DHTK: The Digital Humanities ToolKit

Davide Picca[†], Mattia Egloff[†]

[†]University of Lausanne

Abstract. Digital Humanities have the merit of connecting two very different disciplines such as humanities and computer science. In this paper we present a new python library, The Digital Humanities ToolKit (DHTK), whose scope is to provide a fast and intuitive tool for a large-scale study of large literary databases by leveraging on some well-known semantic knowledge resources. In addition, DHTK has the ambition to go beyond textual resources to integrate other human resources such as images (e.g., paintings, comics, etc) or sounds (e.g., music, transcripts, etc). DHTK is a collaborative project and we invite anyone who has the interest or some ideas in this context to reach out to us.

1 Introduction

In recent years, Digital Humanities (DH) have received a particular attention in both the humanities and computer science communities. Although automatic tools for text or image analysis have already been created for some time, many of these tools have technical difficulties or scientific objectives that do not properly fit the needs of the humanities community. In fact, one of the major limitations in the study of literary works is not only the access to a wider range of works but also the ability to analyze such resources in an extensive way given the intrinsic cognitive limits of the human being. In this paper we present a new python library whose scope is to overcome these limits by providing a fast and intuitive tool for a large-scale study of large literary databases by leveraging on some well-known semantic knowledge resources.

The Digital Humanities ToolKit (DHTK)[1] was conceived with the intent to offer a tool which is able to meet the needs of both communities. Inspired by the NLTK library [3], on which our work partially relies, DHTK is written in python and proposes similar objectives which are detailed in section 2. In particular, DHTK's main purpose is to provide the human scientist with a tool that leverages on the main semantic repositories as DBpedia to complete annotation and search for metadata (e.g., the year of the first edition, main characters, book categories, etc.). In this paper we describe DHTK, which we have developed in conjunction with a course taught at the University of Lausanne. Specifically, we describe its architecture, its modules and some case studies to better highlight its potential and its use. As DHTK is a work-in-progress, we conclude this presentation by exposing future improvements.

[1] DHTK will be freely accessible and anyone is free to participate to the project by giving its technical development contribution.

2 Design Criteria

In order to guarantee compatibility with the main NLP tools, we decided to write DHTK in python. From a programming point of view, DHTK takes into account some of the scope's criteria already listed in *Loper and al.* [3] and includes one additional criteria specific to DH. In particular, since DHTK is born with the aim of offering the researcher the opportunity to exploit the semantic resources available in the LinkedOpen Data (LOD) [1], one important DHTK's feature is its modularity. Thanks to this feature, DHTK can be expanded using other resources of the LOD independently without interfering with the pre-existing modules. In addition to the requirements of DHTK programming already mentioned above, the following subsection provides a list of further specifications of our work.

2.1 Requirements

- **Digital Humanities Oriented**. This is the main and exclusive feature to DHTK. The toolkit has been conceived to exploit and treat works coming from human science. Thus, access to corpora from literary or visual arts has a greater importance than the corpora's processing in itself for which other libraries, like NTLK, have been conceived.
- **Ease of use**. The main purpose of the library is to facilitate the exploitation of LOD resources such as Gutenberg.org and DBpedia, thereby being accessible to researchers with a more modest experience in programming. In fact, DHTK is conceived as a high-end library that provides APIs that can be easily recalled as we will see in the section 4. Given the simplicity of DHTK, it could be adopted by universities as a tool for computer science courses specifically designed for humanists.
- **Modularity**. We conceived the library following the KISS principle "Keep It Simple and Straightforward." Our objective is to make each module self-contained to the extent possible so that we can easily add higher modules as needed.
- **Efficiency**. Since we deal with large database as Gutenberg and DBpedia, we privilege efficiency and time-effectiveness over the coherence of the programming language.
- **Extensibility**. As for NLTK, our tool's main feature is extensibility. In our library we focus on Semantic Web resources such as LOD, which counts on extensive resources. DHTK aims to create a simple and constant interface to allow its extension to new modules and resources.
- **Documentation**. The toolkit, its data structures and its implementation are carefully documented with all the nomenclatures and acronyms carefully selected. The main purpose of the documentation is not only to facilitate the use, but also to help future developers who wish to contribute to the project.

3 Modules

The DHTK toolkit is composed of three independent modules, organized according to the logic task they perform. The logic modules are: *common modules, text resources, metadata search and NLP Processing.*

3.1 Common Modules

The common modules' main purpose is to ensure a continuity and a structural consistency between the various textual resource modules that will be added over time. They are composed of classes that define the general concepts of *Author, Book, Corpus, TextRepository*, and the Fuseki and RDFPro [2] wrappers used to handle RDF files.

3.2 Text Resources

This module is designed to contain the textual resources available on the LOD. We initially focused on Project Gutenberg because of its relevance in the humanist community and we found a lack of automatic tools available in order to exploit this specific resource. The main purpose of this module is to facilitate access to texts. At the time of writing this article, in Table 1 we report some basic statistics on data obtained from the results automatically crawled by DHTK.

# of books	# of authors	# of bookshelves	# of book categories
54975	19236	246	15106

Table 1. Basic statistics on Gutenberg Project

3.3 Metadata search

One of the most important aspects for a humanist is not just the availability of texts but also the metadata coming with texts. Unfortunately, not all repositories have complete information such as the date of the first publication or the first publisher. For example, Project Gutenberg does not store the original publishing date of its books, which can represent a problem if a corpus needs be delimited to a decade of published literature. DBpedia, thanks to its encyclopaedic nature, helps to rethink these elements. To this end, we have created the metadata search module on DBPedia that allows to complete and expand the missing information.

3.4 NLP Processing

This module aims to integrate existing NLP tools into other libraries. For the time being, DHTK integrates entirely the NLTK library to which this work is inspired.

