Large-scale Data Harvesting for Biographical Data

Alistair Plum*, Marcos Zampieri*, Constantin Orăsan*, Eveline Wandl-Vogt†, Ruslan Mitkov*

*Research Group in Computational Linguistics
University of Wolverhampton, UK
{a.j.plum, m.zampieri, c.orasan, r.mitkov}@wlv.ac.uk

†Austrian Centre for Digital Humanities
Austrian Academy of Sciences, Austria
eveline.wandl-vogt@oeaw.ac.at

Abstract

This paper explores automatic methods to identify relevant biography candidates in large databases, and extract biographical information from encyclopedia entries and databases. In this work, relevant candidates are defined as people who have made an impact in a certain country or region within a pre-defined time frame. We investigate the case of people who had an impact in the Republic of Austria and died between 1951 and 2019. We use Wikipedia and Wikidata as data sources and compare the performance of our information extraction methods on these two databases. We demonstrate the usefulness of a natural language processing pipeline to identify suitable biography candidates and, in a second stage, extract relevant information about them. Even though they are considered by many as an identical resource, our results show that the data from Wikipedia and Wikidata differs in some cases and they can be used in a complementary way providing more data for the compilation of biographies.

Keywords: Austria, Biographies, Information Extraction, Natural Language Processing, Wikidata, Wikipedia

1. Introduction

In the last decade, large biographical databases have become available in a number of languages (Reinert and Ebneth, 2017). This is the case with many online data sources and sources which were digitised such as the Slovenian Biography (Erjavec et al., 2015), the Deutsche Biographie (Reinert et al., 2015), and the Österreichisches Biographisches Lexikon (der Wissenschaften, 2012). As a result of the large amounts of data available, researchers have been exploring ways to process this data using computational methods. In particular, natural language processing (NLP) and information extraction (IE) methods play an important role in processing these large amounts of data, ranging from tasks like tokenisation, part of speech tagging and sentence splitting, to toponym resolution, semantic role-labeling and relation extraction. Due to their size and availability, Wikipedia[1] and Wikidata[2] have become popular online data sources of information for biographies (Biadsy et al., 2008; Chisholm et al., 2017). In addition, DBPedia[3] provides structured information that have been used to generate biography summaries using natural language generation methods (Moussallem et al., 2018). A number of projects such as the *A Prosopographical Information System* (APIS) project at the Austrian Academy of Sciences (AAS) (Schlögl and Lejtovicz, 2017) and the Dutch BiographyNet project (Fokkens et al., 2014) have addressed the problem of retrieval of information from biographical encyclopedias and dictionaries.

The aforementioned APIS project aims to develop new methods for re-using qualitative (biographical) research products (encyclopedias) for quantitative research and, in doing so, facilitate a digital transformation process against the background of Humanities (Gruber and Wandl-Vogt, 2017). To achieve this, the project has developed a web-based, customisable virtual research environment that allows researchers to work with programs especially designed for processing biographical texts. Another goal of the APIS project is to reveal information encoded in texts such as people names, institutions, places, and to detect relationships between them and the person depicted in the biography, primarily in relation to the *Österreichisches Biographisches Lexikon* (ÖBL) project, by which APIS is third-party funded. In this context, the aim is to collect and make visible the lives and careers of persons with impact in the area of the former Austrian-Hungarian monarchy, as well as the first Republic of Austria. In order to find relevant candidates, ÖBL is aimed at looking beyond the primarily usual suspects, in order to find lesser known and less easy to find knowledge carriers, influencers and impact holders. Currently, about 18,500 biographies are available and ÖBL aims to publish the final volumes in 2020. Since 2004, ÖBL went digital and a database has been established to support the manifold editorial processes. A rich network to neighbouring analogue endeavours and close personal relationships exist, for instance the European Biography-Portal (Gruber and Wandl-Vogt, 2017).[4]

In this paper, we present a work-in-progress processing pipeline which can be used to identify relevant biography candidates in Wikidata and Wikipedia, and to extract information about these candidates. The work presented here is within the scope of the aforementioned APIS project and

