The dbpedia R Package: An Integrated Workflow for Entity Linking (for ParlaMint Corpora)

Christoph Leonhardt and Andreas Blätte
University of Duisburg-Essen
{christoph.leonhardt, andreas.blaette}@uni-due.de

Abstract

Entity Linking is a powerful approach for linking textual data to established structured data such as survey data or adminstrative data. However, in the realm of social science, the approach is not widely adopted. We argue that this is, at least in part, due to specific setup requirements which constitute high barriers for usage and workflows which are not well integrated into analyitical scenarios commonly deployed in social science research. We introduce the `dbpedia` R package to make the approach more accessible. It has a focus on functionality that is easily adoptable to the needs of social scientists working with textual data, including the support of different input formats, limited setup costs and various output formats. Using a ParlaMint corpus, we show the applicability and flexibility of the approach for parliamentary debates.

Keywords: Entity Linking, ParlaMint, Parliamentary Data

1. Introduction

Recent innovations such as transformer-based machine learning and large language models come with huge promises and great potential for scholars of different disciplines (Linegar et al., 2023). The unprecedented wealth of available data and tools continuously inspires new research questions and innovative methodological approaches. At the same time, the analysis of well-established types of structured data such as survey data or administrative data is methodologically mature and advanced at the same time, and continues to provide invaluable insights into social processes. In consequence, the possibility to combine findings from both textual data and structured data constitutes an important perspective for innovative research.

In the field of parliamentary research, these potentials are particularly apparent. Given the efforts of projects such as ParlaMint (Erjavec et al., 2023a) to create interoperable corpora of parliamentary debates and the large variety of data sets which can enrich these collections such as the Chapel Hill Expert Survey (Bakker et al., 2015), the Manifesto Project (Budge and Bara, 2001) or other statistical or administrative data sets, the combination of different types of data opens up novel perspectives for research questions which previously would be impossible or hard to address due to a lack of data and integrated analyses.

A central way to link textual data with structured data is the method of Entity Linking. Entity Linking is both an established but also actively researched area of study in the field of Natural Language Processing and Information Retrieval. In a nutshell, it comprises the disambiguation and assignment of entities in a document – often representing persons, organizations and locations – to corresponding entities in an external knowledge graph (Linhares Pontes et al., 2020, p. 218; Möller et al., 2022, p. 925). This way, the text can be represented in a “computer-processable form” (Al-Moslmi et al., 2020, p. 32862), thus potentially facilitating integrated analyses by shared unique identifiers and access to other data sets.

However, realizing this potential can be challenging. While the large number of analyses using all kinds of approaches to text analysis illustrates the interest of social scientists and beyond to apply innovative approaches in their research, Entity Linking is – until now – adopted only sporadically. We argue that this is, at least in part, due to a lack of integrated workflows and established best practices. Existing approaches do not necessarily provide guidance for social science applications and often constitute individual use cases which do not necessarily generalize well enough or are poorly maintained. Improving the accessibility of such innovative methods by approaching them from a perspective of social science and the humanities may thus be an important driver of progress.

To address this, this contribution introduces the `dbpedia` R package which is currently developed by the authors of this paper. `dbpedia` constitutes a wrapper for the statistical programming language R (R Core Team, 2023) for the Entity Linking service DBpedia Spotlight (Mendes et al., 2011; Daiber et al., 2013). In particular, it integrates the communication with the service into an R-based analysis workflow which makes Entity Linking available for existing text analysis pipelines.

This contribution proceeds as follows: First, existing applications of Entity Linking with a focus on

ParlaCLARIN IV Workshop on Creating, Analysing, and Increasing Accessibility of Parliamentary Corpora, pages 133–144
May 20, 2024.

parliamentary textual data are presented. This is followed by a discussion of requirements for the adoption of the approach. In the third section, the `dbpedia` R package is presented, using a sample of the UK corpus of the ParlaMint project (Erjavec et al., 2023a) as a show case to illustrate input formats and enrichment. This contribution concludes with a discussion of limitations and necessary next steps to contribute to the adoption of Entity Linking in social science research.

2. Related Work

2.1. Entity Linking in Social Science and Parliamentary Research

There is a number of approaches and services to facilitate the linking of entities to knowledge graphs (for a comprehensive overview, see Al-Moslmi et al., 2020). Prominent proponents of the approach are DBpedia Spotlight (Mendes et al., 2011; Daiber et al., 2013) which includes the identification, disambiguation and linking of entities in text and targets the DBpedia knowledge graph (Auer et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) which can be used to link previously identified entities to the knowledge graph of the same name.

Focusing on parliamentary debates, Olieman et al. (2015) evaluate the performance of DBpedia Spotlight and discuss associated challenges when deploying such Entity Linking systems in domain-specific settings. Using DBpedia Spotlight as a baseline to perform Entity Linking on Dutch parliamentary proceedings, they show that the tool provides links with a precision of 0.69 and a recall of 0.40 (Olieman et al., 2015; see also Olieman et al., 2017). The authors show that these measures vary between targeted entity types and provide further suggestions on improving the approach. Similarly, van Heusden et al. (2022) compare the Entity Linking solutions of DBpedia Spotlight, YAGO and Wikidata. Using corpora of ParlaMint (Erjavec et al., 2023a), they show that while feasible, the approach can be challenging when confronted with different languages and alphabets as well as with "real world data" (van Heusden et al., 2022, p. 47). The performance of the approach varies between languages and deployed systems, but all in all they "found that the WikiData [sic!] system performed the best overall for the local politicians, although all systems performed relatively well" (van Heusden et al., 2022, p. 53).

