DBpedia Abstracts: A Large-Scale, Open, Multilingual NLP Training Corpus

[1]Martin Brümmer, [1,2]Milan Dojchinovski, [1]Sebastian Hellmann
[1]AKSW, InfAI at the University of Leipzig, Germany
[2]Web Intelligence Research Group, FIT, Czech Technical University in Prague
{bruemmer,dojchinovski,hellmann}@informatik.uni-leipzig.de

Abstract

The ever increasing importance of machine learning in Natural Language Processing is accompanied by an equally increasing need in large-scale training and evaluation corpora. Due to its size, its openness and relative quality, the Wikipedia has already been a source of such data, but on a limited scale. This paper introduces the DBpedia Abstract Corpus, a large-scale, open corpus of annotated Wikipedia texts in six languages, featuring over 11 million texts and over 97 million entity links. The properties of the Wikipedia texts are being described, as well as the corpus creation process, its format and interesting use-cases, like Named Entity Linking training and evaluation.

Keywords: training, dbpedia, corpus, named entity recognition, named entity linking, nlp

1. Introduction

Wikipedia is the most important and comprehensive source of open, encyclopedic knowledge. The English Wikipedia alone features over 5 million articles, representing a vast source of openly licensed natural language text.

Besides the Wikipedia web site, Wikipedia data is available via a RESTful API as well as complete XML dumps. However, API access is officially limited to one request per second, prohibiting a web scraping approach to acquire the data. Even after downloading the XML dump files, consuming and processing them is a complicated task, further hindering large-scale data extraction. To alleviate these issues, the DBpedia project (Auer et al., 2008) has been extracting, mapping, converting and publishing Wikipedia data since 2007. The main focus of the DBpedia extraction lies in mapping of info boxes, templates and other easily identified structured data found in Wikipedia articles to properties of an ontology. Article texts and the data they may contain are not focused in the extraction process, although they are the largest part of most articles in terms of time spent on writing, informational content and size. Only the text of the first introductory section up until the first heading of the articles is extracted and contained in the DBpedia, and called *abstract*. Links inside the articles are only extracted as an unordered bag, showing only an unspecified relation between the article that contains the link and the articles that are being linked, but not where in the text the linked article was mentioned or what its surface form was. But because the links are set by the contributors to Wikipedia themselves, they represent entities in the text that were intellectually disambiguated by URL. This property makes extracting the abstracts including the links and their exact position in the text an interesting opportunity to create a corpus usable for, among other cases, NER and NEL algorithm training and evaluation.

This paper describes the creation of a large-scale, multilingual Wikipedia abstract corpus that contains Wikipedia abstract texts, entity mentions occuring in the text, their position, surface form as well as the entity link. The vast majority of entity links are linking Wikipedia pages, allowing for extraction of related categories and entity types. It contributes by making a large number of Wikipedia abstracts in currently six languages available for bulk processing with NLP tools. NIF (Hellmann et al., 2013) was used as the corpus format to provide DBpedia compatibility using Linked Data as well as NLP tool interoperability. Furthermore, lists of link surface forms as well as the number of occurrences per surface form per link have been extracted as a secondary resource.

1.1. Wikipedia Abstract Properties

The abstracts represent a special form of text. The articles have an encyclopedic style (Nguyen et al., 2007), describing the topic by relating them to other entities often explicitly linked in the text. Following the official Wikipedia guidelines on article style and writing[1], the first sentence usually tries to put the article in a larger context. Thus Wikipedia abstracts can be understood as "Is-A"-corpus, defining an entity in relation to a broader class of related entities or an overarching topic. This interesting property is exemplified by the fact that by clicking the first link in a Wikipedia article, one will eventually reach the article on "Philosophy" for 94,5% of all articles[2].

Wikipedia guidelines on linking[3] prescribe that:

1. every word that is needed to understand the article should be linked,
2. links should only appear in the text once, and
3. the article topic itself should not be linked in its own article.

