considered in the graph.
2) Alternatively, we can select a category on the right, so that only entities, which fall into this category (also transitively) are selected. This feature can be applied multiple times.
3) The third option we have is selecting an entity from the left side of the selection box. In this case, the selected entity is added to the actual central entity or entity group and forms an entity group with it. This is the same as clicking on the edge between two entities or an entity and an entity group.

If the user is not interested in entities and categories matching a given prefix but wants to see all possible entities and categories, he simply has to type the asterisk symbol '*' into the search field. Figure 7 shows an extraction of the suggested entities and categories sorted by their relevance. In this way, one can find all related entities, and not only fifty most, as shown in the graph.

 ISBN: 978-1-61208-637-8

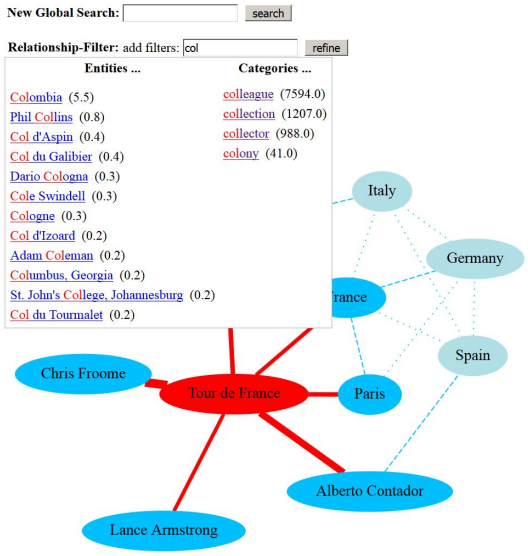

Figure 6. Entity *Tour de France* with related entities and categories, satisfying the given prefixes `col`.

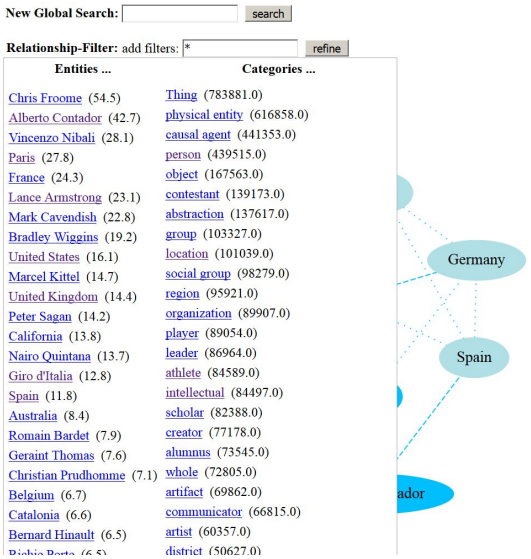

Figure 7. All possible (related) entities and categories for entity *Tour de France*, sorted by their relevance.

IV. Related Work

Our work is comparable with previous works in the field of entity recommendation, particularly which what is reported by Bi et al. [9]. Unlike what was described for our method, the related entities came from a knowledge graph and the click behavior of a user. In our approach, the knowledge is extracted from a document corpus. Schmidt et al. [10] have published a related work where related entities matching a prefix are suggested in the search interface to speed up query formulation (context-sensitive suggestions). This work, on the contrary, explicitly shows relations between entities in a graphical and navigational manner. Indeed, the data structures used are partly the same.

V. Conclusion and Further Work

The paper reported on a system for identifying and analyzing entities and their relationships. The system enables navigation along entity relationships as well as filtering relationships based on prefixes and/or categories. However, calculating the relationships at indexing time makes our system usable for interactive exploration of hidden relationships in a given document corpus. The relationships are automatically extracted from a given text corpus. The system not only considers bidirectional relationships, but also relationships between more entities (the so-called entity groups).

For further work, we intend to extend our interface, so that the documents most relevant to an entity or entity group can be inspected along with the parts of the documents, providing the most needed boost for that entity or entity type. The same can be implemented considering the edges of the graph, which represent the relationships between entities (entity groups).

Our approach of representing the most relevant entities and relationships can be applied to single documents rather than to multiple documents, by simply building a relationship graph for a single document (containing the m most relevant entities and relationships). This graph can potentially be used as a filter for searching for similar documents. In the case presented, the similarity of graphs [11] has to be computed.

References

[1] S. Seufert, K. Berberich, S. J. Bedathur, S. K. Kondreddi, P. Ernst, and G. Weikum, "Espresso: Explaining relationships between entity sets," in Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16), 2016, pp. 1311–1320.

[2] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ser. ACL '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 363–370.

[3] J. Hoffart, "Discovering and disambiguating named entities in text," Ph.D. dissertation, Universität des Saarlandes, Saarbrücken, 2015.

[4] J. Hoffart, D. Milchevski, and G. Weikum, "STICS: searching with strings, things, and cats," in SIGIR 2014, 2014, pp. 1247–1248.

[5] A. Schmidt, D. Kimmig, and M. Dickerhof, "Search and graphical visualization of concepts in document collections using taxonomies," in HICSS 2013, 2013, pp. 1429–1434.

[6] A. Schmidt and S. Scholz, "Quantitative considerations about the semantic relationship of entities in a document corpus," in HICSS 2018, 2018, pp. 933–942.

[7] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum, "AIDA: an online tool for accurate disambiguation of named entities in text and tables," PVLDB, vol. 4, no. 12, 2011, pp. 1450–1453.

[8] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in WWW 2007, 2007, pp. 697–706.

[9] B. Bi, H. Ma, B. P. Hsu, W. Chu, K. Wang, and J. Cho, "Learning to recommend related entities to search users," in WSDM 2015, 2015, pp. 139–148.

[10] A. Schmidt, J. Hoffart, D. Milchevski, and G. Weikum, "Context-sensitive auto-completion for searching with entities and categories," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016, 2016, pp. 1097–1100.

[11] M. Dehmer, F. Emmert-Streib, and J. Kilian, "A similarity measure for graphs with low computational complexity," Appl. Math. Comput., vol. 182, no. 1, Nov. 2006, pp. 447–459.

 ISBN: 978-1-61208-637-8