DBpedia Spotlight is also used by Glaser et al. (2022) who provide a very illustrative example on how to use Entity Linking with DBpedia Spotlight to facilitate a substantive analysis in the realm of debates of the United Nations Security Council. They discuss the method and potential limitations, thus providing some guidance on how to deploy the approach in general. Using DBpedia Spotlight instead of other Entity Linking solutions was, among other things, also informed by its relative ease-of-use and the possibility to run the service locally (Glaser et al., 2022, pp. 54-55).

For the `dbpedia` R package, we follow similar motivations when selecting DBpedia Spotlight as the service of choice. Aside from the relatively low effort to use the service (as discussed below), DBpedia Spotlight is also considered because it can be configured depending on the use case. The authors of DBpedia Spotlight describe the service as a "comprehensive and flexible solution" to annotate mentions of entities in a text with resources from the DBpedia knowledge graph (Mendes et al., 2011, p. 1). As it performs the identification of entities and uses the ontology of the underlying knowledge graph, it is not limited to pre-annotated entity types or to specific classes (Mendes et al., 2011, p. 1). The flexibility and architecture of DBpedia Spotlight are also discussed in Olieman et al. (2014, pp. 14-16).

2.2. The Need for a Package

While the projects discussed above provide great insight into the potential of the approach, a broader adoption requires that the cost of setup is minimized; workflows must integrate well into those commonly deployed in the social sciences and humanities. Accordingly, we argue that a software solution which can provide a robust framework for analyses, is easy enough to use and which provides a code base which can be maintained easier than, for example, a stand-alone script is thus an essential building block towards this goal.

In this vein, there are software implementations and wrappers for DBpedia Spotlight which address a part of the problem. As an interesting example, `spacy-dbpedia-spotlight`[1] is a library implemented in Python for users who are familiar with the popular spaCy NLP suite.[2] It seems to be well maintained and is comprehensively documented. However, its main focus is to extend the NLP pipeline of spaCy which, by itself, is not directly integrated into common social science workflows. This is true for many packages which provide the core functionality to query DBpedia Spotlight but do not provide easy paths, clear guidelines and best practices on how to use the approach in substantive analyses.

Accordingly, the `dbpedia` R package should be both robustly developed – providing options with

[1] `https://github.com/MartinoMensio/spacy-dbpedia-spotlight` (2024-02-13).

[2] `https://spacy.io` (2024-02-14).

useful default values, telling error messages, etc. – and flexible enough to be deployed in different scenarios. By providing an integrated workflow for different input types, a condensed but configurable set of commands, and including the possibility to add the enriched data to the initial input structure, the package should address some common issues when adopting the approach and equip researchers of various fields with a tool which enables them to focus on substantive research.

3. The dbpedia R Package

3.1. At a Glance

Currently only available on GitHub, the installation of dbpedia is described in some detail in the online documentation.[3] In principle, it can be run like any ordinary R package. Without any additional setup, it only needs a few lines of code to query the English public endpoint and receive Uniform Resource Identifiers (URIs) from the DBpedia knowledge graph for identified entities in a document. At the time of writing, this endpoint is provided by the maintainers of DBpedia Spotlight and can be used for minimal setup. Being a public endpoint, rate limits might apply and availability might not be guaranteed.[4]

Running the following chunk of code will result in the output similar to that shown in table 1. The results will include character offsets describing the start positions of tokens, the entities itself as well as the identified URI of the entity.

```r
library(dbpedia) # v0.1.2.9004 or higher

annotations <- get_dbpedia_uris(
  x = "The city of Turin is located
  at the river Po."
)
```

3.2. Advanced Setup

As described above, one of the advantages of DBpedia Spotlight is the easy local deployment which improves performance, avoids potential rate limits and saves resources of the publicly available endpoint. Accordingly, for our experiments and examples, we run the service locally in a Docker container. This is described by its maintainers in the corresponding online documentation.[5] Necessary computational resources depend on the language model used, but should, in general, be manageable for most modern systems.

3.3. Advanced Scenario

In the example above, we simply sent a character vector to the service. In this instance, the get_dbpedia_uris() method is somewhat limited to a wrapper which sends and receives HTTP requests and parses results. This is realized using established R packages such as httr (Wickham, 2023) and jsonlite (Ooms, 2014). However usually, challenges occur in more advanced scenarios. They include the preparation of different input formats and the presentation of results in a useful way, for example by mapping identified entities back to the input data. In the following, we present the functionality of dbpedia to adopt Entity Linking in a plausible social science scenario.

3.3.1. Input Data

Textual data comes in different shapes and forms. While sometimes, it is provided as a single continuous string, other times it is already separated into individual tokens. Sometimes the data is available in a tabular representation and other times it is represented in more complex formats such as XML or the Corpus Workbench format (Evert and Hardie, 2011). The dbpedia package is designed to account for this variety of input formats and provides workflows for different data types such character vectors, quanteda corpora (Benoit et al., 2018), Corpus Workbench subcorpora and XML.

As discussed before, parliamentary debates are an attractive subject for Entity Linking. Accordingly, this contribution focuses on an emerging standard for encoding this type of textual data and presents the workflow of dbpedia for corpora represented in the XML schema of the ParlaMint project (Erjavec et al., 2023a). The corpora of ParlaMint follow strict encoding guidelines for parliamentary data in the XML data format, thus ensuring interoperability and comparability. The corpora include different levels of structural and linguistic annotation. Named entities are identified, but not linked to an external knowledge base.