[1] https://www.wikipedia.org/

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page

[3] https://wiki.dbpedia.org/

[4] https://www.biographie-portal.eu/

aimed at enriching the ÖBL, by means of automatically suggesting candidates for inclusion in the newer additions of the ÖBL. We investigate the case of biography candidates that died between 1951 and 2019 and who had an impact in the present Republic of Austria. The current work focuses on processing the English texts, which can easily be adapted to process other languages as well, and selecting relevant candidates for inclusion in the ÖBL. However, since previous work on adding biographies to the ÖBL was carried out by a team of historians, with decisions sometimes being based on factors which are difficult to model, evaluation of our pipeline is not straight-forward. We discuss this aspect in Section 4.

2. Related Work

The field of *information extraction* (IE) has long been a thriving area of research within *natural language processing* (NLP) and has steadily maintained a close relationship with gathering information from the web. A large amount of the early work on IE was done during the Message Understanding Conferences (MUC) which took place from 1987 until 1997, and was later continued at the Automatic Content Extraction (ACE) program (Grishman and Sundheim, 1996; Hirschman, 1998; Doddington et al., 2004). These well funded conferences laid the ground work for extracting entities and information pertaining to the same from text, audio and image data. Although the earliest approaches were rule-based (Chinchor et al., 1993), machine learning-based contributions quickly followed suit. Most notably these were made by Freitag (1998) and Soderland (1999), who both emphasised that their approaches work on many different types of text, including HTML and unstructured text, most commonly found on the web.

Modern approaches in IE are usually machine learning-based approaches, and much focus has been put on an area of IE called *open information extraction* (OIE). These modern methods require little to no human supervision and are focused on gathering information from web sources (Del Corro and Gemulla, 2013). One of the first OIE systems was TextRunner, presented by Yates et al. (2007). Another system that gained wide-spread attention is ClausIE, which made use of syntactic knowledge in English (Del Corro and Gemulla, 2013). The most recent approach that relies heavily on machine learning has been presented by Stanovsky et al. (2018). This approach, although supervised, makes use of state-of-the-art machine learning architectures as well as semantic role labelling.

Other areas related to IE are *named entity recognition* (NER), *named entity linking* (NEL) and *wikification*. NER refers to the NLP task of detecting entities in text, including (proper) names, locations, institutions, dates and so on. On the other hand, NEL refers to the process of linking named entities to entries in large databases or knowledge bases, essentially linking information together. Both Petram et al. (2015) and Brouwer and Nijboer (2017) have explored entity linking in the context of biographical data, in order to gain further information on persons. Hachey et al. (2013) implement and evaluate three existing NEL approaches, while making use of Wikipedia to augment their approach. However, this should not be confused with wikification, which commonly refers to the task of linking wikipedia pages to concepts, persons and so on mentioned in texts, i.e. not linking named entities and databases in a strict sense (Hachey et al., 2013).

Extracting biographical information from the web is an area that has been gaining more attention, especially for the (automatic) creation of biographies and biographical databases. Increasingly, this area uses IE and closely related methods. Garcia and Gamallo (2015) have explored different machine learning methods to extract biographical relations in Portuguese. Furthermore, Wikipedia has increasingly become a common source of information for various applied methods. Approaches have either been used to extract information from Wikipedia as a source of information, such as Gotti and Langlais (2017). Russo et al. (2015) explores methods to extract biographical information from Wikipedia and DBpedia. Relevant information that was extracted includes the name, birthplace, birth date, and so on, of a person.

Although NLP methods are being used in the context of creating new biographies and biographical databases, there are many issues that need to be addressed. More specifically, Fokkens et al. (2014) point out these issues that were encountered in the creation of a new database called *BiographyNet*. The authors point out that historic methods can often be hard to transfer to computational or automatic methods, since they rely on facts that may not be extracted directly, in addition to being based on interpretations, logic, analysis and so on. However, they also raise awareness of the potential biases that historians could face when using NLP methods. In particular, the authors argue that when using rule-based methods, the rules and heuristics need to be clearly indicated, and when machine learning approaches are used, the training data and features used should be described. Fokkens et al. (2014) use the example of provenance modelling to demonstrate where these biases could occur and how obvious they would be. More obvious cases could be ambiguous geographical locations, which would factor heavily in a rule-based approach. Less obvious would be unbalanced datasets that may be used in machine learning, leading to persons to be associated mistakenly with certain topics. In their final conclusion, the authors emphasise that the awareness on both the historian and NLP sides needs to be raised to the problems explained in the paper.

