Wordify: Bridging Multilingual Knowledge Bases for Seamless Entity Disambiguation

Fábio Araújo de Sá
up202007658@up.pt

Marcos Rafael Peixoto Aires
up202006888@up.pt

Pedro Pereira Ferreira
up202004986@up.pt

January 4, 2025

Abstract

Wordify is a multilingual entity disambiguation system leveraging Web Semantics and Linked Data principles to address challenges in identifying and linking entities across diverse knowledge bases. By using technologies such as RDF, SPARQL, SKOS, and OWL, Wordify bridges data from sources like DBpedia and Wikidata to resolve ambiguities and detect false friends. Its user-friendly interface simplifies entity searches, supports language-specific filtering, and adheres to Tim Berners-Lee's principles for structured, interconnected data. This paper discusses Wordify's architecture, key functionalities, and limitations, highlighting its potential to improve cross-lingual knowledge base interoperability.

1 Introduction

In Web Semantics and Linked Data contexts, it is common to handle a significant number of entities derived from the Web. Moreover, those are stored in various Knowledge Bases (KBs), such as the WikiData [1] and DBPedia [2], containing various entities, which majority of them are related to each other. In the worst cases, the entities are associated with the equivalent ones in several languages.

However, it's difficult to link these types of data. In fact, there's no simple way to connect similar information across KBs because the way the information is organized in the KBs varies. Also, an entity may have multiple entries because it has multiple values in the same or different knowledge bases, which makes it unclear what the entity is.

To overcome the aforementioned challenges, it would be ideal to build a system capable to link these entities across the different KBs, so that a computer could disambiguate them. As a result, it was built Wordify. Wordify is a system that disambiguates existing entities, possibly in different languages, using Web Semantics and Linked Data principles [3], concepts, and technologies. It will help a computer and humans to perceive which entity is being referred to, if it is ambiguous in a certain context, or to consult false friends between languages.

This paper presents a proposal for a Wordify application to address the ambiguity problem. It covers the motivation, requirements, and state-of-the-art solutions. The application architecture is detailed in Section 5, highlighting the interconnectedness of frameworks and tools. Section 6 discusses the results of the proposed solution, as well as challenges and limitations. The paper concludes in Section 7 with the future work, accompanied by references and annexes.

2 Motivation

Humans always tried to communicate in a clear and simple way with others, as well as with computers. However, it is a critical, inherent problem of the languages used by humans to communicate, as they are ambiguous. As a result, the entities present in the Web are also equivocal in some cases.

It is easy to think of a word, or something that can have different meanings, for different contexts: for instance, when someone is talking about a "balance", this word can mean the state in which opposing forces, or can also mean the object used to measure weights.

Another problem faced with natural languages is the fact that the same word can also have divergent meanings in different languages (false friends): the word "citrine" in some contexts, can refer to a yellow gemstone or a yellowish color. Similarly, the word "tuna" in English refers to a group of fishes, while in Portuguese, it refers to a musical academic group. As a consequence, it becomes difficult to connect the same word to distinct entities, especially when used in different domains.

With the exposure of this problem, it is clear there is the necessity for a system that can manipulate a knowledge base in a manner that elucidates the entities' search, regarding the multi-language characteristics and contexts. To this end, the Wordify system was developed.

3 Functionalities & Requirements

In light of the emphasis placed on comprehending the significance of our system and the rationale behind the relevance of our solution in the preceding section, the following one will describe the key functionalities of our solution.

Firstly, the basic capability of our system is to link various entities to the input search term. Thus, by means of example, if a user searches for "tuna", our prototype links this word with the entities that are associated with it. Moreover, various definitions, as well as their respective languages, are linked to the entities retrieved. If the word "tuna", is associated with two entities, each of those can be connected with one or more definitions, each of which is linked to a certain language. Figure 1 provides a generic example of the aforementioned flow.