The interoperable format of ParlaMint also benefits the development of tools such as the dbpedia R package, as it increases the number of potential use cases. While DBpedia Spotlight supports many languages out of the box,[6] ParlaMint also

[3] The package is available on https://github.com/PolMine/dbpedia (2024-03-30).

[4] Also see the presentation of the tool here: https://www.dbpedia-spotlight.org (2024-02-14).

[5] https://github.com/dbpedia-spotlight/spotlight-docker (2024-02-13).

[6] The documentation of the spacy-dbpedia-spotlight Python library referred to above provides a useful overview regarding supported languages.

start	text	dbpedia_uri
5	city	http://dbpedia.org/resource/City
13	Turin	http://dbpedia.org/resource/Turin
45	Po	http://dbpedia.org/resource/Po_(river)

Note: Entity types annotated by DBpedia Spotlight are omitted for legibility.

Table 1: Entities returned by DBpedia Spotlight

provides a machine-translated English version of all corpora, further broadening the applicability of the approach. Realizing a robust Entity Linking workflow for ParlaMint thus opens up avenues for a host of corpora in the realm of parliamentary research, facilitating both longitudinal and comparative research by enriching the textual data with URIs (see also van Heusden et al., 2022).

The data used in this example application is taken from the linguistically annotated sample of ParlaMint for Great Britain provided in the ParlaMint GitHub repository. The chosen single sample file is based on the corpus prepared by Matthew Coole as part of the ParlaMint 4.0 release (Erjavec et al., 2023b).[7] Since the following steps only illustrate the Entity Linking process in general, the specific file has been chosen rather arbitrarily after it became apparent that the document contained substantive speech and, in consequence, entities which could be linked to a knowledge graph.

With ParlaMint being well-formed XML, the data is first read into R using the `xml2` R package (Wickham et al., 2023).

3.3.2. Entity Linking and Parsing

To start the Entity Linking process, the package is loaded.

```
library(dbpedia)
```

When the package is first loaded, setup messages inform the user about the endpoint of the service and the chosen language. It will also indicate whether DBpedia Spotlight is running locally in a Docker container. While the endpoint indicates where the queries are sent to, the language parameter indicates the chosen language model and is used to select a list of stop words which are excluded from the Entity Linking process. Both the endpoint and the language parameters are used as arguments in the main function of the package presented below.

`dbpedia` provides the `get_dbpedia_uris()` method which takes care of pre-processing the data, interaction with DBpedia Spotlight as well as the parsing of the linking results into a format which is appropriate for different analysis scenarios. The method can handle different input formats such as tokenized XML.

In keeping with the motivation to streamline the process of Entity Linking when working with textual data, the set of commands and parameters was carefully chosen to limit the number of confusing and potentially overwhelming options. Nevertheless, the process should also be transparent and open for configuration. As such, a number of parameters can be set. The package, while still in development, provides documentation for a number of basic scenarios. The most important arguments specific for XML input are the following:

- `x`: the input XML
- `feature_tag`: a `character vector` containing the name of XML elements which should be considered for Entity Linking. Can be used to select pre-annotated named entities.
- `segment`: a `character vector` describing segments into which the document should be split (e.g. paragraphs), to account for the maximum length of documents supported by DBpedia Spotlight.
- `token_tags`: a `character vector` containing the names of XML tags representing tokens

Setting these parameters requires some knowledge about the input data. For ParlaMint it seems reasonable to segment the input using the `<seg>` tag provided in the data. Assuming that these nodes represent paragraphs, this segmentation should provide sufficient context for the entity linking approach (see also Glaser et al., 2022, p. 55). This is also related to the `max_len` parameter which indicates the maximum length of segments of text to be sent to the server in one query. The default is mainly informed by the maximum length of

[7] The file was downloaded from https://github.com/clarin-eric/ParlaMint/blob/main/Samples/ParlaMint-GB/ParlaMint-GB_2022-07-21-commons.ana.xml on 2024-02-06. As stated in this example file, the corpus is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/) and the Open Parliament Licence v3.0 (https://www.parliament.uk/site-information/copyright-parliament/open-parliament-licence/).

characters which can be reliably processed in one query before DBpedia Spotlight starts to return errors. The `feature_tag` parameter can be useful when data is already pre-annotated with named entities and the envisioned analysis focuses on specific elements such as persons, organizations and locations. In this case, `dbpedia` will limit the output to entity links which exactly match the pre-annotated entities. Otherwise, the method will return a large number of entities of all kinds of types. The parameters `confidence`, `support` and `types` are described by Mendes et al. (2011, pp. 3-4). All arguments are also documented in the package.

```r
annotations <- get_dbpedia_uris(
  x = xml_doc,
  language = getOption("dbpedia.lang"),
  feature_tag = NULL,
  segment = "seg",
  token_tags = c("w", "pc"),
  text_tag = "text",
  max_len = 5600L,
  confidence = 0.7,
  api = getOption("dbpedia.endpoint"),
  types = character(),
  support = 20,
  expand_to_token = FALSE,
  drop_inexact_annotations = TRUE,
  verbose = TRUE
)
```

After this call, the method creates a token stream using the elements ("token_tags") of each segment ("seg") and sends it to the DBpedia Spotlight service. DBpedia Spotlight identifies token spans representing entities and assigns types as well as URIs of entries in the DBpedia knowledge graph to these spans.