Work on extracting biographical information from Wikipedia using Wikidata on a larger-scale has been carried out by Plum (2018). Research carried out for that project can be seen as preliminary work for the methods we describe here. It features a dataset of around $130,000$ entities, which was selected by using Wikidata with similar parameters set out here. A short analysis of common structures containing information was carried out. Plum (2018) not only demonstrates how these common structures can be exploited and simple rules applied in order to extract information about the date of death and occupation of a person. The author also points out some of the pit-falls when working with such a large data-set, including the amount of processing time and choosing an appropriate data-structure. We make use of the analysis put

forth and apply similar rules to extract basic information. In addition, we make sure to take into account the possible obstacles to overcome.

3. Methods

In this section we present the processing pipeline used to carry out the experiments described in Section 4. The pipeline consists of three main steps, and is depicted in Figure 1. First, we pre-select a large amount of entities using Wikidata and in accordance to parameters set out by the project scope (Section 3.1.). We make the gathered meta information easily accessible via a local MongoDB[5] database. Next, we preprocess the data, using Stanford CoreNLP (Manning et al., 2014) to perform a variety of NLP tasks including NER and dependency parsing (Section 3.2.). We carry out shallow information extraction, using rules similar to those developed by Plum (2018), and described in Section 3.3. Finally, we describe an experimental approach to find relevant candidates using location matching, as well as a basic method of ranking these candidates (Section 3.4.).

3.1. Data

The selection of relevant entities was carried out in a two-fold approach. First, we developed a simple Wikidata query to return a selection of entities. Using this list of entities, we then retrieved corresponding articles from Wikipedia. Wikipedia is a large repository of information and offers a vast amount of articles for almost any conceivable topic, in various different languages. There are a number of projects that extract information from Wikipedia and make it accessible in the form of structured databases. One of these projects is Wikidata, which we utilise here. As will become clearer in later sections of this paper, it is clear that Wikidata not only includes information from Wikipedia, but also other sources. For this reason, we believe that by combining information from Wikidata and Wikipedia we are able to extract more relevant information than we would obtain from only one of the sources.

According to the parameters of the joint project, we select entities that are listed as human and that have died between 1951 and 2019. The query returns the date and place of death and birth, a short description, as well as the Wikidata link and other identification numbers. Figure 2 shows the query that was used and Figure 3 shows an example of the returned results.

A problem that occurs with the links to the individual articles is that Wikipedia uses two types of links, one using the name and one using an id number. Therefore, we use a second Wikidata query to retrieve the ids for each page in addition to the previously returned links (which use the name). The id number cannot be returned using a standard Wikidata query, i.e. there is no relation that can be specified in order to gather this for all entities at the time of the query. Instead, we use a separate query which takes each Wikidata id individually, and generates the corresponding Wikipedia id by extracting it from the Wikipedia article itself. The choice was made to use these ids, as it easier to retrieve the article for each entity from the Wikidump, which we describe in due course. Using the name could lead to some errors, due to differences in spelling across different languages. This problem was also pointed out by (Plum, 2018), where it was found that using the ID is unambiguous. Some examples are shown in Table 1.

The Wikidata query returned $401,695$ entities, and of these $172,131$ had corresponding articles in English. From the total number returned, we hope to use cross-lingual methods in future to extract from articles in languages other than English (see Section 5.). It is also worth mentioning at this point that some discrepancies exist between Wikidata and Wikipedia information, leading to some entities of the overall retrieved not being used. A account of this will be presented in Section 3.2.