These guidelines manifest themselves in certain properties of the corpus:

[1]Wikipedia Manual of Style, Lead section, Opening paragraph: `http://en.wikipedia.org/wiki/Wikipedia:MOSBEGIN#Opening_paragraph`

[2]Data compiled by Wikipedia user "Ilmari_Karonen" using English Wikipedia articles from 26 May 2011: `http://en.wikipedia.org/wiki/User:Ilmari_Karonen/First_link`

[3]Wikipedia Manual of Style, Linking `https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#Principles`

1. Most major concepts important to the topic at hand will be mentioned in the text and their article will be linked.

2. If a concept has been linked, repeat mentions of it will not be linked again.

3. The article topic itself is often mentioned in the text, but never linked.

While the first property is crucial to be able to use the data as training corpus, the second and third properties have been adressed in the conversion process to guarantee a high-quality language resource.

2. Related Work

Wikilinks and their surface forms are one data source used to train entity linking tools such as DBpedia Spotlight (Mendes et al., 2011) or AGDISTIS (Usbeck et al., 2014). The articles themselves serve as a source of contextual information to (Cucerzan, 2007) in their named entity disambiguation approach. They have also been used for named entity linking evaluation (Hachey et al., 2013) and wikification (Cheng and Roth, 2013). (Nothman et al., 2012) present a small subset of Wikipedia articles with entity annotations[4] manually checked for correctness in the CoNLL format. NIF corpora are being used for evaluation in the GERBIL entity annotator benchmark (Usbeck et al., 2015).

3. Conversion Implementation

Although Wikipedia article texts can be acquired using the official Wikipedia API, it is recommended to not use it on a large scale as a courtesy to the Wikipedia project, so it cannot be used for large scale extractions. Wikipedia XML dumps[5] provide an alternative, but contain the articles in Wiki markup[6], a special syntax that is used to format Wikipedia articles and add further data. Besides text formatting, like representing a `==Heading==`, it also features calls to external LUA scripts and Wikipedia templates, making it a very tedious and Wikipedia language-specific task to implement a tool to render Wikipedia articles just like Mediawiki[7] does. To our knowledge, no tool exists that implements all templates and LUA scrips used in Wiki markup, making Mediawiki the only tool available to produce high-quality text from Wiki markup. Therefore a local Mediawiki instance as mirror of the Wikipedia was installed and configured as first part of the extraction pipeline, tasked with rendering the Wiki markup to plain text.
Figure 1 shows the data flow of the pipeline used. Central to the extraction was the DBpedia extraction framework[8] an open source framework to convert Wikipedia data to LOD.

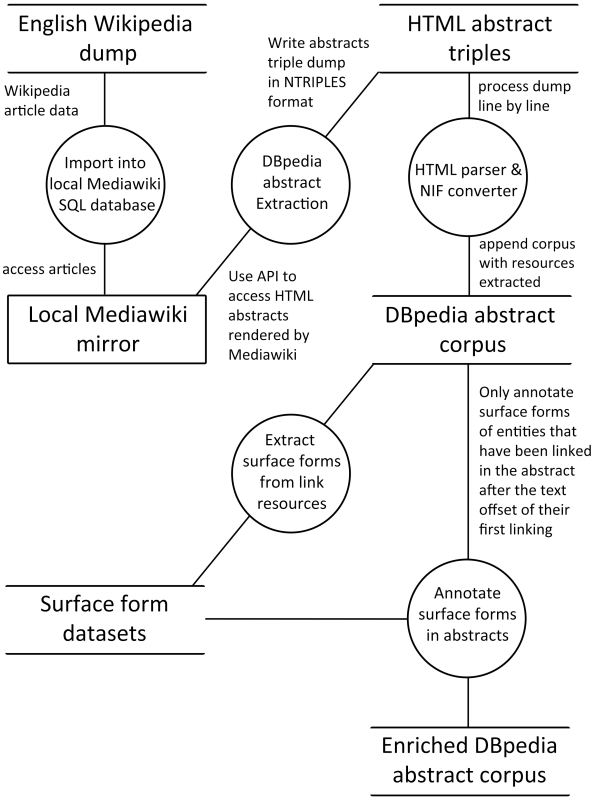

Figure 1: Data flow diagram showing the data conversion process

For the extraction of abstracts, the framework uses Mediawiki's API. For every page to extract data from, an HTTP request is made to the locally hosted API with the parameters `action=parse` and `section=0` to obtain the complete HTML source of the first section of each article[9].
Running the framework using this extractor and the local Mediawiki mirror creates one file in ntriples format per language, containing a triple of the following form for each article:

```
<$DBpediaUri> <http://dbpedia.org/ontology/abstract>
    "$htmlAbstract".
```

