A Prosopographical Information System (APIS)

Matthias Schlögl, Katalin Lejtovicz

Austrian Centre for Digital Humanities
Sonnenfelsgasse 19, 1010 Vienna, Austria
matthias.schloegl@oeaw.ac.at, katalin.lejtovicz@oeaw.ac.at

Abstract

During recent years massive amount of biographical datasets have been digitized and - at least some of them - made available open access. However, an easy to use system that allows non-experts to work with the data is still missing. The APIS system, designed within the framework of the APIS project at the Austrian Academy of Sciences, is a web-based, highly customizeable virtual research environment that allows researchers to work alongside programs designed for processing natural language texts, so called Natural Language Processing pipelines.

Keywords: biographical data, virtual research environment, natural language processing

1 Introduction

During recent years massive amounts of biographical datasets have been digitized and - at least some of them - made available open access (Reinert et al., 2015; Fokkens et al., 2014). Additionally, collaborative efforts such as Wikipedia/Wikidata[1] have created even more partly structured prosopographical and biographical datasets (Gergaud et al., 2016). Reference resources such as Gemeinsame Normdatei[2] and the Virtual Internationa Authority File (VIAF)[3] have also been utilized for prosopographical research (Andert et al., 2014). Since the first endeavours, researchers have worked on tools that allow for extracting structured data of these biographical texts. Various Natural Language Processing (NLP) techniques have been used for these objectives (local grammars, regular expressions, machine learning and deep learning based approaches etc.). However, the goal of the researchers was not limited to transforming full-text data into structured data, but also included the interpretation of textual resources by applying statistical and network research methodologies. In this sense computer linguistic processing, statistical analysis and network visualization of biographies has been started at ÖBL - the Austrian Biographical Dictionary - in the context of the APIS project. The results of the various analysis methods are later evaluated and interpreted by scholarly researchers. In this paper we describe the Virtual Research Environment (VRE) (Schlögl and Andorfer, 2018) - from now on referred to as APIS - that has been developed during the project and Natural Language Processing (NLP) techniques we use for (semi)automatically structuring the data. The APIS VRE is a Django based web application published under a open-source license (MIT) on GitHub: https://github.com/acdh-oeaw/apis.

[1] or resources such as Freebase that have been included in these endeavours.

[2] An authority file for persons, events, locations, works, institutions operated cooperatively by the German National Library, the German Union Catalogue of Serials and other institutions. The GND has recognized these developments and will open the system to actors outside traditional libraries. http://www.dnb.de/EN/Standardisierung/GND/gnd_node.html (Kett, 2017)

[3] An international authority file compiled by national libraries. https://viaf.org/.

2 APIS virtual research environment

The approaches for extracting structured information from biographical data sets have been brought forward by a relatively small scholarly community using locally runned, tailor made systems that almost never have a user interface. Compared to the conventional methods that researchers apply when evaluating textual data (e.g. taking notes in a Word Document, filling out an Excel sheet manually), APIS allows for a semi-automatic exploration of the information in a large scale data set. It enables researchers to find answers to their research questions more easily and much faster than with conventional methods.

APIS is a web-based, highly customizeable VRE that allows traditional researchers to work alongside NLP pipelines. This hybrid approach (the possibility to manually annotate texts and edit entities/relations alongside automatic systems) allows researchers to "use the best of both worlds", and computer scientists to improve the tools directly on real world data. The web application not only helps researchers to systematically and semi-automatically process large amounts of data, but also to analyze and visualize connections between entities detected in the documents. Visualization of the data allows the researchers to get an overall picture of the entities and relations encoded in the documents, that otherwise would be hard to access. APIS provides the users an easy and intuitive workflow to process large amounts of data.

It therefore tackles two main problems and will make the work with biographical data easier for historians as well as data scientists:

- It allows historians to annotate biographies with exactly that information they need for their research, easily link the annotations to the Linked Open Data cloud[4], and export it for further research.

[4] LOD - data that is being published so that it can easily be interlinked with other datasets, which allows for more refined, detailed queries of the content.

- It allows data scientists to easily access annotated data via APIs, use it for (re)training models, store new annotations to the system and use the built-in evaluation system for retrieving precision, recall, F1 and other metrices.