Once the selection of the entities and retrieval of basic meta information was carried out, the next task involved extracting the corresponding articles from the Wikidump. A Wikidump is a snapshot of the whole Wikipedia encyclopedia in XML format. We use a Python script called WikiExtractor[6] to convert the Wikidump from XML format to plain text. The script processes the whole Wikidump, returning each article as plain text, as well as minimal meta information including the Wikipedia id and name of the article. It does not retain any structure of the XML file or provide any further markup or information. This was the main reason for using the script, as processing a Wikidump is time-consuming in itself and can involve many complications. Therefore, we opted to use this already available script in order to be able to process the vast XML files which have a complex structure. From the converted plain text, we extract all relevant articles using the previously obtained Wikipedia ids. Articles are grouped together, indicated by a begin and end tag and stored across plain text files with sizes between 1Mb - 2Mb. We run a Python script to extract each article by identifying the beginning and end tags, and store it in individual JSON files, which makes the following step of pre-processing easier.

3.2. Text Preprocessing

In order to facilitate the information extraction task, we automatically annotate each article with linguistic information. Due to the large amount of data we are processing, this task could not be carried out during extraction. Therefore, we process each article using Stanford CoreNLP (Manning et al., 2014) accessed via a Python script to carry out annotation tasks. We run a tokenizer, sentence splitter, part of speech tagger, lemmatizer, dependency parser and NER. Previous work carried out in Plum (2018) has shown that selecting an output format that is well structured and easy to process is vital to this task. Each annotated article is saved individually using a preset XML format available within CoreNLP. It is important to point out that processing such large amounts of data is extremely time-consuming. Using very modern system with good system specifications (6-core CPU, 32 GB Ram) the task took around one week of continuous processing time. Using a more multi-threaded workload could optimise this task, however, this will be addressed in future work.

[5] https://www.mongodb.com/

[6] https://github.com/attardi/wikiextractor

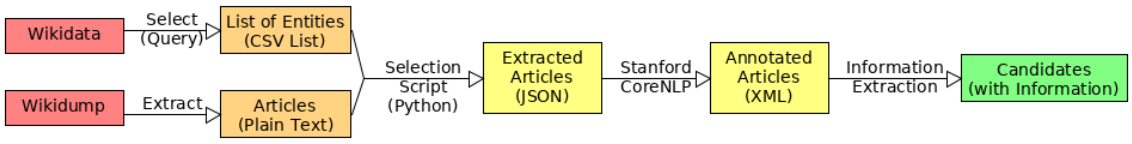

Figure 1: Sequence diagram of the processing pipeline.

```
SELECT ?h
(SAMPLE(?gnd) AS ?GND_ID)
(MIN(?name) AS ?name)
(MIN(?dob) AS ?born)
(MIN(?date) AS ?died)
(MIN(?pobLabel) AS ?loc_born)
(MIN(?podLabel) AS ?loc_death)
(MIN(?desc) AS ?description)
(GROUP_CONCAT(DISTINCT ?jobLabel; SEPARATOR=" / ") AS ?jobLabels)
WHERE {
  ?h wdt:P31 wd:Q5. # human
  ?h wdt:P569 ?dob. # born
  ?h wdt:P570 ?date. # died w/ filter
  ?h wdt:P19 ?pob. # born in
  ?h wdt:P20 ?pod. # died in
  OPTIONAL { ?h wdt:P106 ?job. } # has job
  OPTIONAL { ?h wdt:P227 ?gnd. } # show/filter GND ID
  FILTER(?date > "1950-12-31T00:00:00Z"^^xsd:dateTime)
  FILTER(?date < "1952-01-01T00:00:00Z"^^xsd:dateTime)
  #FILTER(?date < "2019-01-01T00:00:00Z"^^xsd:dateTime)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en,de,cs,hu" .
                           ?h rdfs:label ?name.
                           ?h schema:description ?desc.
                           ?job rdfs:label ?jobLabel.
                           ?pob rdfs:label ?pobLabel.
                           ?pod rdfs:label ?podLabel. }
}
GROUP BY ?h
ORDER BY ?died
```

Figure 2: Query used to retrieve entities from Wikidata.