Moreover, the system must be capable to filter the search results by a language selected by the user. If none is selected, it will be used english by default on the entity retrieving process. For instance, if the user wants to look for the entity "chat" in french, they must choose the "French" option, when they start their search using Wordify.

Furthermore, our system offers two other functionalities, namely entity disambiguation and detection of false friends. The former consists in showing the different meanings that a certain word can have in a specific language. The latter consist in words that are written in the same way in different languages (multilingual homographs) but have distinct meaning. For example, the word "citrine" has meaning both in italian and english. However, in italian it refers to a mineral, and in english to a shade of the color yellow.

In addition to these features, it is possible to see what are the most likely languages the entities may belong to. This functionality calculates the ratio between the number of times a language is associated to the retrieved entities, and the sum of languages for all the retrieved entities, when a search occurs.

Our system adheres to the Tim Berners-Lee's four principles [3], which serve as essential guidelines for publishing and interconnecting structured data on the web. These principles support the evolution of the Semantic Web, promoting the creation of a data web that is machine-readable, interoperable, and richly interconnected.

The first one consists in using Uniform Resource Identifiers (URIs) to identify resources, as URIs are the foundation for identifying and referencing resources unambiguously on the web. The second principle lies in using HTTP URIs, allowing users and machines to retrieve data. This makes resources discoverable, reusable, and linkable across datasets. The third one aims to provide useful information in standard formats according to the Resource Description Framework (RDF) [4], considering that the returned data should describe the resource and its relationships to other resources. Finally, the last highlighted principle by Tim Berners-Lee is centered on including links to other URIs to connect data, enabling navigation and discovery across datasets, as these links create a connected web of data, similar to how hyperlinks connect web pages.

4 Existing Approaches

This section details three related existing works that also aim to address entity disambiguation on the Web.

- **DBPedia Spotlight [5]**: is an entity disambiguation tool that uses DBpedia to identify and link entities in texts. It applies Natural Language Processing (NLP) techniques to connect terms to URIs.

- **Wikidata Query Service [6]**: provides an interface for SPARQL [7] queries, allowing the retrieval of structured data about entities in Wikidata, facilitating data integration and disambiguation across different resources and languages. Besides the SPARQL interface, it also permits the user to visualize queries as tables, maps, graphs, and timelines, making the data more comprehensible.

- **YAGO [8]**: semantic knowledge base that organizes knowledge in an RDF ontology and provides links to various sources, such as DBpedia, Wikidata and WordNet [9]. It is used for entity disambiguation and building semantic relationships across multiple contexts. These semantic relationships can be hierarchical (e.g., "is-a") and non-hierarchical (e.g., "located-in", "works-at").

Those works can, in fact, show to the user in a simple and clear way what one or more entities can refer to. However, their users need to know how those entities are represented, as well as a solid background of the languages used in this context, especially SPARQL and RDF. As a result,ordify provides a more user-friendly interface, allowing users to search for those entities, without needing to know how they are organized, neither using a structured code to make that query.

5 Architecture

The proposed architecture for Wordify's development is illustrated in Figure 2. To meet the system requirements, the application leverages a Docker [10] deployment for portability, a web interface to facilitate end-user interaction with data, and SPARQL for querying resources in DBpedia and Wikidata. It employs Simple Knowledge Organization System (SKOS) [11] to organize concepts, catalog synonyms, and manage hierarchies for ambiguous terms, enabling seamless integration of different data sources. Additionally, Web Ontology Language (OWL) [12] is used to differentiate word interpretations, resolve ambiguities, and identify false friends across entities.

5.1 RDF Graph Structure

Given a user-provided word in a specific language, the system performs SPARQL queries on the selected knowledge databases to retrieve the possible entities associated with the input and, for each entity, its meaning in the available languages. The results, returned in RDF format, are then used to construct a graph, also in RDF, which supports more structured queries aligned with Wordify's needs and requirements. The structure of the resulting graph is outlined in Figure 3.