2.1 General Idea

The design of APIS meets three basic criteria, based on experience from previous projects:

- a simple datamodel that can be serialized to other formats and datamodels later on
- use of a solid and widely used software stack to keep the development and maintenance effort as low as possible
- a hybrid approach that allows researchers as well as automatic tools/pipelines to work in parallel on the same dataset.

While this design has some advantages, it brings some downsides along. Most commonly used high level ontologies, such as CIDOC CRM (a structure designed to describe concepts and relationships used in the cultural heritage domain) are based on an event driven datamodel. Our internal datamodel (discussed in more detail below) is simpler and easier to use, but needs to be mapped to event based models later on. Similarly, the use of well proven technologies such as Django and SQL databases brings some obvious advantages in the development of the web application, but in a world of Linked Open Data at some point we will need to serialize our data into RDF-triples[5] and publish it to include it in the Linked Open Data cloud. However, during the project our design decisions have proven to be successful. Due to the simple datamodel and the easy and fast development of the web application we were able to (manually) annotate much more data than we anticipated.

2.2 Datamodel

The APIS datamodel is a hybrid between an event-based and a relation-based model. Figure 1 shows a simplified

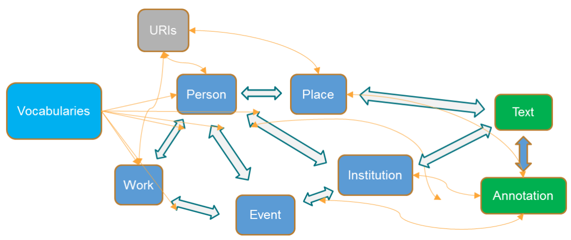

Figure 1: APIS datamodel (simplified version)

version of the APIS datamodel. It consists of 5 entities (person, place, institution, event and work) that are all interrelated. Relations can be added between persons and places, persons and institutions, institutions and works, persons and persons and so on. All entities share a set of basic attributes (name, start-, end-dates etc.) and some have additional ones (e.g. place has longitudes and latitudes). Every entity can be related to several URIs (if they do not share the same top-level domain) and grouped in so-called collections. Relations on the other hand have a fix set of attributes (start-, end-date, kind, notes, references).[6] Every entity can have as much full-texts as needed. These full-texts in return can have offset annotations grouped in so-called annotation projects and - if useful - linked to other entities or relations[7] in the database. All entities and relations are typed with Simple Knowledge Organisation System (SKOS) vocabularies (SKOS defines standards for working with knowledge systems such as thesauri, taxonomies and classification schemes). Additionally the system features a very fine grained user permission system, that allows to set permissions on collection basis.

2.3 The web frontend

The APIS web frontend allows to search the data, work on it and analyze it. The list views can be used to search the data[8], sort and export it and to access the edit views. Figure

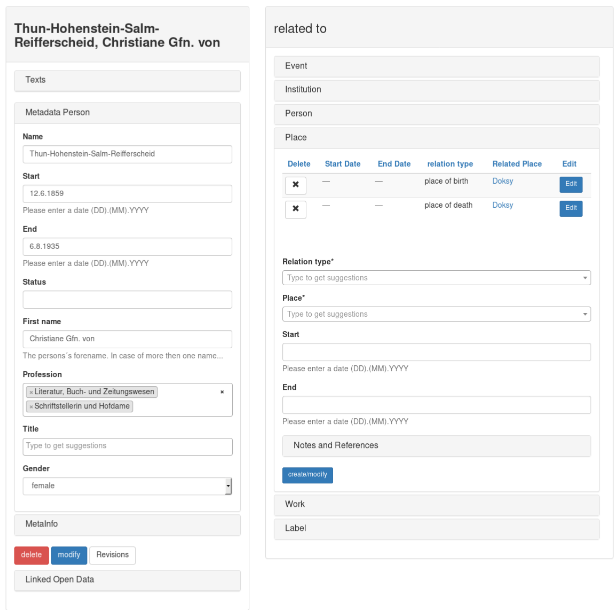

Figure 2: APIS edit view of person

2 shows the edit view of a person. The view consists of two panes, in the left pane one can work on the entities metadata, in the right the entity can be related to other entities. The forms feature wherever possible/useful autocompletes

[5]RDF is a framework for representing information in the Web. In RDF statements about resources are expressed in the form of subjectpredicateobject, known as triples.