3.3.3. Working with the Output

`get_dbpedia_uris()` returns a tabular representation of identified entities and additional information such as individual entity types for many entities. Depending on the input format, character offsets or token IDs describing the position of the enriched entity are returned as well. Table 1 above illustrates this for the input of character vectors, while table 2 shows the output for the ParlaMint XML format. Table 3 visualizes retrieved entities for a single segment.

3.3.4. Enrichment with SPARQL

Often, the addition of DBpedia URIs is not the final objective of the approach but only an intermediate step to enrich entities with information available in external knowledge graphs such as DBpedia itself or Wikidata. Since the community can directly add information to the latter, Wikidata can be particularly interesting as a target knowledge graph to enrich textual data with additional information via Entity Linking (Möller et al., 2022, pp. 936-938).

In line with the aspiration to provide a cohesive workflow, `dbpedia` integrates the functionality to query DBpedia as well as Wikidata using the SPARQL query language. The respective functions `dbpedia_get_wikidata_uris()` and `wikidata_query()` facilitate this enrichment. Both functions work as wrappers included to alleviate some of the burden to construct valid SPARQL queries for specific endpoints of the knowledge graphs. In a nutshell, both functions take URIs as an input, prepare a SPARQL query using a template and send it to the respective SPARQL endpoints. The main functionality of `dbpedia_get_wikidata_uris()` is the retrieval of Wikidata IDs based on the `owl:sameAs` property provided by the knowledge graph. If desired, additional information – e.g. the ISO code of countries – could be retrieved. In this example, we focus only on the retrieval of Wikidata IDs. Note that rate limits and other limitations apply for the public endpoint.[8]

```r
endpnt <- "https://dbpedia.org/sparql/"

wd_uris <- dbpedia_get_wikidata_uris(
  annotations[["dbpedia_uri"]],
  endpoint = endpnt,
  wait = 5,
  chunksize = 100,
  progress = TRUE
)
```

The returned values suggest that mapping DBpedia URIs to Wikidata IDs is not without challenges. `owl:sameAs` often contains multiple Wikidata IDs for a single DBpedia URI. For example, for the entity "United_Kingdom", three Wikidata IDs are returned by DBpedia which describe the "United Kingdom" as a "country in northwest Europe" (Q145), the "United Kingdom of Great Britain and Ireland" as a "historical sovereign state (1801–1922)" (Q174193) and "Great Britain" as an "island in the North Atlantic Ocean off the northwest coast of continental Europe" (Q23666).[9]

This observation is already described by Glaser et al. (2022, p. 55). To address this, Glaser et al. (2022) compare the labels of both knowledge graphs to identify the correct Wikidata ID for each item. van Heusden et al. (2022, p. 49) suggest an approach to identify missing Wikidata IDs by retrieving the Wikipedia page the DBpedia item is based on. This allows them to gather the Wikidata

[8] See the documentation here https://www.dbpedia.org/resources/sparql/ (2024-02-26).

[9] Cited passages refer to the entity labels of the three items on Wikidata as of 2024-02-16.

original_id	dbpedia_uri	text
ParlaMint-GB_2022-07-21-commons.seg5.2.10 ParlaMint-GB_2022-07-21-commons.seg5.2.11	http://dbpedia.org/resource/Free_trade	free trade
ParlaMint-GB_2022-07-21-commons.seg5.2.14	http://dbpedia.org/resource/India	India
ParlaMint-GB_2022-07-21-commons.seg870.1.10	http://dbpedia.org/resource/Glasgow	Glasgow
ParlaMint-GB_2022-07-21-commons.seg870.1.13	http://dbpedia.org/resource/Scotland	Scotland
ParlaMint-GB_2022-07-21-commons.seg870.1.17 ParlaMint-GB_2022-07-21-commons.seg870.1.18	http://dbpedia.org/resource/United_Kingdom	United Kingdom
ParlaMint-GB_2022-07-21-commons.seg870.5.15	http://dbpedia.org/resource/Christmas	Christmas

Note: Two illustrative segments of the sample document. Removed columns 'segment_id' and 'types' for improved legibility. Additional line breaks for Token IDs in column 'original_id'.

Table 2: Entities returned by DBpedia Spotlight - Tabular Overview

segment_id	text	entities
ParlaMint-GB_2022-07-21-commons.seg5	1. What progress her Department has made on securing a free trade agreement with India.	free trade (http://dbpedia.org/resource/Free_trade) | India (http://dbpedia.org/resource/India)

Table 3: Entities returned by DBpedia Spotlight - In Segments

ID indirectly. Following this approach could make it possible to identify a suitable ID if more than one Wikidata ID is provided for an entity in the DBpedia knowledge graph. In this case, instead of using the `owl:sameAs` property, the Wikipedia page would be queried and mapped to its corresponding Wikidata ID.