Once processed, we were left with $170,517$ articles, down $1,614$ from the previous number. Taking the time-consuming nature of this processing into account, we had to set up a time-out for each request to the CoreNLP annotation pipeline, meaning that extremely long articles were not processed. Upon further inspection we also discovered that a small number of texts had not been converted properly, and contained either no text or corrupted text.

3.3. Basic Information Extraction

For the information extraction step, we follow a shallow rule-based approach in order to take advantage of the basic information from Wikidata. Exploiting the structure of Wikipedia articles, we extract the name and date of death of an entity. Not only do we test if using simple rule-based methods are viable for this kind of data, we also test a method to determine candidates of relevance to the project. As described in the introduction, candidates should have had an influence in the Republic of Austria. Therefore, we also extract locations mentioned in the articles and determine whether they are Austrian, hence possibly hinting at the fact that the entity of the article has some kind of connection to Austria.

By extracting the tokens of the heading of each article, we extract the name. We remove any information that is contained in brackets, which is sometimes the case in order to disambiguate certain persons. Furthermore, we use a simple rule based on observations and preliminary work carried out by (Plum, 2018): the second full date that is mentioned in the first sentence of each article is usually the date of death. As each article has been tagged in terms of named entities, including expressions of time, we simply extract this from the annotations by iterating over the time expressions. By full date we accept dates in the form *YYYY-MM-DD*. As a fall-back option, if only one expression is annotated we select this. Should no dates be detected we do not use anything. The extraction is carried out using a Python script, which compares the extracted information with that contained in the meta information.

3.4. Relevance Ranking

As the articles have been annotated in terms of named entities, we are able to extract locations by simply searching for any *LOCATION* or *CITY* tags. We employ this approach for each text, making a list of each location for each article. Next, we determine the country of each location. Our first approach was to use the GeoCoder api[7] connected to GeoNames to retrieve the country of each location. Unfor-

[7] https://geocoder.readthedocs.io/api.html

wikidata_link	GND_ID	name	born	died	loc_born
http://www.wikidata.org/entity/Q29917973		Wilhelm, Károly	1886-01-01	1951-01-01	Österreich-Ungarn
http://www.wikidata.org/entity/Q8019667	117258741	William Valentine Schevill	1864-03-02	1951-01-01	Cincinnati
http://www.wikidata.org/entity/Q51540730		Charles Frederic Ramsey	1875-01-01	1951-01-01	Pont-Aven
http://www.wikidata.org/entity/Q1270329		Heinrich Reese	1879-02-19	1951-01-01	Basel
http://www.wikidata.org/entity/Q17471830		Fazıl Doğan	1892-01-01	1951-01-01	Mytilini
http://www.wikidata.org/entity/Q1358977	141494964	Ernst Klein	1876-04-15	1951-01-01	Wien
http://www.wikidata.org/entity/Q5276568	122114973	Dikran Kelekian	1868-01-01	1951-01-01	Kayseri
http://www.wikidata.org/entity/Q21010063		Leon Karp	1903-01-01	1951-01-01	New York City
http://www.wikidata.org/entity/Q2383242		Sievert Allen Rohwer	1887-01-01	1951-01-01	Telluride
http://www.wikidata.org/entity/Q35226291	124032192	Arthur David Gayer	1903-03-19	1951-01-01	Pune

Figure 3: Example of results returned by the Wikidata query.

tunately, GeoNames is restricted to $1,000$ requests for locations per hour. With such a large dataset, this would not be a viable approach. Other than buying requests as part of a premium service, we opt to download the full GeoNames list of locations, which is freely available. This list contains locations, as well as their country. Using MongoDB, we create a database and index the location names to ensure fast searching. This way we are able to query a local database to determine the country for each location.

Using a custom Python script we query whether a location among those found in each article belongs to the Republic of Austria. If this is the case, we include the article or entity as a candidate. In the first iteration, we found that locations were being found in the documents, such as *City*. These location names are always part of a longer name, but are picked up as they are tagged individually. These locations returned matches in the database, although these were mostly mistakenly added. Therefore, we added the criterion that any location has to have a population of more than zero. We found that locations that have been mistakenly added or are meant for some other purpose usually have a population of zero.