[6]Relations are a kind of mini-event: The relation can be only connected to two entities and has a limited set of attributes, but nonetheless the relation of two entities has some additional data attached. We therefore call our model a hybrid between relation-based and event-based.

[7]Entities and/or relations that are annotated in the full-text can be automatically added to the database.

[8]The search fields and functions can be defined in the main settings file of the application.

that make the editing process more convenient and less error prone for the researcher.

2.4 Full-text annotation

APIS also allows for annotation of biographical full texts. Instead of just adding a relation between two entities to the database, this relation can be annotated directly in the text. When highlighting a part of the text a context menu opens that allows to select the relation type.[9] After selecting the relation type (e.g. Person-Place) another form is loaded that allows for selecting the related entity (e.g. Vienna) and the kind of relation (e.g. 'educated in'). We already explained

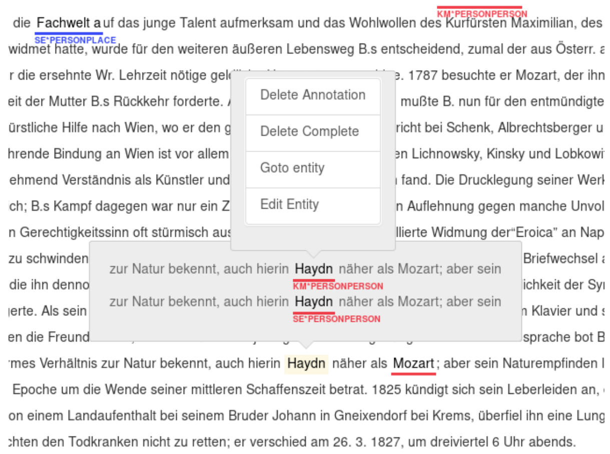

Figure 3: Image of overlapping annotations

that annotations in APIS are stored as offsets and related to the user and something we call annotation project. This allows to view the biography from different angles. A simple form allows to filter for the annotations one wants to look at (annotation project, user, type of annotation). Additionally, the visualization allows for overlapping annotations. As figure 3 shows when clicking on overlapping annotations - visualized with yellow background color - a context window opens and shows a copy of the text snipped for every existing overlapping annotation.

2.4.1 Automatic import of LOD entities

The APIS webapplication allows the use of external resources - such as Linked Open Data resources - in the autocomplete search. Whenever a researcher searches for an entity in the autocomplete, not only local entries are searched, but also external resources integrated into the APIS system.[10] When a researcher selects an entity that is not yet present in the database the system retrieves the original entity and parses it into the database. The parser can be defined in an instance wide settings file.

2.5 Inter annotator agreement

In section 3 we will elaborate on the Natural Language Processing (NLP) techniques we used to (semi)automatically enrich the ÖBL biographies. One of the prerequisites of automatic text processing is a gold standard of annotations and a high inter-annotator agreement.[11] Getting towards a gold standard and a high agreement among the annotators is a time consuming and tedious process. We try to foster this process by visualizing overlapping annotations in the frontend and providing readymade metrices to compute the agreement over large collections of texts and/or annotators.[12]

2.6 Versioning

One important aspect of (historic) research is provenance. Ideally every step in the data generation and data analysis process is logged and reproduceable. To allow for full provenance information in the APIS process, we implemented a system that serializes every edit of a data point and adds a timestamp and a user-ID to the serialization. The revision can be accessed in the GUI and used for recreating any former state of the database. We are currently working on building a Rest API endpoint for providing machine readable access to this versioning system.