The ontology of Wikidata could also be used to distinguish different entities. Using the example above, the different versions of "United Kingdom" could be queried on Wikidata to retrieve the instances they are a part of (property P31) such as "sovereign state" (Q3624078), "island" (Q23442) or "historical country" (Q3024240). `dbpedia` includes the functionality for this to make this step easier via the `wikidata_query()` function which uses the `WikidataQueryServiceR` R package (Popov, 2020) and queries the Wikidata Query Service SPARQL endpoint.[10] As above, rate limits apply.

```r
wd_ids <- c("Q145", "Q174193", "Q23666")

wd_props <- wd_ids |>
  wikidata_query(
    id = "P31",
    progress = TRUE)
```

However, when using Wikidata in this way, the assignment relies on the specific configuration of the knowledge graph. For instance, while this would allow to select only items which describe "sovereign states", both item Q145 (which we likely would keep as the appropriate Wikidata ID) and item Q174193 (the "historical sovereign state") are instances of this class in the knowledge graph. For the latter item, the instance of "sovereign state" is not returned by the SPARQL query above because in this specific query the returned value is limited to the highest ranked value in the statement.[11]

In consequence, while the integration of querying additional knowledge graphs seems useful for the scope and purpose of the package, there are limits to its current implementation. Addressing more complex applications is still to be tested. Ultimately, what `dbpedia_get_wikidata_uris()` and `wikidata_query()` facilitate are basic queries and the enrichment of DBpedia URIs with plausible Wikidata IDs and some additional data. More complex scenarios which also require some knowledge about the underlying knowledge graph and its ontology and structure can still be addressed with SPARQL queries regardless of the features of this package, however.

[10] https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service (2024-02-21).

[11] This is described in the documentation of Wikidata: https://www.wikidata.org/wiki/Help:Ranking (2024-02-16).

3.3.5. Enrichment of XML

A crucial feature of dbpedia is the possibility to map identified entities back to the tokens in the input data. After the DBpedia URIs, Wikidata IDs and additional properties are retrieved, the function xml_enrich() takes care of this. It extracts entities from the annotation table and maps them onto the input data via their relative IDs. For ParlaMint this technically comprises of either adding parent nodes to tokens which are identified as entities by DBpedia Spotlight or enriching existing entity annotations with additional attributes describing the type and URI of the entity. Regarding the enrichment of existing entity annotations, it has to be noted that the alignment of pre-existing and newly added entity spans can be challenging and is under development in the current version of the dbpedia R package. As discussed above, DBpedia Spotlight returns types for many entities. These often include references to types in a number of different knowledge graphs and ontologies. Since the encoding guidelines of ParlaMint limit possible values for the “type” attribute, types returned by DBpedia Spotlight can be mapped onto this allowed set of values to adhere to specific guidelines or applications.[12]

Aside from the annotation table created by the get_dbpedia_uris() method, the arguments of the function account for the name of nodes which potentially contain entities, a name for the entity nodes to be added or enriched as well as the names of columns in the annotation table which should be added as XML attributes. For a visualization of these modifications, please see the listings in the appendix (A and B) which represent a single sentence of the document.

```
xml_enrich(
  xml = xml_doc,
  annotation_dt = annotations,
  entity_name = "name",
  token_tags = c("w", "pc"),
  feature_tag = "name",
  ref = "dbpedia_uri",
  type = "category"
)
```

4. Limitations and Next Steps

The R package dbpedia provides an intuitive and cohesive workflow to perform Entity Linking using the DBpedia Spotlight Entity Linking tool with a variety of input formats. In its current state there are some limitations concerning Entity Linking in social science research as a whole and the design principles and applicability of the R package dbpedia in particular.

Regarding Entity Linking with DBpedia Spotlight, we currently lack benchmarks on the actual performance of DBpedia Spotlight when applied to parliamentary research and beyond. While benchmarks provided by the developers of DBpedia Spotlight (Daiber et al., 2013) and others indicate the usefulness of the approach, given the specificities of research scenarios in social science research, further steps of quality control should be taken. This is of particular relevance as the importance of the specific domain of textual data for approaches and corresponding benchmarks for Entity Linking is subject of some discussion and challenges (van Erp et al., 2016, pp. 4377-4378). As discussed above, the study by Olieman et al. (2015) presents some crucial insights into the performance of DBpedia Spotlight concerning Dutch parliamentary proceedings and van Heusden et al. (2022) provide some valuable perspectives on the general performance of different approaches for parliamentary debates across different languages. However, further evaluation would be crucial for substantive research. When does the approach work and when does it fail? Which accuracy can be expected? How does this affect substantive downstream tasks? In comparative parliamentary research, for example, the applicability of the approach might not only depend on the language or genre of a text but also on other aspects such as time. If the reliability of results varies over time, substantive results might depend on whether the performance of Entity Linking is worse on older documents than on more recent ones or vice versa.

Focusing on making the approach easier to use as a necessary starting point, this contribution does not yet add to these perspectives on the performance of DBpedia Spotlight. However, despite potential challenges when evaluating Entity Linking systems and creating reliable gold standard annotation (Olieman et al., 2017), given its relevance for the applicability of the approach in the envisioned scenarios and its broader adoption, the estimation of its performance as well as accompanying guidelines and advice on how to best facilitate reliable research is a crucial next step.

DBpedia Spotlight was purposefully chosen as the backbone of the package. Given its relatively easy deployment in particular, the implementation of Entity Linking with this tool provides a great baseline to address questions of usefulness, accessibility and the actual usage of the approach in social science research. This also means that the current approach relies on the DBpedia knowledge graph. However, considering the recent promi-

[12] According to https://clarin-eric.github.io/ParlaMint/#sec-ner allowed types in ParlaMint are PER (person), LOC (location), ORG (organization) and MISC (miscellaneous) (2024-03-31).

nence of Wikidata which could also be used as a direct target for Entity Linking (Möller et al., 2022) and the challenges of mapping DBpedia URIs to Wikidata, finding better solutions to access other knowledge graphs is worth pursuing.