These triples present an intermediary data format that only serves for further extraction of HTML abstracts and the respective DBpedia URI. The HTML abstract texts were then split into relevant `paragraph` elements, which were converted to valid XHTML and parsed with a SAX-style parser.
All child elements of the relevant `paragraph`s where traversed by the parser. If a `#text` element was encountered, its content was appended to the abstract string. If an `a` element with attribute `href` was encountered, a `Link` object was created, containing the start and end offsets of the `#text` child element of the `a` element, as well as the surface form and the linked URL. Thus, links are anchored to the text position they are found at. Other elements were skipped to exclude `table`s and `span`s containing various

[4] `http://schwa.org/projects/resources/wiki/Wikiner`
[5] `http://dumps.wikimedia.org/`
[6] `http://en.wikipedia.org/wiki/Help:Wiki_markup`
[7] Wikipedia's software, `http://www.mediawiki.org/`
[8] `https://github.com/dbpedia/extraction-framework`
[9] Example: `https://en.wikipedia.org/w/api.php?action=parse&section=0&prop=text&page=Leipzig&format=xml`

other information not representable as plain text. Special consideration was given to Mediawiki specific elements used for visual styling. For example, `kbd` elements were not skipped, because they are used to style the text they contain as if it was a keyboard letter.[10] Thus they contain important content easily representable as text. After completely parsing the paragraphs, the resulting strings where concatenated to the abstract string, the `Link` objects created were aggregated. Using simple string processing, the resulting data was written to disk in NIF 2.1 in the turtle format before consuming the next line, producing the abstract corpus.

Using the finished corpus, surface forms of all links including their total number of occurrence in the corpus were extracted as a secondary resource. For example, in the English surface form dataset, the record {Berlin; http://dbpedia.org/resource/Berlin; 9338} indicates 9.338 occurrences of the entity "Berlin" and includes the link to the corresponding DBpedia resource. We provide these surface forms for each language as complementary datasets[11].

Finally, the corpus was enriched using these surface forms to account for Wikipedia linking guidelines described in Section 1.1. Because the topic itself is never linked, each abstract was enriched by linking occurrences of surface forms of the article topic (identified by its URI) to the article entity in the abstract texts. For example, in Listing 4., the first occurrence of "Pizza" is linked to its respective DBpedia resource, although it is not linked in the Wikipedia article, being the article's topic.

Furthermore, in the Wikipedia, each entity is only linked once in the text, subsequent mentions are not linked. We therefore annotate all surface forms of an entity linked in the text after its first mention. For example, in Listing 4., subsequent mentions of "bread" would not be linked in the article, but are linked in our corpus. Should a subsequent mention already be linked, that is, it is found in a text span already covered by another link, no new link is established. A typical example are phrases that contain the article topic but link more specific articles. In the scope of our example, the occurrence of "pizza" in a phrase "pizza stone" that links another article, would not be additionally linked. We reason that, where a link is established by Wikipedia editors, it will be more likely correct than our automatic enrichment efforts.

To enable users to judge the correctness of annotations better and differentiate between human and machine annotators, links established by enrichment have been differentiated from links made by Wikipedia editors using the `prov:wasAttributedTo` property. Original Wikipedia links are marked `prov:wasAttributedTo <http://wikipedia.org>` while our enriched links are marked `prov:wasAttributedTo <http://nlp.dbpedia.org/surfaceforms>`.

[10] For example the mentions of keyboard keys on this page: `https://en.wikipedia.org/wiki/Delete_key`