2.7 Visualization

The APIS system also includes a rudimentary visualization module. Several projects have shown that social network analysis (SNA) is a very useful visualization and analysis method (Armitage, 2016; Warren et al., 2016) The

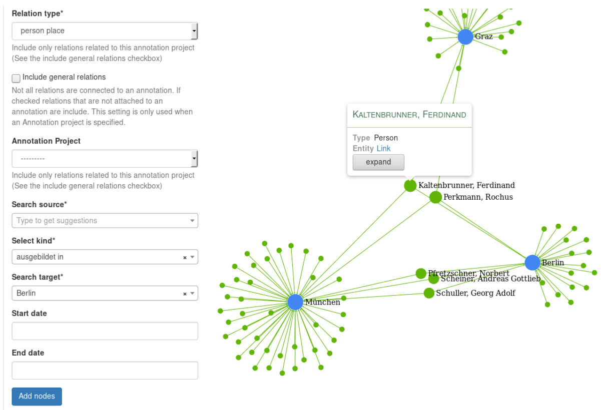

Figure 4: Network visualization

APIS network visualization allows for iterative creation of networks by specifying the source node[13], the relation type and/or kind and/or the target node. The form supports the researcher in creating the network with autocompletes that show existing entries in the database. Nodes

[9]The context menu is defined in a system wide settings file accessible via the admin backend.

[10]We use a local Apache Stanbol instance for fast access to Geonames and GND, but have also implemented bridges to SPARQL (the query language for RDF data) endpoints for less frequently used sources.

[11]Most of the time the latter is needed to produce the former.

[12]As described above the APIS application does not distinguish between human researchers and automatic tools. Tools communicate with the database via a Rest API, researchers via the GUI, both have an user account that allows APIS to version the edits.

[13]It is also possible to select whole collections of nodes.

can be extended[14] by accessing the context menu of the nodes. Figure 4 shows a network that was created by adding person-place relations with the target node set to 'München', 'Berlin' and 'Graz'. After creating the network it can be downloaded either as JSON[15] or graphml.[16] The downloaded file includes all the attributes - such as longitudes and latitudes for places - that exist in the database. The APIS project also cooperates with external partners to explore the potential of other more experimental visualization methods. One of these methods is the space-time-cube developed by colleagues from the University of Krems (Windhager et al., 2017).[17]

3 Information extraction

One of the goals of the APIS project is to offer automated text processing to facilitate the work of researchers. The processing and interpretation of the texts were carried out using computer linguistic methods, which include identification of entities (individuals, places, institutions, etc.), automatically linking them to Linked Open Data Cloud resources, and disambiguating and manually curating the results. In the following section we will outline the above described steps in more detail.

3.1 Entity Linking

Although biographies are available in XML format, these do not contain all relevant information about a person's life in structured format, except for some key events such as birth and death. One of the main goals of the project is to reveal information encoded in natural language text (e.g. names of persons, places, institutions, events, etc.) and to automatically detect relationships between them and the person depicted in the biography. In order to tackle this problem efficiently, we combined automated and manual information retrieval techniques. The information extraction in APIS consists of three main steps: Named Entity Recognition, Entity Linking, and Disambiguation/Curation. For the automatic information extraction we use the open source software Apache Stanbol[18], which detects entities in natural language texts and connects them to ontologies and knowledge databases such as the GND, GeoNames[19], or DBpedia[20]. The connections that are created between entities and biographies not only allow for the enrichment of the biographies with semantic information, but also for the automatic correction of missing or erroneous data. The advantage of using Apache Stanbol for Entity Identification and Linking is that it provides a straightforward mechanism how entities are identified and how any ontology in RDF/XML format can be converted into a semantic reference resource, which is later used for the semantic enrichment of the documents. To perform the semantic annotation, we produce so-called Referenced Sites from the data available in RDF/XML format (i.e. from GeoNames, GND). In the Referenced Sites the indexed data is stored in a Solr [21] index.

3.1.1 Abbreviations

The information extraction process created in APIS consists of two steps. First, we resolve abbreviations of person names, institution names, academic titles, place names, and common verbs. We developed two versions to resolve abbreviations, a Java program based on regular expressions and a Python based script that uses regular expressions, a dictionary of German words and a large German-language corpus (AMC) (Ďurčo et al., 2014) to resolve ambiguous abbreviations and choose the correct variant. The program queries the abbreviation and its context in the AMC corpus, and the resolution with the most hits is chosen.