5. Conclusion

dbpedia aims to make innovations in Natural Language Processing and Information Retrieval accessible in the social science community in order to facilitate the combination of unstructured data such as textual data and data such as survey data and administrative data. This contribution illustrated the possibilities of an integrated workflow for parliamentary debates in the form of corpora in the ParlaMint encoding schema. The package allows to create immediate representations of extracted and disambiguated entities, but also facilitates the addition of the enriched data to the initial corpora. This, in turn, makes it possible to use this additional information – for example statistical data added via extracted URIs – in workflows scholars working with corpora are already familiar with, for example by creating relevant subsets of documents or deploying common methods of corpus analysis.

Since it is work-in-progress, the functionality of the package is subject to future changes. The current focus of dbpedia is on the development of a slim set of functions and commands which apply in different scenarios.

As indicated in the previous section, there are some obvious next steps: We neither discussed the substantive performance of the approach, nor is DBpedia Spotlight the only Entity Linking solution worth considering. While the local deployment and performance are advantages, there are more recent developments which should be evaluated. For future research, this might entail complementing dbpedia with other components which build on the functionality and API of the presented package and facilitate the integration of different approaches. In consequence, a modular design of tools and workflows might be needed which can handle different but standardized input formats such as the ParlaMint corpora and beyond.

6. Acknowledgement

The development of the dbpedia R package is made possible by funding from KonsortSWD (project number 442494171) within the German National Research Data Infrastructure (Nationale Forschungsdateninfrastruktur/NFDI). This funding is gratefully acknowledged. We also want to thank the anonymous reviewers for their insightful comments and useful suggestions.

7. Bibliographical References

Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. 2020. Named Entity Extraction for Knowledge Graphs: A Literature Overview. *IEEE Access*, 8:32862–32881.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ryan Bakker, Catherine de Vries, Erica Edwards, Liesbet Hooghe, Seth Jolly, Gary Marks, Jonathan Polk, Jan Rovny, Marco Steenbergen, and Milada Anna Vachudova. 2015. Measuring party positions in Europe: The Chapel Hill expert survey trend file, 1999–2010. *Party Politics*, 21(1):143–152.

Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30):774.

Ian Budge and Judith Bara. 2001. Introduction: Content Analysis and Political Texts. In Ian Budge, Hans-Dieter Klingemann, Andrea Volkens, Judith Bara, and Eric Tanenbaum, editors, *Mapping Policy Preferences. Estimates for Parties, Electors, and Governments 1945-1998*, pages 1–16. Oxford University Press, Oxford; New York.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA. Association for Computing Machinery.

Tomaž Erjavec, Maciej Ogrodniczuk, Petya Osenova, Nikola Ljubešić, Kiril Simov, Andrej Pančur, Michał Rudolf, Matyáš Kopp, Starkaður Barkarson, Steinþór Steingrímsson, Çağrı Çöltekin, Jesse de Does, Katrien Depuydt, Tommaso Agnoloni, Giulia Venturi, María Calzada Pérez, Luciana D. de Macedo, Costanza Navarretta, Giancarlo Luxardo, Matthew Coole, Paul Rayson, Vaidas Morkevičius, Tomas Krilavičius, Roberts Dar̗gis, Orsolya Ring, Ruben van Heusden, Maarten Marx, and Darja Fišer. 2023a. The ParlaMint corpora of parliamentary proceedings.

Language Resources and Evaluation, 57:415–448.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century Corpus Workbench: Updating a query architecture for the new millenium. In *Proceedings of the Corpus Linguistics 2011 conference*, University of Birmingham, UK.

Luis Glaser, Ronny Patz, and Manfred Stede. 2022. UNSC-NE: A Named Entity Extension to the UN Security Council Debates Corpus. *Journal for Language Technology and Computational Linguistics*, 35(2):51–67.

Mitchell Linegar, Rafal Kocielnik, and R. Michael Alvarez. 2023. Large language models and political science. *Frontiers in Political Science*, 5.

Elvys Linhares Pontes, Luis Adrián Cabrera-Diego, Jose G. Moreno, Emanuela Boros, Ahmed Hamdi, Nicolas Sidère, Mickaël Coustaty, and Antoine Doucet. 2020. Entity Linking for Historical Documents: Challenges and Solutions. In *Digital Libraries at Times of Massive Societal Transition*, volume 12504 of *Lecture Notes in Computer Science*, pages 215–231, Cham. Springer International Publishing.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems*, Graz, Austria.

Cedric Möller, Jens Lehmann, and Ricardo Usbeck. 2022. Survey on English Entity Linking on Wikidata. Datasets and approaches. *Semantic Web*, 13:925–966.

Alex Olieman, Hosein Azarbonyad, Mostafa Dehghani, Jaap Kamps, and Maarten Marx. 2014. Entity linking by focusing DBpedia candidate entities. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 13–24, New York, NY, USA. Association for Computing Machinery.

Alex Olieman, Kaspar Beelen, Milan van Lange, Jaap Kamps, and Maarten Marx. 2017. Good Applications for Crummy Entity Linkers? The Case of Corpus Selection in Digital Humanities. In *Proceedings of the 13th International Conference on Semantic Systems*, Amsterdam, Netherlands.