The graph has its root at the user's input, cataloged using "SKOS.Concept" as the primary concept. Then, for each entity identified, it associates the entity to the user input using the predicate "SKOS.Related". Subsequently, for each of these entities, it links all the definitions found, and for each definition, the associated language. The predicates "SKOS.Lang" and "SKOS.Definition" are employed for these associations. Finally, the knowledge database consulted to locate the entity is linked using the predicate "OWL.Source". Where applicable, the predicates "OWL.FalseFriend" and "OWL.Ambiguity" are also applied between entities. Thus, both SKOS and OWL were used to meaningfully link different elements from various sources. The data transformation process for inclusion in this graph follows a proposal aimed at minimizing duplicate data as much as possible.

5.2 Computing Ambiguities

From the graph created and detailed in the previous subsection, it is easy to obtain pairs of entities for the same input where ambiguities occur. An ambiguity arises between two entities, as shown in Figure 4, when there is a language in the system that returns different meanings for the same input.

5.3 Computing False Friends

Given the nature and structure of Wordify's internal graph, searching for pairs of entities that are false friends is also straightforward. False friends occur when entities with different connotations are found between two different languages. An example is present in Figure 5.

5.4 Computing Top Languages

The presentation of the top five languages in percentage in Wordify is ensured through a simple query to the leaves/languages, considering all the entities in the system resulting from the user's input, followed by mathematical computation, as shown in Figure 6. This feature enables the system to infer, based solely on the data returned by the knowledge sources, the language in which the input might have been written. In other words, given a text containing the user's input word, the system can compute the probable language of that text.

6 Discussion

With the Wordify architecture, the Tim Berner-Lee's Four Principles [3] are satisfied successfully. In fact, it is possible to perceive the accomplishment of the first two: all the entities are identified with a unique URI. Moreover, it provides useful information in standard formats, as the entities are all structured in RDF. Furthermore, the user can use HTTP URIs to look up for data, since each entity has a URI associated to it (/search/<entityname>/), as well as they can access to other URIs present in each entities' page, to see the respective ambiguities and false friends.

There are some limitations that were considered before the inception of Wordify. Firstly, the vast number of potential entities that could be ambiguous, both within and across different languages, poses a computational challenge. To mitigate this and reduce processing time, we limited the number of entities retrieved.

Moreover, there are instances where a word may exist in one language but cannot be automatically associated with another representation in another entity in a different language. For instance, "saudade" means missing someone in portuguese, but there is no direct english translation. Unfortunately, there is no direct solution to overcome this challenge with the current Wordify's implementation.

Although our system is able to disambiguate entities and present false friends, it faces some limitations, which we consider in this section as well.

The first one is related to the difficulty of finding definitions in specific languages. The search feature of our prototype supports only seven languages, limiting the range of available language options to search in. This limitation can be observed in Figure 7.

The second limitation pertains to the integration of data from the two knowledge sources utilized, DBpedia and Wikidata. When a user searches for the same entity and applies a source filter, the entity retrieved from Wikidata differs from the one retrieved from DBpedia. This discrepancy is evident in their distinct meanings, top five languages, and ambiguities, as illustrated in Figure 8.

The third limitation of our system relates to the inability to identify regionalisms due to the absence of detailed geographical data for the countries whose languages are being searched. For instance, as illustrated in Figure 9, our system cannot distinguish between "fino" and "imperial", two Portuguese terms specific to Porto and Lisbon, respectively. This limitation arises because the prototype lacks geographical data about specific regions within countries.

Finally, users are also limited to search for single words in our system, as it is not tailored for searches with more than one word, as it can be seen in Figure 10.

7 Conclusions

Wordify demonstrates the potential to enhance cross-lingual entity disambiguation and knowledge base interoperability. By leveraging Web Semantics and Linked Data principles, the system offers a user-friendly interface that simplifies complex operations such as linking entities across diverse knowledge bases and resolving ambiguities. Despite its success in adhering to standards such as RDF and OWL and providing functionality for detecting false friends, Wordify is limited by its support for a small subset of languages, difficulties in addressing regionalisms, and the inability to process multi-word queries.