3.1.2 Creating an index

The second step in the semantic annotation process is to create Solr indices from ontologies. During Entity Linking Apache Stanbol searches the entities (persons, places, institution names, etc.) in the indexed ontologies. In the APIS project we created indexes from GeoNames and GND to link the place names, personal names and institution names in the text to the Linked Open Data Cloud. The indexes were created as follows: we downloaded the RDF/XML dumps of the aforementioned resources, which were cut into smaller files in order to get manageable sized data, and to make it easy to create separate indexes for the different entity types. After this we created the Apache Solr indexes from the above mentioned files using Apache Stanbols Java package for indexing.

3.1.3 The NLP pipeline

After creating and installing the Solr index the Entity Linking component is configured. Stanbol allows various configuration options to achieve an accurate and efficient Entity Linking process. For example, one can narrow down the search to proper nouns only. In this case the NLP algorithm of Stanbol identifies proper nouns and queries only them in the Solr index, this yields more accurate Entity Linking and a better runtime. Another configuration option is to use the types of entities in the matching process. If this setting is turned on and the index contains information regarding the type of the entities, the user gets the results categorized into different types such as "Person", "Location", "Event", etc. (depending on what types are available in the index).
Following the configuration of the Entity Linking component, the Natural Language Processing component is constructed, which defines what NLP steps have to be carried out. In APIS we use the Apache OpenNLP[22] open source software for the computer linguistic analysis of the biographies. Our pipeline consists of the following steps: Determine the language of the input text. (langdetect) Divide

[14]By 'extending' we mean adding all relations for the node to the visualization.

[15]a format that allows for easy data interchange between applications - see: https://www.json.org/

[16]Graphml is a XML-based format for storing graphs. See http://graphml.graphdrawing.org/ for details.

[17]Please also see Windhager et al in this proceedings for details.

[18]https://stanbol.apache.org/index.html, last accessed: 26.02.2018

[19]http://www.geonames.org/, last accessed: 26.02.2018

[20]http://wiki.dbpedia.org/, last accessed: 26.02.2018

[21]Solr is an open source search platform, which allows for full-text search, faceted search, hit highlighting amongst other features.

[22]https://opennlp.apache.org/, last accessed: 27.02.2018

the text into sentences (opennlp-sentence). Tokenize the sentences (opennlp-token). Determine the Part of Speech tag of the words (opennlp-pos). Search for noun phrases (opennlp-chunker). Perform Entity Linking. (Custom Referenced Site)

In the last step, the nouns and noun phrases are compared with the Solr index (Entity Linking). If a term matches an entry in the index, the entry from the Solr index is returned by the application in the requested output format (e.g. JSON, RDF/XML, Turtle, N3, JSON-LD). If there are multiple results, a score between 0 and 1 indicates which is the most likely result. The advantage of the Apache Stanbol Entity Linking software is, that it can effectively index any ontology available in RDF/XML format, and allows the user to select the data resource for semantic annotation.

3.2 Relation Extraction

Entity Linking is the first step in automatically interpreting the meaning of a natural language document. Through Entity Linking strings in the documents can be replaced by URIs (Uniform Resource Identifiers). The concepts in the LOD resources are not only clearly identifiable and referenceable by their URIs, but they can also be shared between applications, unstructured texts can be enriched with information attached to them or inconsistencies in the data can be detected and corrected.

The second step is to determine the relationships and the types of the relationships that hold between the entities, also known as automatic Relation Extraction. During Relation Extraction the NLP module looks for semantic relationships such as 'parent-child', 'traveled to a place', 'learned somewhere', 'participated in an event' between people, places, and events detected in the text. We have tried three different methods for the automatic relationship recognition, which will be tested and the best solution will be permanently integrated into the APIS system.

The first version is a rule-based algorithm implemented using the GATE framework.[23] The implementation uses the JAPE regular expressions language of GATE to automatically extract semantic links from the text. In a first step, the output of the Entity Linking module is converted to XML format, where each Named Entity is an element in the XML. These XML files were then uploaded to GATE, and processed by the ANNIE NLP module.[24] The Entity Linking results as well as the output of the NLP pipeline are stored as annotations in GATE. The JAPE regular expressions work with these annotations and search for linguistic patterns in the documents that can express a relationship. If the application finds a text snippet that corresponds to the pattern that is specific to that relationship, it automatically provides a new annotation, which defines the type of the relationship. The output of the relation extraction was exported to XML - widely used in NLP applications - and imported back in the APIS system.