In order to test out how a method of ranking relevant candidates could work, we count all locations in one article that are in Austria. In addition, we try to count the false matches as well. As there are a large amount of false matches we put these in contrast to the main location count. The idea is that these count could put the ranking into better perspective, if for instance the counts are equal, this candidate could possibly be excluded.

4. Results

In this section we present the results of the information extraction task, as well as the selection of possible candidates. The evaluation of our results presents a challenge, due to problems selecting a gold standard. We compare the extracted names and dates of birth to those returned by Wikidata, but this assumes Wikidata as a gold standard. Concerning the location extraction, we rely on the performance of Stanford CoreNLP and the rules we employ as to extracting the country. For reasons that will be explained later on, we do not have a gold standard for evaluation purposes here. Therefore, it should be clear that this is not an evaluation of the extraction itself, but rather the process of selecting candidates.

4.1. Wikipedia vs. Wikidata

As described in the previous section, we extracted the name and date of death from the Wikipedia articles. We compared each result with the information obtained from Wikidata. Of the $170,517$ articles, the name did not match in $18,267$ cases. Upon closer inspection, we found that this is largely due to differences in spelling, and slight differences in the name. Table 1 shows a selection of the most common errors: The first two rows are examples of differences in shorter names. Rows 3 and 4 show different levels of preciseness in naming, i.e. abbreviations. The last row shows an example where Wikidata returned the name in German, whereas we extracted the name in English.

Wikidata	**Extracted**
Robert Joshua	*Bob* Joshua
Francisco Javier Vidarte	*Paco* Vidarte
Joe C. Davis**,** Jr.	Joe C. Davis Jr.
Vincent Graber	Vincent **J.** Graber **Sr.**
Karl Aloys von und zu Liechtenstein	Prince Karl Aloys of Liechtenstein

Table 1: Various examples of naming differences.

In terms of the date of death, we had $30,153$ cases where the date of death did not match the Wikidata records. Using a Python script we counted the different errors that occurred, and found that we could classify three main errors: "no date" errors, "minor difference" errors and "birthday" errors. A breakdown of how many times each error occurred is shown in Table 2. In the first case, our extraction rule returned *0000-00-00*, indicating no date was extracted. This error was caused by the Stanford NER algorithm not detecting a date, or it being missing in the article itself. The second most common error was the "minor difference error". In this case, the difference between the Wikidata date and our extracted date was minor, i.e. only between one to five days difference. We suggest that this shows that Wikidata also gathers information from other sources, or that it could be caused by timezone differences. The last error to occur was the "birthday" error. Here, the date we extracted did not match the date of death extracted from Wikidata, but rather the corresponding date of birth. This is caused by the fact that we extract a date in sentences, even if only one date is found by the NER algorithm.

Taking these results into account, it is interesting to see where differences in data lie. Using Wikidata and Wikipedia as complementary components has many benefits. On the one hand, Wikidata serves excellently as a tool to pre-select data according to some criteria. As processing all Wikipedia, or complete Wikidumps is extremely time-consuming, this reduces the time dramatically. On the

Error	Count	Percentage
No date	19,895	11.67%
Minor	6,654	3.90%
Birthday	3,604	2.11%
Total	30,153	17.68%

Table 2: Error count for the location extraction. Compared Wikipedia vs. Wikidata.

other hand, Wikidata can serve to some extent as a kind of gold standard against which to compare the results of any extraction carried out on Wikipedia articles. Of course, this is only to a limited extent, as not all relations are available in Wikidata. This is the case with our extraction of locations in order to determine candidates.
Going beyond the use as a gold standard, the two data sources can also be used to extract information in a more complementary way, i.e. using Wikidata for basic information, and building on these known relations to extract further information from Wikipedia. It may also be of interest to compare contradicting information, as seen here with the differences in date of death.