4 Usage

After having provided an overview of the different logic modules, this section explains their usage. First of all, we will explain the usage procedure. Then, we will show two concrete examples: a search in the Gutenberg catalog and a metadata search in DBpedia.

4.1 How to use DHTK: the Use Case workflow

As we mentioned earlier, DHTK is a work-in-progress project and, for the time being, the user can search for authors or works in the RDF Gutenberg Catalog that DHTK has previously loaded into a local instance of Fuseki[2]. If the information returned are incomplete, the user has the possibility to query against DBpedia in order to get further data to complete annotation and search for metadata (e.g., the year of the first edition, main characters, etc.). All metadata available in DBpedia on a specific work or author are in principle retrievable. Once the corpus has been created, the user can use NLTK library to perform text processing and storing it a local database provided by DHTK. Our local database is an instance of PostgreSQL.

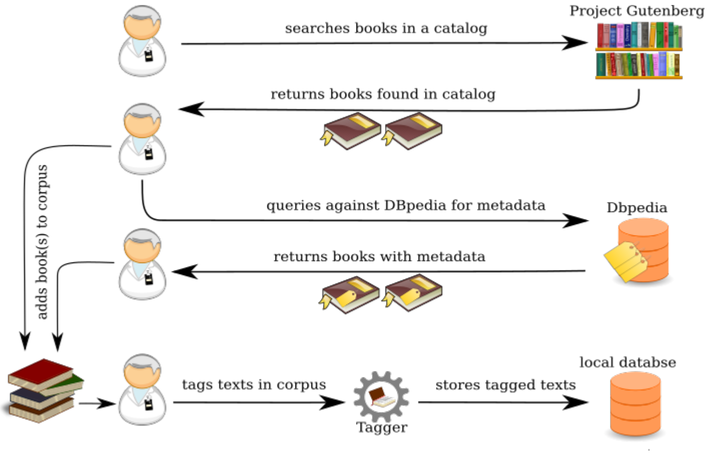
searches books in a catalog
Project Gutenberg
returns books found in catalog
adds book(s) to corpus
queries against DBpedia for metadata
Dbpedia
returns books with metadata
local databse
tags texts in corpus
Tagger
stores tagged texts

Fig. 1. The workflow schema

4.2 Some examples

As already mentioned, the following examples are aimed at showing the specific aspects that demonstrate the suitability of this library for researchers in DH. In particular, in collaboration with colleagues who teach literature, we have

[2] https://jena.apache.org/documentation/fuseki2/

identified at least two main topics that are generally of interest to the humanists' community. The first is the automatic building of corpora and the second is the search for metadata on the work to complete missing information like the list of main characters.

Example 1: Searching in Gutenberg catalog

The first example shows the easiness of building automatic corpora using DHTK. In this example, we want to build a corpus based on the author "Jane Austen." It would be possible to build corpora based on subject or bookshelf for example. Once Austen's works have been retrieved, we specify the languages we are interested in and the tool returns a list of all books in those specific languages. We can finally decide to download them in a local repository in order to process them into NLTK.

```
#Search by author
books = gs.search_by_author("Jane", "Austen")
#Retrieve only books in English
books = [gs.book_from_book_id(book["bookid"])
for book in books if book["language"] == "en"]
#Build the corpus
corpus = Corpus("Austen",
description="The books of Jane Austen",
corpora_path=corpora_path, book_list=books)

#Print the list of books in the corpus
corpus.print_book_list()
#Output:
0 Jane Austen Northanger Abbey
1 Jane Austen Lady Susan
...
#Download all texts in a local
directory corpus.download_book_corpus()
#Output:
[' 'Jane_Austen-Mansfield_Park.txt',
'Jane_Austen-Pride_and_Prejudice.txt', ...]
```

Example 2: Get metadata from DBpedia

This example shows how to obtain metadata from DBpedia in an easy way. In only few lines we can retrieve the missing information from Dbpedia and obtain the list of characters or the categories to which the work belongs.

```
#You can now get metadata using the BookID
book = gutenberg_search.book_from_book_id
('http://www.gutenberg.org/ebooks/2489')
```

```
dbpedia_metadata = DbpediaMetadata()
dbpedia_uri= dbpedia_metadata.search_book_uri(book)
dbpedia_metadata.get_book_metadata(book)
#Output:
'characters':
{'dbpedia': 'http://dbpedia.org/resource/
List_of_Moby-Dick_characters'}
subjects':
['http://dbpedia.org/resource/Category:1851_novels',
'http://dbpedia.org/resource/
Category:19th-century_American_novels',
'http://dbpedia.org/resource/Category:Allegory']
#complete list of 'characters' omitted for lack of space
```

5 Conclusion

In this paper we have described DHTK, a work-in-progress library aiming to provide non-computer science specialists with a tool to access textual resources available on LOD. Being a work-in-progress, only Gutenberg and DBpedia are available for the moment. We are working to integrate other repositories like Europeana [5] and Yago [4]. In addition, DHTK has the ambition to go beyond textual resources to integrate other human resources such as images (e.g., paintings, comics, etc) or sounds (e.g., music, transcripts, etc). DHTK is a collaborative project and we invite anyone who has the interest or some ideas in this context to reach out to us.

References

1. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
2. Francesco Corcoglioniti, Marco Rospocher, Michele Mostarda, and Marco Amadori. Processing billions of rdf triples on a single machine using streaming and sorting. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 368–375, New York, NY, USA, 2015. ACM.
3. Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
4. Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web (WWW07)*, pages 697–706, New York, NY, USA, 2007.
5. Bjarki Valtysson. Europeana : The digital construction of europe's collective memory. *Information, Communication and Society*, 15(2):151–170, 2012.