HiDER: Query-Driven Entity Resolution for Historical Data

Bijan Ranjbar-Sahraei[1], Julia Efremova[2], Hossein Rahmani[1], Toon Calders[3], Karl Tuyls[4], and Gerhard Weiss[1]

[1] Maastricht University, Maastricht, The Netherlands
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
[3] Université Libre de Bruxelles. Brussels, Belgium
[4] University of Liverpool, Liverpool, UK

Abstract. Entity Resolution (ER) refers to the task of finding references in different data sets that refer to the same entity. Cleaning an entire data warehouse and applying ER on it can be a very computationally demanding task, particularly for large data sets which change dynamically. Therefore, a query-driven approach which analyses a small subset of the entire data set and integrates the results in real-time has significant advantage. In this demonstration, we present an interactive tool which allows for query-driven ER in large collections of uncertain dynamic historical data. The inputs of this tool are different sources of historical data such as birth, marriage and death certificates in the form of structured data, and notarial acts such as estate tax and property transfers in the form of free text. The output of this tool are family networks and timelines for historical events visualized in an integrated way. Despite the uncertainties of the input data, coming from spelling errors, name variations and missing data, the extracted entities have high certainty and are enriched by available information, such as place and location of the events, details about relatives, and also facts about the sold/bought properties and the inheritance data.

1 Introduction

In the domain of historical research, large amounts of historical data exists. Digitization and correction of data is an everyday process in historical centers. Additionally, some projects such as Ancestory.com[5] are using crowdsourcing and volunteering efforts to improve the quality of their database on census records and civil registers. This results in many dynamically changing large data corpuses, requiring efficient ER processes.

This work develops, based on the work in [1], a query-driven tool for Historical Data Entity Resolution called HiDER, which has the following advantages: (a) HiDER allows for ER across different data sources; (b) changes in input data and ER algorithms can be incorporated in generating outcomes in real time; (c) by using *Lucene*'s inverted indexing, both structured and unstructured data are handled, and fuzzy search allows for compensating missing data and spelling variations, and (d) graph-based ER allows for detecting and visualizing "family networks".

[5] `http://www.ancestry.com`

2 The HiDER System

The HiDER system is written in Python with Flask web framework, and is developed on an Apache web server, equipped with Solr search platform. HiDER works as follows: A user gives a query which consists of at least a family name, but can also include names of a couple, date and location or other relative names. Subsequently, HiDER searches for relevant records existing in different sources and presents them in an integrated way. To do this, HiDER uses an inverted index data structure to retrieve a subset of records from multiple corpuses, and applies an ER process, developed previously by the authors in [2, 3], on this subset on the fly. As such, the system is flexible in the sense that it adapts with minimal effort to changes in the corpus. HiDER is composed of various modules as shown in Figure 1. Next we introduce each of these modules, in detail.

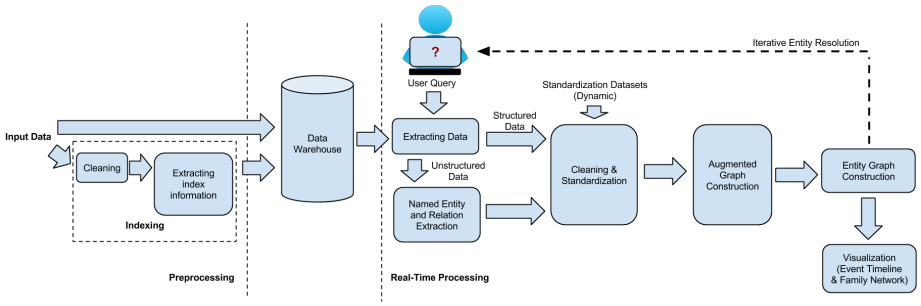

Fig. 1. The HiDER query-driven ER process.

Preprocessing: The **input data** consists of historical documents of 18^{th} and 19^{th} centuries in the form of structured civil registers (i.e., birth, marriage and death certificates) and unstructured notarial acts (e.g., property transfers and estate taxes). We refer to each civil register or notarial act as a *record* and each person mentioned in a record as a *reference*. Upon arrival of a new record or when an existing record is updated, the important information of the record is **cleaned** (by e.g., lower-casing and removing word stems) and stored in an **inverted index**. For structured data, the names, locations, date and type of the record are the indexed information, and a *general text* field is used to generate an inverted index for every term which appears in the record. For the unstructured data the *general text* field is used to generate the inverted index for every term in the text of document. The indexing procedure is computationally light and still captures every information in the record.

Real-Time Processing: Real-Time processing is the main part of HiDER system. Depending on the user query, HiDER uses the available indexes in the data warehouse for **Extracting Data**. The available *faceting feature* guides a user to drill into his/her target data (see left column in Fig. 2). Furthermore, user can choose between strict and fuzzy search, where the latter one allows for compensation of spelling errors and missing data. The retrieved unstructured

data is then further processed for **Named Entity** and **Relation Extraction**; for more information refer to [2]. Extra **Cleaning** and **Standardization** is applied to the outputs of previous modules. For instance, extra symbols are removed from names, and names with spelling variations are standardized. The standardization databases[6] are continuously updated based on user feedback and experts knowledge, and any update can easily be incorporated in answering future queries. In the **Augmented Graph Construction** phase, the contextual information available in each record is translated to a graph component. The graph consisting of these components is then augmented by adding so called *block nodes*, which capture the important features of each name such as its first and last few letters and its length (see [3] for examples). Once the augmented graph is constructed, a *random walk*-based entity detection approach is used to detect all references with highly similar contextual information (i.e., similar neighbor nodes in the augmented graph), indicating that they all refer to the same entity. Once the entities are detected, the **Entity Graph Construction** is accomplished by merging each set of references that correspond to the same entity (this technique is elaborated by the authors in [3]).