4.2. Biographical Dictionary Candidates

Using our location matching script, we were able to obtain $13,521$ possible candidates. A short investigation of candidates picked at random shows that our technique is probably not precise enough. For each candidate, we list the location that caused its inclusion in our candidates list. The ten most common locations are listed in Table 3. While this list includes many valid locations, it is clear that many articles are chosen as candidates due to matches caused by *Hall* and *Point*. Further examples include *Sand*, *Fall* and *Gray* and so on. These match proper locations in Austria, however, they also match English nouns and adjectives, and are most probably part of longer location names. Another common problem in this regard was the matching of names which are ambiguous, as they also match locations in Austria, and therefore contributed to being considered as candidates.

Location	Count
Hall	4,191
Vienna	3,529
Point	997
Salzburg	359
Bergen	339
Nassau	197
Innsbruck	192
Graz	180
Königsberg	173
Inn	120

Table 3: Top 10 locations that led to candidate selection.

In terms of the ranking of the candidates, this is just as hard to evaluate as to measure. A brief manual analysis indicates that the ranking mechanism at this point is too crude. Quite often candidates rank very highly, even though there is no relevance to Austria whatsoever. This is mainly due to the false matches, described previously. An extract of some of the candidates below shows how highly some irrelevant candidates rank. At the other end of the scale, the reverse applies. Candidates that should probably be considered with priority are ranked very low, due to only a few ore one locations being matched. However, this is mostly due to extremely short Wikipedia articles, which do not hold much information.
Further evaluation of the locations extracted against a gold standard is not possible. Wikidata queries rely on a relation, such as *born in*, to be present in order to extract the corresponding location. In our case, we want to go beyond these relations and find any kind of mention of locations that are relevant. Ultimately, the candidates we are able to derive are to be evaluated in an iterative process by historians from the APIS/OEBL team in order to say how well our method performs. Other forms of automatic evaluation do not exist at this point in time, especially considering there is no gold standard for this work, as it is mainly aimed at finding completely new candidates.

5. Conclusion and Future Work

In this paper we presented an NLP pipeline to identify biography candidates and to extract information about them from Wikipedia and Wikidata. We show that shallow extraction methods work well for obtaining basic information about biography candidates. However, for determining possible relevant candidates there is still work to be done. While our simple method of matching locations works as a wide net, there are many irrelevant inclusions. We acknowledge that this metric by itself is too simple, however, we feel that it could become an aspect of a future metric.
As the goal of this project is detecting relevant candidates, we are currently working on improving our basic method, hopefully making use of statistical or machine learning based approaches in order to determine whether a person has had some kind of relevance in a certain area. This could also allow us to rank candidates according to their relevance for that particular area and time period.
In collaboration with the AAS we are working on a ranking system which at present is based on implicit expert knowledge. We hope to convert this knowledge to the machine, and as a part of this ongoing effort we are working on an annotating the dataset of the previous ÖBL biographies. By extracting sentences that show some kind of surface relevance to Austria, we are researching the possibility of training a machine learning classifier on the word and/or context embeddings of these sentences, in order to automatically detect them in text. This would also eliminate the need of the large-scale pre-processing beforehand.
In the future, we also aim to refine our information extraction methods and to test them on different encyclopedic repositories. In addition, we plan to explore cross-lingual methods for extracting information from data sources in other languages such as Czech, German, Hungarian, and Slovak as the core languages of the former Austrian-Hungarian monarchy.
This research is a pilot endeavour to detect relevant candidates for a biographical dictionary in online sources. It aims to contribute to three goals, 1) the further compilation of a digital, semi-automatic biographical dictionary on the case

study of the ÖBL, 2) the further development of an editing system for biographical dictionaries, which might be used as a research infrastructure, on the use case of ÖBL, and 3) triggering interdisciplinary collaboration and further pilot studies on methods and tools to detect people of "relevance".

Acknowledgements

The APIS project is funded by a research grant (project number ÖAW0405) of the Austrian Nationalstiftung für Forschung, Technologie und Entwicklung (Programm "Digital Humanities - Langzeitprojekte zum kulturellen Erbe"). We are grateful for technical support to access the APIS and ÖBL data sets by Katalin Lejtovicz and Matthias Schlögl and are thankful for the feedback and evaluation of the (art)historians Ágoston Bernad and Maximilian Kaiser.