[11] `http://downloads.dbpedia.org/current/ext/nlp/abstracts/surfaceforms/`

4. NIF Format

A `nif:Context` resource was established for each article, containing the article abstract in the `nif:isString` property, as well as the string offsets denoting its length and the URL of the source Wikipedia page. For each link, another resource was created, containing the surface form in the `nif:anchorOf` property, its position in the string of the referenced `nif:Context` resource, as well as the URL of the linked resource via the property `itsrdf:taIdentRef`. Thus the identity of the link's surface form and the linked resource is made explicit, leading to the disambiguation of the link text by URL. An example resource can be found in Listing 1.

```
<http://dbpedia.org/resource/Pizza/abstract#offset_0_112>
  a nif:String , nif:Context ;
  nif:isString """Pizza ([ˈpittsa]) is an oven-baked flat bread typically topped with a tomato sauce, cheese and various toppings."""^^xsd:string;
  nif:beginIndex "0"^^xsd:nonNegativeInteger;
  nif:endIndex "112"^^xsd:nonNegativeInteger;
  nif:sourceUrl <http://en.wikipedia.org/wiki/Pizza> .

<http://dbpedia.org/resource/abstract#offset_0_5>
  a nif:String , nif:RFC5147String ;
  nif:referenceContext <http://dbpedia.org/resource/Pizza/abstract#offset_0_112> ;
  nif:anchorOf """Pizza"""^^xsd:string ;
  nif:beginIndex "0"^^xsd:nonNegativeInteger ;
  nif:endIndex "5"^^xsd:nonNegativeInteger ;
  prov:wasAttributedTo <http://nlp.dbpedia.org/surfaceforms> ;
  a nif:Word ;
  itsrdf:taIdentRef  <http://dbpedia.org/resource/Pizza> .

<http://dbpedia.org/resource/abstract#offset_40_45>
  a nif:String , nif:RFC5147String ;
  nif:referenceContext <http://dbpedia.org/resource/Pizza/abstract#offset_0_112> ;
  nif:anchorOf """bread"""^^xsd:string ;
  nif:beginIndex "40"^^xsd:nonNegativeInteger ;
  nif:endIndex "45"^^xsd:nonNegativeInteger ;
  prov:wasAttributedTo <http://wikipedia.org> ;
  a nif:Word ;
  itsrdf:taIdentRef  <http://dbpedia.org/resource/Bread> .
```

Listing 1: Example resource of the abstract corpus

5. Validation

Due to the size of the data and the diversity of the HTML that had to be processed, a special effort was made to validate the data. Validation was 3-fold:

1. The Raptor RDF syntax parsing and serializing utility 2.0.13[12] (raptor2-utils) was used to make sure the turtle files are syntactically correct RDF.
2. The GNU tools iconv[13] together with wc[14] was used to make sure the files only contain valid unicode codepoints. Testing was performed by dropping wrongly

[12] `http://librdf.org/raptor/`

[13] `http://www.gnu.org/savannah-checkouts/gnu/libiconv/documentation/libiconv-1.13/iconv.1.html`

[14] `https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html`

Language	Abstracts	Entity links	Triples
Dutch	1,740,494	11,344,612	114,284,973
English	4,415,993	39,650,948	387,953,239
French	1,476,876	11,763,080	116,205,859
German	1,556,343	15,859,142	153,626,686
Italian	907,329	7,705,247	75,698,533
Spanish	1,038,639	11,558,121	111,293,569
All	11,135,674	97,881,150	959,062,859

Table 1: Corpus statistics

encoded characters and comparing the number of characters afterwards to the character counts of the original files. This was necessary because the corpus contains citations from various languages and IPA sequences denoting phonology, that also heavily rely on unicode for presentation.

Iconv is typically used to convert text files from one format to another. However, starting it with the `-c` parameter makes it discard inconvertible (thus wrongly encoded) characters. Doing a character count before and after the iconv execution and comparing the number of characters would show that characters where dropped. This procedure again was run in a script on all files of the corpus, considering the unicode characters valid after this step.

3. RDFUnit[15] was used to ensure adherence to the NIF standard. (Kontokostas et al., 2014) provides a comprehensive description of the procedure. RDFUnit is a Java framework for test-driven evaluation of Linked Data quality. It uses SPARQL test queries generated from constraints given by an ontology as well as manually defined test cases. Using the framework with the ontologically defined constraints given by NIF, the corpus was validated for the correctness of property domains and ranges, as well as datatypes.

   In addition, manual test cases as described in (Kontokostas et al., 2014) were used to ensure that the `nif:anchorOf` string of each `nif:String` resource (i.e. the link annotations) equals the part of the respective abstract text indicated by begin and end offset of said string. Thus, the correctness of link positions and surface forms have been tested and verified for each link.