Alex Olieman, Jaap Kamps, Maarten Marx, and Arjan Nusselder. 2015. A Hybrid Approach to Domain-Specific Entity Linking. *arXiv:1509.01865*.

Jeroen Ooms. 2014. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. *arXiv:1403.2805*.

Mikhail Popov. 2020. *WikidataQueryServiceR: API Client Library for 'Wikidata Query Service'*. R package version 1.0.0.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Marieke van Erp, Pablo N. Mendes, Heiko Paulheim, Filip Ilievski, Julien Plu, Giuseppe Rizzo, and Joerg Waitelonis. 2016. Evaluating Entity Linking: An Analysis of Current Benchmark Datasets and a Roadmap for Doing a Better Job. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4373–4379, Portorož, Slovenia. European Language Resources Association (ELRA).

Ruben van Heusden, Maarten Marx, and Jaap Kamps. 2022. Entity Linking in the ParlaMint Corpus. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, pages 47–55, Marseille, France. European Language Resources Association.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85. Publisher: ACM New York, NY, USA.

Hadley Wickham. 2023. *httr: Tools for Working with URLs and HTTP*. R package version 1.4.7.

Hadley Wickham, Jim Hester, and Jeroen Ooms. 2023. *xml2: Parse XML*. R package version 1.3.5.

8. Language Resource References

Erjavec, Tomaž and Kopp, Matyáš and Ogrodniczuk, Maciej and Osenova, Petya and Agerri, Rodrigo and Agirrezabal, Manex and Agnoloni, Tommaso and Aires, José and Albini, Monica and Alkorta, Jon and Antiba-Cartazo, Iván and Arrieta, Ekain and Barcala, Mario and Bardanca, Daniel and Barkarson, Starkaður and Bartolini, Roberto and Battistoni, Roberto and Bel, Nuria and Bonet Ramos, Maria del Mar and Calzada Pérez, María and Cardoso, Aida and Çöltekin, Çağrı and Coole, Matthew and Darģis, Roberts and de Does, Jesse and de Libano, Ruben and

Depoorter, Griet and Depuydt, Katrien and Diwersy, Sascha and Dodé, Réka and Fernandez, Kike and Fernández Rei, Elisa and Frontini, Francesca and Garcia, Marcos and García Díaz, Noelia and García Louzao, Pedro and Gavriilidou, Maria and Gkoumas, Dimitris and Grigorov, Ilko and Grigorova, Vladislava and Haltrup Hansen, Dorte and Iruskieta, Mikel and Jarlbrink, Johan and Jelencsik-Mátyus, Kinga and Jongejan, Bart and Kahusk, Neeme and Kirnbauer, Martin and Kryvenko, Anna and Ligeti-Nagy, Noémi and Ljubešić, Nikola and Luxardo, Giancarlo and Magariños, Carmen and Magnusson, Måns and Marchetti, Carlo and Marx, Maarten and Meden, Katja and Mendes, Amália and Mochtak, Michal and Mölder, Martin and Montemagni, Simonetta and Navarretta, Costanza and Nitoń, Bartłomiej and Norén, Fredrik Mohammadi and Nwadukwe, Amanda and Ojsteršek, Mihael and Pančur, Andrej and Papavassiliou, Vassilis and Pereira, Rui and Pérez Lago, María and Piperidis, Stelios and Pirker, Hannes and Pisani, Marilina and van der Pol, Henk and Prokopidis, Prokopis and Quochi, Valeria and Rayson, Paul and Regueira, Xosé Luís and Rudolf, Michał and Ruisi, Manuela and Rupnik, Peter and Schopper, Daniel and Simov, Kiril and Sinikallio, Laura and Skubic, Jure and Tamper, Minna and Tungland, Lars Magne and Tuominen, Jouni and van Heusden, Ruben and Varga, Zsófia and Vázquez Abuín, Marta and Venturi, Giulia and Vidal Miguéns, Adrián and Vider, Kadri and Vivel Couso, Ainhoa and Vladu, Adina Ioana and Wissik, Tanja and Yrjänäinen, Väinö and Zevallos, Rodolfo and Fišer, Darja. 2023b. *Linguistically annotated multilingual comparable corpora of parliamentary debates ParlaMint.ana 4.0*. ISSN: 2820-4042.