6. References

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. 2008. An Unsupervised Approach to Biography Production Using Wikipedia. In *Proceedings of ACL*.

Judith Brouwer and Harm Nijboer. 2017. Golden Agents. A web of linked biographical data for the Dutch Golden Age. In *Proceedings of BD2017*.

Nancy Chinchor, David D Lewis, and Lynette Hirschman. 1993. Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3). *Computational linguistics*, 19(3):409–449.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of ACL*.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-based Open Information Extraction. In *Proceedings of WWW*.

Österreichische Akademie der Wissenschaften. 2012. *Österreichisches Biographisches Lexikon 1815–1950*, volume 63. Verlag der Österreichischen Akademie der Wissenschaften.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) program-tasks, data, and evaluation. In *Proceedings of LREC*.

Tomaž Erjavec, Joh Dokler, and Petra Vide Ogrin. 2015. Slovenian biography. In *Proceedings of BD2015*.

Antske Fokkens, Serge Ter Braake, Niels Ockeloen, Piek Vossen, Susan Legêne, Guus Schreiber, et al. 2014. Biographynet: Methodological issues when nlp supports historical research. In *LREC*, pages 3728–3735.

Dayne Freitag. 1998. Information extraction from HTML: Application of a general machine learning approach. In *Proceedings of AAAI/IAAI*.

Marcos Garcia and Pablo Gamallo. 2015. Exploring the effectiveness of linguistic knowledge for biographical relation extraction. *Natural Language Engineering*, 21(4):519–551.

Fabrizio Gotti and Philippe Langlais. 2017. From french Wikipedia to Erudit: A test case for cross-domain open information extraction. *Computational Intelligence*, 34(2):420–439.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A brief history. In *Proceedings of COLING*.

Christine Gruber and Eveline Wandl-Vogt. 2017. Mapping historical networks: Building the new Austrian Prosopographical Biographical Information System (APIS). Ein Überblick. In *Europa baut auf Biographien. Aspekte, Bausteine, Normen und Standards für eine europäische Biographik.*, pages 271–282.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating Entity Linking with Wikipedia. *Artificial Intelligence*, 194:130 – 150.

Lynette Hirschman. 1998. The Evolution of evaluation: Lessons from the message understanding conferences. *Computer Speech & Language*, 12(4):281 – 305.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL*.

Diego Moussallem, Thiago Castro Ferreira, Marcos Zampieri, Maria Claudia Cavalcanti, Geraldo Xexéo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. 2018. RDF2PT: Generating Brazilian Portuguese Texts from RDF Data. In *Proceedings of LREC*.

Lodewijk Petram, Jelle van Lottum, Rutger van Koert, and Sebastiaan Derks. 2015. Small Lives, Big Meanings Expanding the Scope of Biographical Data through Entity Linkage and Disambiguation. In *Proceedings of BD2015*.

Alistair Plum. 2018. Rule-based Information Extraction Using Wikipedia and Wikidata. Master's thesis, University of Wolverhampton.

Matthias Reinert and Bernhard Ebneth. 2017. Interfaces: Accessing biographical data and metadata. In *Proceedings of BD2017*.

Matthias Reinert, Maximilian Schrott, Bernhard Ebneth, and Malte Rehbein. 2015. From Biographies to Data Curation-The Making of www.deutsche-biographie.de. In *Proceedings of BD2015*.

Irene Russo, Tommaso Caselli, and Monica Monachini. 2015. Extracting and Visualising Biographical Events from Wikipedia. In *Proceedings of BD2015*.

Matthias Schlögl and Katalin Lejtovicz. 2017. A Prosopographical Information System (APIS). In *Proceedings of BD2017*.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1):233–272.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised Open Information Extraction. In *Proceedings of NAACL*.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. TextRunner: Open Information Extraction on the Web. In *Proceedings of NAACL*.