

Towards a Neural Extraction Framework — GSoC 2024

[gsoc2024](#) [gsoc2024-ideas](#)

tsoru  GSoC admin / mentor

Feb 1

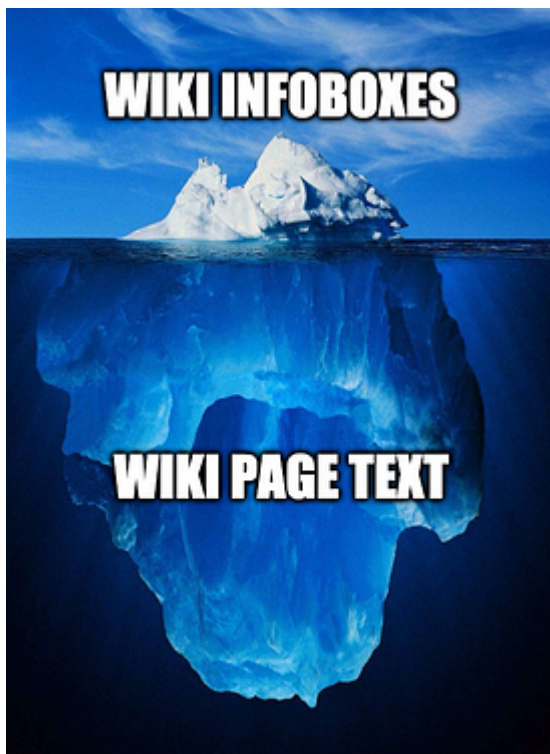
This project started in 2021 and is looking to its 4th participation in DBpedia's GSoC.

Description

Every Wikipedia article links to a number of other articles. In DBpedia, we keep track of these links through the `dbo:wikiPageWikiLink` property. Thanks to them, we know that the `:Berlin_Wall` entity (at the time of writing this) is semantically connected to 299 base entities.

However, only 9 out of 299 base entities are linked from `:Berlin_Wall` via also another predicate. This suggests that in the large majority of cases, it is not clear what kind of relationship exists between the entities. In other words, DBpedia does not know what specific RDF predicate links the subject (in our case, `:Berlin_Wall`) to any of the objects above.

Currently, such relationships are extracted from tables and the infobox (usually found top right of a Wikipedia article) via the [Extraction Framework](#). Instead of extracting RDF triples from semi-structured data only, we want to leverage information found in the entirety of a Wikipedia article, including page text.



The repository where all source code will be stored is the following:

[Skip to main content](#)

dbpedia/neural-extraction-framework
Repository for the GSoC project 'Towards a Neural Extraction Framework'



GitHub - dbpedia/neural-extraction-framework: Repository for the GSoC project...

Repository for the GSoC project 'Towards a Neural Extraction Framework' - GitHub - dbpedia/neural-extraction-framework: Repository for the GSoC project 'Towards a Neural Extraction Fram...

Goal

The goal of this project is to develop a framework for predicate resolution of wiki links among entities.

- During GSoC 2022, we employed **a suite of machine-learning models** to perform joint entity-relation extraction on open-domain text.
- Last year, we implemented **an end-to-end system** that translates any English sentence into triples using the DBpedia vocabulary.

However, the current algorithm still has the following issues. Now, we want to devise a method that can solve as many of them as possible.

1. When an RDF property representing the predicate is not found, our algorithm cannot make any suggestions for the creation of a new property.
2. The current models are not efficient enough to scale to millions of entities.
3. The extracted relations are not categorised with respect to their semantics (e.g. reflexive/irreflexive, symmetric/antisymmetric/asymmetric, transitive, equivalence).
4. The generated triples were not validated against the DBpedia ontology and may thus lead to inconsistencies in data.
5. Our algorithm should be able to adapt its output not only to the DBpedia vocabulary but to any specified one (e.g., SKOS, **schema.org**, Wikidata, RDFS, or even a combination of many).

Extraction examples

The current pipeline targets relationships that are explicitly mentioned in the text. The contributor may also choose to extract complex relationships, such as:

- **Causality.** (Addressed during GSoC 2021, but not completed.) The direct cause-effect between events, e.g., from the text

The Peaceful Revolution (German: Friedliche Revolution) was the process of sociopolitical change that led to the opening of East Germany's borders with the west, the end of the Socialist Unity Party of Germany (SED) in the German Democratic Republic (GDR or East Germany) and the transition to a parliamentary democracy, which enabled the reunification of Germany in October 1990.

extract: **:Peaceful_Revolution** ---dbo:effect--> **:German_reunification**

Skip to main content tract entity assigned to some agent, e.g., from the text

Messi won the award, his second consecutive Ballon d'Or victory.

extract: [:2010_FIFA_Ballon_d'Or](#) ---dbo:recipient--> [:Lionel_Messi](#)

Material

The contributor may use any Python deep learning framework and/or existing tool. The following resources are recommended (but not compulsory) for use in the project.

- The project repository linked above and the machine-learning models mentioned in the readme files found in each GSoC folder.
- [Last year's blog](#) to understand the project status quo.
- [Python Wikipedia](#) makes it easy to access and parse data from Wikipedia.
- [Huggingface Transformers for Natural Language Inference](#) can be extremely useful to extract structured knowledge from text or perform zero-shot classification.
- [DBpedia Lookup](#) is a service available both online and offline (e.g., given a string, list all entities that may refer to it).
- [DBpedia Anchor text](#) is a dataset containing the text and the URL of all links in Wikipedia; the indexed dataset will be available to the student (e.g., given an entity, list all strings that point to it).
- An [example of an excellent proposal](#) that was accepted a few years ago.

Project size

The size of this project can be either medium or large. Please state in your proposal the number of total project hours you intend to dedicate to it (175 or 300).

Impact

This project will potentially generate millions of new statements. This new information could be released by DBpedia to the public as part of a new dataset. The creation of a neural extraction framework could introduce the use of robust parsers for a more accurate extraction of Wikipedia content.

Warm-up tasks

- Get familiar with SPARQL on the [DBpedia endpoint](#) .
- Understand the [science behind relation extraction](#) .
- Run and understand the pipeline implemented [last year](#) .

Mentors

[@tsoru](#) , [@zoelevert](#) , TBD

[smilingprogrammer](#)

Feb 2

[Skip to main content](#)