Future improvements could focus on expanding language support, incorporating detailed geographical metadata to better handle regionalisms, and enabling multi-word input processing. Addressing these limitations will further enhance Wordify's role as a tool in bridging multilingual knowledge bases, fostering better understanding, and enabling seamless integration of the data of Web.

References

[1] WikiData. Wikidata knowledge database. Available at https://www.wikidata.org/wiki/Wikidata:Main_Page, last accessed in November 2024.

[2] DBPedia. Dbpedia knowledge database. Available at https://www.dbpedia.org/about/, last accessed in November 2024.

[3] Tim Berners-Lee. Semantic web road map, 1998. Available at https://www.w3.org/DesignIssues/Semantic.html, last accessed in December 2024.

[4] RDF. Resource description framework. Available at https://www.w3.org/RDF/, last accessed in November 2024.

[5] DBPedia. Dbpedia spotlight shedding light on the web of documents. Available at https://www.dbpedia-spotlight.org/, last accessed in October 2024.

[6] WikiData. Wikidata query service. Available at https://query.wikidata.org/, last accessed in October 2024.

[7] SPARQL. Sparql query language. Available at https://www.wikidata.org/wiki/Wikidata:Main_Page, last accessed in November 2024.

[8] Yago Project. Yago: A high-quality knowledge base, 2025. Available at https://yago-knowledge.org/, last accessed in October 2024.

[9] WordNet Knowledge Database. Wordnet. Available at https://wordnet.princeton.edu, last accessed in November 2024.

[10] Docker. Container application development. Available at https://www.docker.com, last accessed in November 2024.

[11] SKOS. Simple knowledge organization system. Available at https://www.w3.org/2004/02/skos/, last accessed in October 2024.

[12] OWL. Web ontology language. Available at https://www.w3.org/OWL/, last accessed in October 2024.

8 Annexes

On the following page.