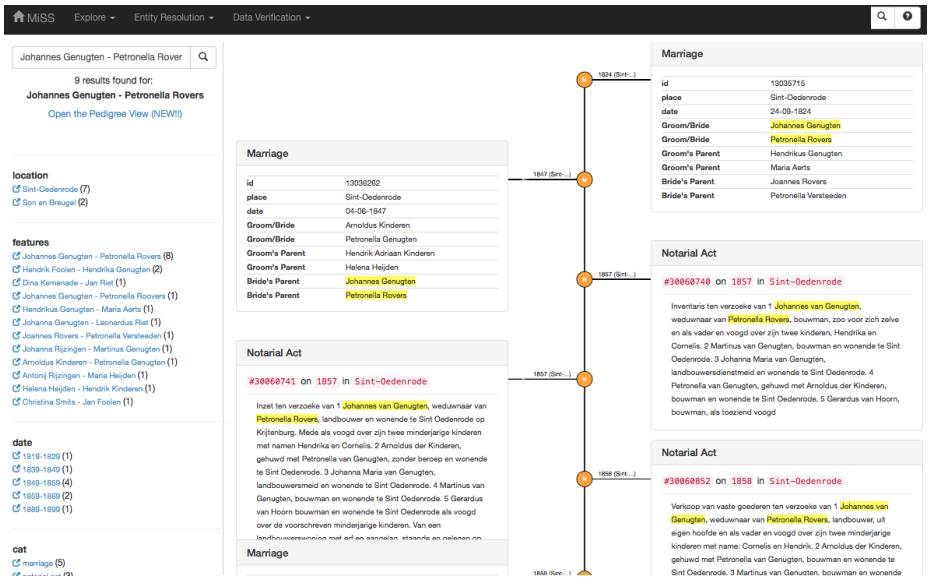

Fig. 2. HiDER interface upon arrival of a query: Searching tool and faceting are shown on left, and the event timeline is shown or right.

Visualization serves as an indispensable tool to evaluate the entity graph manually, and is also a way to deliver the results to the user. HiDER is capable of visualizing the entity graph in the form of *event timelines*. In event timeline the information of each record is shown in the form of a floating card, while the

[6] e.g., `http://www.meertens.knaw.nl/cms/en/collections/databases`

important entities are highlighted, and the cards are sorted based on the date of the records (Fig. 2). To visualize the *family networks*, due to complexity of the generated entity graphs, we propose a novel visualization scheme for genealogical data by combining every two individuals with marriage relations into single couple nodes, and use graph traversing algorithms to categorize nodes into different generations. The family network of a sample query is shown in Fig. 3. Users can interactively focus on nodes, expand them or see the corresponding timeline.

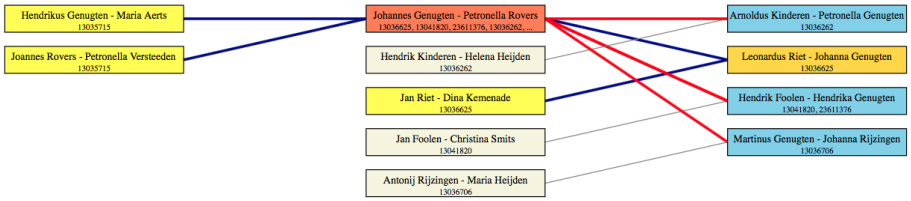

Fig. 3. HiDER visualization of a family network: each link connects parents, on the left, to a child and his/her spouse, on the right.

Last but not least, HiDER allows for **Iterative ER**, which allows for using the entity graph constructed in one round for extending the current query and iteratively constructing new entity graphs. This allows for extending the family network to farther relatives of specific entities, and also helps the user to manually compensate some of the existing uncertainties.

3 Conclusions and Future Work

HiDER interactive tool targets different experts including data scientists, genealogists and demographers. Any individual who is interested in generating his/her family tree is among the main audience of HiDER, too. According to evaluations by experts of BHIC[7], using HiDER for searching the available 3,000,000 documents generates precise results (e.g., the precision of ER in [3] is 92%). Compared with the available searching platforms in many historical centers which need days or weeks to retrieve information about a specific family, HiDER can perform the same task in a few seconds, thus opening a very promising venue for historians.

References

1. Hotham Altwaijry, Dmitri V Kalashnikov, and Sharad Mehrotra. Query-driven approach to entity resolution. *Proceedings of the VLDB Endowment*, 6(14):1846–1857, 2013.
2. Julia Efremova, Bijan Ranjbar-Sahraei, Hossein Rahmani, Frans A Oliehoek, Toon Calders, and Karl Tuyls. Multi-source entity resolution for genealogical data. In *Population Reconstruction*. Springer, 2015 (in press).
3. Hossein Rahmani, Bijan Ranjbar-Sahraei, Gerhard Weiss, and Karl Tuyls. Entity resolution in disjoint graphs: an application on genealogical data. *Intelligent Data Analysis*, 20(2), 2016 (in press).

[7] Brabant Historical Information Center `https://www.bhic.nl`.