6. Access, Persistence, Maintenance

The corpus is available on the DBpedia downloads server at `http://downloads.dbpedia.org/current/ext/nlp/abstracts/`. Example files are available for all languages[16]. There is currently no Linked Data version, because we don't believe this contributes to its intended purpose for large-scale processing. The surface forms are available in seperate TSV files for each language[17].
The DBpedia Project guarantees the accessibility and persistence of the presented resources. The corpus generation and developement is integrated into the DBpedia release cycle and will be maintained accordingly.

7. Use Cases

7.1. Training NER Systems

Named entity recognition systems perform entity spotting, entity linking and entity classification. For each individual task, appropriate training data is required. For training entity spotting, texts with annotation mentions are needed. For entity linking, surface forms and their corresponding knowledge base link is needed to generate a list of candidates for linking. Finally, for entity classification, classes of each entity have to be known. The DBpedia abstracts provide all required information for training these tasks. For training entity spotting, the dataset provides the underlying textual content with a list of entity mentions and their exact location in the text. The entity linking task can be supported with the surface forms dataset compiled from the DBpedia abstracts.The entity types required for training classification can be then easily retrieved from DBpedia and the DBpedia Ontology.

7.2. Large Scale Multilingual Gazetteer

Maintenance and validation of gazetteers as well as keeping them up-to-date is an expensive and time consuming tasks. Moreover, in gazetteers listing organizations, musicians, events, etc. new and previously unknown entites pop up continuously. Wikipedia editors constantly provide new entities, validate them and maintain the existing list of entities. The DBpedia abstracts can be used to generate up-to-date gazetteer lists with latest entity names, which are validated and maintained by humans. Even more, gazetteers can be generated for different languages complemented with various inflected forms of their names.

8. Conclusion and Future Work

We presented a large, multilingual corpus generated from enriched Wikipedia data that makes annotated Wikipedia abstracts in six languages available in bulk for NLP processing. Including intellectually annotated entity mentions and enriched with mentions generated from extracted link surface forms, this corpus presents an improvement upon the existing DBpedia abstract texts. We are currently using both, the original and the enriched corpus for training and comparative evaluation of Named Entity Linking tools. We are also including further languages for subsequent releases with the goal of providing the corpus in all 125 DBpedia languages. Immediate language candidates for inclusion are Arabic, Czech, Japanese, Polish and Portuguese. Finally, we will include annotations for basic NLP tasks, such as sentence detection and tokenization to further ease NLP processing.

[15] `http://aksw.org/Projects/RDFUnit.html`

[16] For example English: `http://downloads.dbpedia.org/current/ext/nlp/abstracts/en/leipzig_en.ttl`

[17] `http://downloads.dbpedia.org/current/ext/nlp/abstracts/surfaceforms/`

9. Acknowledgement

This work is supported by grants from the EU's 7th Framework Programme provided for the project LIDER (GA no. 610782), the H2020 Programme provided for the project FREME (GA no. 644771) and the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) for the SmartDataWeb Project (Förderkennzeichen 1MD15010A).

10. Bibliographical References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2008). DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Cheng, X. and Roth, D. (2013). Relational inference for wikification. In *EMNLP*, pages 1787–1796.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716.

Hachey, B., Radford, W., Nothman, J., Honnibal, M., and J., C. (2013). Evaluating entity linking with wikipedia. 194:130–150.

Hellmann, S., Lehmann, J., Auer, S., and Brümmer, M. (2013). Integrating nlp using linked data. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*.

Kontokostas, D., Brümmer, M., Hellmann, S., Lehmann, J., and Ioannidis, L. (2014). Nlp data cleansing based on linguistic ontology constraints. In *Proc. of the Extended Semantic Web Conference 2014*.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM.

Nguyen, D. P., Matsuo, Y., Hogan, A., and Ishizuka, M. (2007). Relation extraction from wikipedia using subtree mining. In Robert C. Holte et al., editors, *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1414–1420. The AAAI Press.

Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2012). Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Usbeck, R., Ngonga Ngomo, A.-C., Röder, M., Gerber, D., Coelho, S., Auer, S., and Both, A. (2014). Agdistis - graph-based disambiguation of named entities using linked data. In *The Semantic Web - ISWC 2014*.

Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Waitelonis, J., and Wesemann, L. (2015). GERBIL – general entity annotation benchmark framework. In *24th WWW conference*.