[23]GATE is an open source software designed to automatically process natural language documents. See: https://gate.ac.uk/

[24]ANNIE is a system within the GATE framework, which was designed to automatically process and extract information from textual data.

The second solution we tested was IEPY (Information Extraction in Python)[25], an open source software implemented in Python which realizes relation extraction. IEPY performs machine learning based relationship recognition. On the web interface of the application, the user annotates occurrences of predefined relationships (e.g. 'traveled somewhere', 'married somebody', etc.) from which the software learns a model, that can be used to identify relations in documents that have not been seen before by the system. In case of the ÖBL, IEPY has not proven to be a suitable software, because it requires the selection of both members of a relationship (eg. in case of 'learned somewhere' both the person and the place). However in ÖBL, to avoid the repetition of the person, the biography was written about, his/her name is usually only mentioned once, at the beginning of the biography.

The third approach we have examined is the recognition of the tree structure obtained from the syntactic parsing of the sentences with Deep Learning. We use a standard NLP pipeline[26] to process the text. When the module finds a named entity it climbs up the parse tree and extracts predefined classes - in the sense of POS tags - of words (e.g. verbs). The extracted list of words is converted into a vector which is used for classification. This method makes use of the inherent advantages a biography brings along: in many cases a biography talks about the portrayed person, therefore we skipped the search for the subject and just assumed that the portrayed person is the subject. First tests with a model trained on roughly 4000 and evaluated on 1000 examples of person-place relations shows the potential of the method[27], but also the problems automatic tools have with the very specific language in the ÖBL.

The training data set was annotated during a small research project dealing with members of the 'Künstlerhaus'.[28] Given the rather difficult training data, the (for modern NLP tools) problematic language of the ÖBL, and the relation types to extract[29] the model performed rather well, even though obviously not precise enough for historians to only rely on the extracted data. The evaluation on 30 randomly chosen artist biographies[30] showed a recall of 0.79 and a precision of 0.44 (F-beta 0.56). The combination of high recall and low precision is due to the named entity recognizer annotating places where a human annotator wouldn't do so (e.g. 'Vienna' in 'University of Vienna'). We believe that the precision of the method can be significantly raised by improving the named entity recognizer.[31]

[25]https://github.com/machinalis/iepy

[26]https://spacy.io

[27]Please see https://apis.acdh.oeaw.ac.at/presentation_innsbruck17/ for a more detailed presentation and a live version of the model.

[28]The fact that this data was not specifically produced for training purposes is important. It is very unevenly distributed: about 2/3 of all annotations bear only two labels out of eight. The annotations were also done by only one annotator and are therefore not very concise over the whole corpus.

[29]Relation types were only chosen based on the research question and not for how easy they are to find by automatic tools.

[30]All members of the 'Künstlerhaus' have been annotated and used for training, we therefore used other artists for evaluation.

[31]We will do so by retraining the model, and by implementing

There have not been many attempts to automatically extract information from biographical articles so far and no one - to our best knowledge - has tried to train models on relations annotated by researchers. However, Fokkens et al. (2014) for example extract metadata on the portrayed person from full text. While this is not (exactly) the same as extracting relations to other entities it is comparable (e.g. metadata on education vs relations to schools and universities). Fokkens et al. (2014) had much higher precision, but significantly lower recall. The overall system performed similar to our deep learning approach. Dib et al. (2015) used a somewhat similar approach to extract professions from wikipedia articles. While they also used the parse tree (and especially the verbs) to find the connection between an actor (in our case the portrayed person) and a circumstance (in our case a Named Entity) they did not use a machine learning algorithm to predict the kind of relation, but used a (more or less) fixed set of words that describe the professions. Even if they have evaluated it only on a limited number of well suited articles the overall performance of their system was much higher than ours (recall: 74.1%, precision: 95.2% and F1: 83.3%). However, as it is focused on extracting professions only the system is not really comparable to ours. Bonch-Osmolovskaya and Kolbasov (2015) also used rules to extract facts from a digital edition of Tolstoy's letters. While the system had a very good performance (comparable to Dib et al. (2015)) for professions it had a F1 of 0.43 for family facts.