Appendices

A. ParlaMint Example XML output before Entity Linking

```
<s xml:id=”ParlaMint-GB_2022-07-21-commons.seg5.2”>
   <w lemma=”what” msd=”UPosTag=DET|PronType=Int” pos=”WDT” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.1”>What</w>
   <w lemma=”progress” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.2”>progress</w>
   <w lemma=”she” msd=”UPosTag=DET|Gender=Fem|Number=Sing|Person=3|Poss=Yes|
       PronType=Prs” pos=”PRP$” xml:id=”ParlaMint-GB_2022-07-21-commons.seg5
       .2.3”>her</w>
   <w lemma=”Department” msd=”UPosTag=PROPN|Number=Sing” pos=”NNP” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.4”>Department</w>
   <w lemma=”have” msd=”UPosTag=VERB|Mood=Ind|Number=Sing|Person=3|Tense=
       Pres|VerbForm=Fin” pos=”VBZ” xml:id=”ParlaMint-GB_2022-07-21-commons.
       seg5.2.5”>has</w>
   <w lemma=”make” msd=”UPosTag=VERB|Tense=Past|VerbForm=Part” pos=”VBN” xml
       :id=”ParlaMint-GB_2022-07-21-commons.seg5.2.6”>made</w>
   <w lemma=”on” msd=”UPosTag=ADP” pos=”IN” xml:id=”ParlaMint-GB_2022-07-21-
       commons.seg5.2.7”>on</w>
   <w lemma=”secure” msd=”UPosTag=VERB|VerbForm=Ger” pos=”VBG” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.8”>securing</w>
   <w lemma=”a” msd=”UPosTag=DET|Definite=Ind|PronType=Art” pos=”DT” xml:id
       =”ParlaMint-GB_2022-07-21-commons.seg5.2.9”>a</w>
   <w lemma=”free” msd=”UPosTag=ADJ|Degree=Pos” pos=”JJ” xml:id=”ParlaMint-
       GB_2022-07-21-commons.seg5.2.10”>free</w>
   <w lemma=”trade” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.11”>trade</w>
   <w lemma=”agreement” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
       ParlaMint-GB_2022-07-21-commons.seg5.2.12”>agreement</w>
   <w lemma=”with” msd=”UPosTag=ADP” pos=”IN” xml:id=”ParlaMint-GB_2022
       -07-21-commons.seg5.2.13”>with</w>
   <w join=”right” lemma=”India” msd=”UPosTag=PROPN|Number=Sing” pos=”NNP”
       xml:id=”ParlaMint-GB_2022-07-21-commons.seg5.2.14”>India</w>
   <pc msd=”UPosTag=PUNCT” pos=”.” xml:id=”ParlaMint-GB_2022-07-21-commons.
       seg5.2.15”>.</pc>
</s>
```

Listing 1: XML before Entity Linking

Note: A single sentence based on sample data for the ParlaMint 4.0 corpora (Erjavec et al., 2023b). Removed syntactic information for better legibility. See footnote 7 regarding the source of the data.

B. ParlaMint Example XML output after Entity Linking

```
<s xml:id=”ParlaMint-GB_2022-07-21-commons.seg5.2”>
  <w lemma=”what” msd=”UPosTag=DET|PronType=Int” pos=”WDT” xml:id=”
      ParlaMint-GB_2022-07-21-commons.seg5.2.1”>What</w>
  <w lemma=”progress” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
      ParlaMint-GB_2022-07-21-commons.seg5.2.2”>progress</w>
  <w lemma=”she” msd=”UPosTag=DET|Gender=Fem|Number=Sing|Person=3|Poss=Yes|
      PronType=Prs” pos=”PRP$” xml:id=”ParlaMint-GB_2022-07-21-commons.seg5
      .2.3”>her</w>
  <w lemma=”Department” msd=”UPosTag=PROPN|Number=Sing” pos=”NNP” xml:id=”
      ParlaMint-GB_2022-07-21-commons.seg5.2.4”>Department</w>
  <w lemma=”have” msd=”UPosTag=VERB|Mood=Ind|Number=Sing|Person=3|Tense=
      Pres|VerbForm=Fin” pos=”VBZ” xml:id=”ParlaMint-GB_2022-07-21-commons.
      seg5.2.5”>has</w>
  <w lemma=”make” msd=”UPosTag=VERB|Tense=Past|VerbForm=Part” pos=”VBN” xml
      :id=”ParlaMint-GB_2022-07-21-commons.seg5.2.6”>made</w>
  <w lemma=”on” msd=”UPosTag=ADP” pos=”IN” xml:id=”ParlaMint-GB_2022-07-21-
      commons.seg5.2.7”>on</w>
  <w lemma=”secure” msd=”UPosTag=VERB|VerbForm=Ger” pos=”VBG” xml:id=”
      ParlaMint-GB_2022-07-21-commons.seg5.2.8”>securing</w>
  <w lemma=”a” msd=”UPosTag=DET|Definite=Ind|PronType=Art” pos=”DT” xml:id
      =”ParlaMint-GB_2022-07-21-commons.seg5.2.9”>a</w>
  <name type=”MISC” ref=”http://dbpedia.org/resource/Free_trade”>
    <w lemma=”free” msd=”UPosTag=ADJ|Degree=Pos” pos=”JJ” xml:id=”ParlaMint
        -GB_2022-07-21-commons.seg5.2.10”>free</w>
    <w lemma=”trade” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
        ParlaMint-GB_2022-07-21-commons.seg5.2.11”>trade</w>
  </name>
  <w lemma=”agreement” msd=”UPosTag=NOUN|Number=Sing” pos=”NN” xml:id=”
      ParlaMint-GB_2022-07-21-commons.seg5.2.12”>agreement</w>
  <w lemma=”with” msd=”UPosTag=ADP” pos=”IN” xml:id=”ParlaMint-GB_2022
      -07-21-commons.seg5.2.13”>with</w>
  <name type=”LOC” ref=”http://dbpedia.org/resource/India”>
    <w join=”right” lemma=”India” msd=”UPosTag=PROPN|Number=Sing” pos=”NNP”
        xml:id=”ParlaMint-GB_2022-07-21-commons.seg5.2.14”>India</w>
  </name>
  <pc msd=”UPosTag=PUNCT” pos=”.” xml:id=”ParlaMint-GB_2022-07-21-commons.
      seg5.2.15”>.</pc>
</s>
```

Listing 2: XML after Entity Linking

Note: A single sentence based on sample data for the ParlaMint 4.0 corpora (Erjavec et al., 2023b). Removed syntactic information for better legibility. See footnote 7 regarding the source of the data.