We are currently working on annotating 300 biographies specifically for training the relation extraction tools. While our training material so far focused on certain professions and on a specific research question, the model trained on these annotations should provide us with a baseline. Additionally, we are working on a gold standard for evaluating this baseline model.

We are also working on evaluating the rule based approach for relation extraction discussed above.

4 Conclusion

APIS provides a integrated system that allows researchers to annotate biographies and link the annotation to LOD resources (and therefore reuse the data that already exists). In a second step it allows for basic visualizations, filtering and export of the data. On the other hand the system provides easy access to the database-backend for data scientists and therefore allows for use of annotations for training models and out of the box evaluation.

The NLP pipelines have some problems with the non-standard language used in biographic dictionaries such as ÖBL. However, we found that the rule based approach as well as the trained models show some possibilities. The former - as others have shown before (Dib et al., 2015; Bonch-Osmolovskaya and Kolbasov, 2015) - especially for extracting data of well defined realms such as professions. The latter even if precision and recall are not high enough yet, to provide historians at least with a useful baseline annotation that they can use as starting point. This tool will - other than the rule based approach - allow historians to train it with whatever they are interested in and get a first - even if not very accurate - annotation of the whole dataset.

some simple rules such as: when the name of an institution contains a place name, the system will annotate the expression as an institution, but not as a place.

5 Copyrights

6 Acknowledgements

The APIS project is funded by a research grant (project number ÖAW0405) of Nationalstiftung für Forschung, Technologie und Entwicklung (Programm "Digital Humanities - Langzeitprojekte zum kulturellen Erbe")

7 References

Martin Andert, Frank Berger, Paul Molitor, and Jörg Ritter. 2014. An optimized platform for capturing metadata of historical correspondence. 30(4):471–480.

Neil Armitage. 2016. The Biographical Network Method. *Sociological Research Online*, 21(2):16.

Anastasia Bonch-Osmolovskaya and Matvey Kolbasov. 2015. Tolstoy Digital: Mining Biographical Data in Literary Heritage Editions. In *Proceedings of the First Conference on Biographical Data in a Digital World 2015*, page 5, Amsterdam.

Firas Dib, Simon Lindberg, and Pierre Nugues. 2015. Extraction of Career Proles from Wikipedia. In *Proceedings of the First Conference on Biographical Data in a Digital World 2015*, page 6, Amsterdam.

Antske Fokkens, Serge ter Braake, Niels Ockeloen, Piek Vossen, Susan Legne, and Guus Schreiber. 2014. BiographyNet: Methodological Issues when NLP supports historical research. pages 3728–3735.

Olivier Gergaud, Morgane Laouenan, and Etienne Wasmer. 2016. A Brief History of Human Time: Exploring a database of 'notable people'. Sciences Po Economics Discussion Paper 2016-03, Sciences Po Departement of Economics, February.

Jürgen Kett. 2017. GND-Entwicklungsprogramm 2017-2021.

M Reinert, M Schrott, and B Ebneth. 2015. From Biographies to Data Curation-The Making of www. deutsche-biographie. de. *BD*.

Matthias Schlögl and Peter Andorfer. 2018. acdh-oeaw/apis-core: Apis-core, May.

Matej Ďurčo, Karlheinz Mörth, Hannes Pirker, and Jutta Ransmayer. 2014. Austrian Media Corpus 2.0.

CN Warren, D Shore, J Otis, and L Wang. 2016. Six Degrees of Francis Bacon: A Statistical Method for Reconstructing Large Historical Social Networks. *DHQ*, 10(3).

Florian Windhager, Paolo Federico, Saminu Salisu, Matthias Schlögl, and Eva Mayr. 2017. *A Synoptic Visualization Framework for the Multi-Perspective Study of Biography and Prosopography Data*. October.