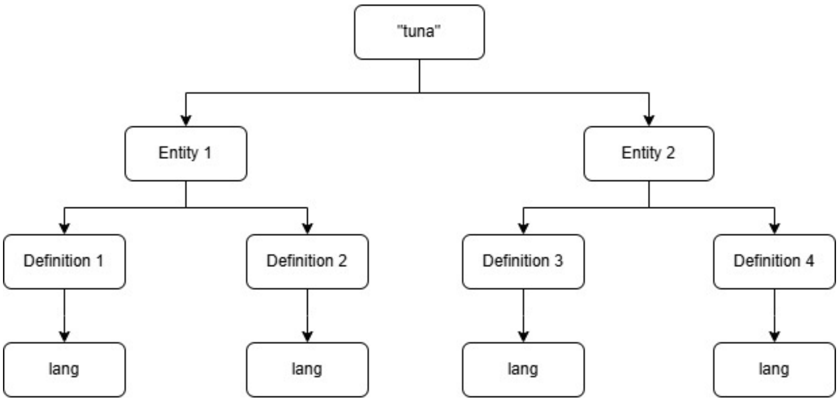


Figure 1: *Example of a retrieved entity's structured.*

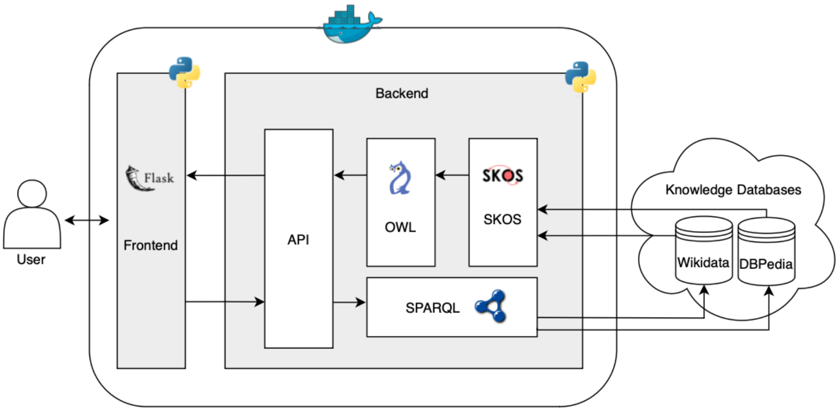


Figure 2: *Wordify's architecture*

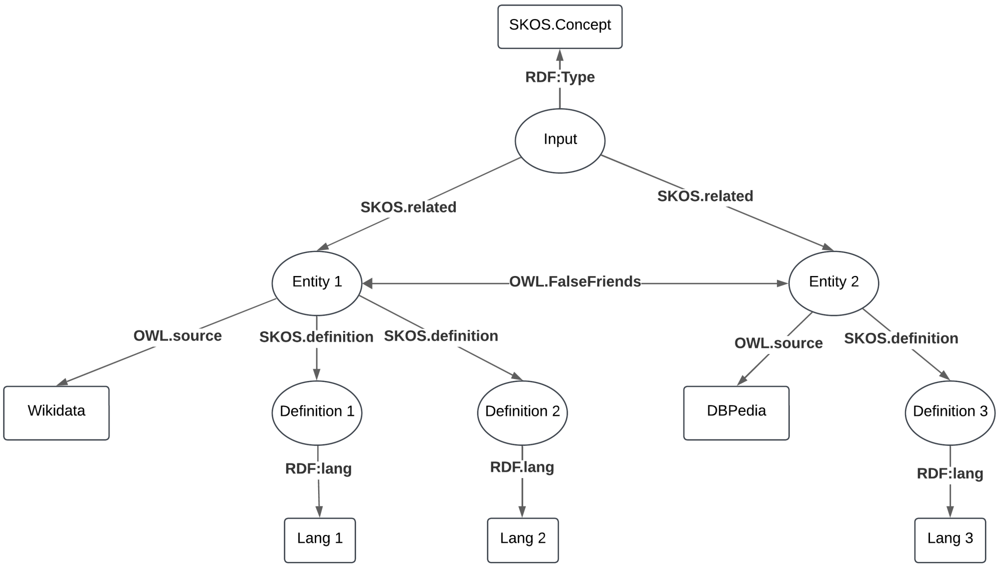


Figure 3: *Wordify RDF resultant graph structure*

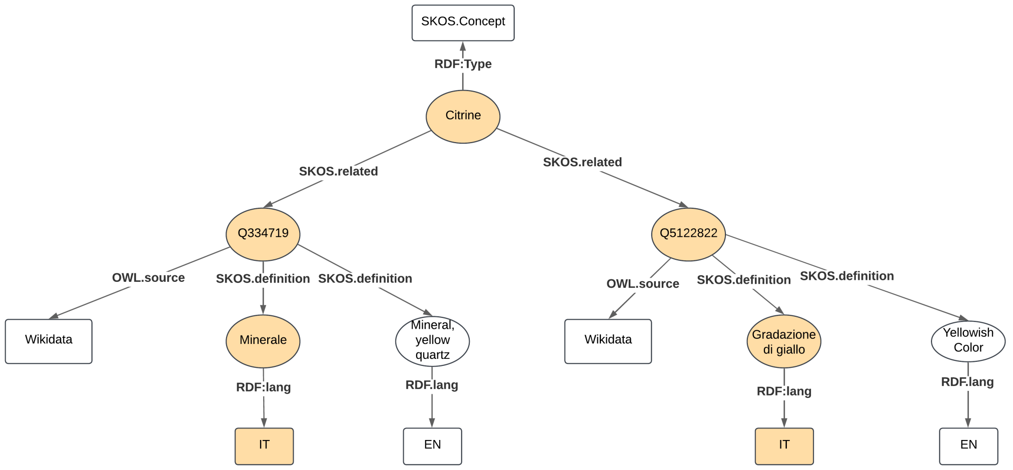


Figure 4: *Computing ambiguities using Wordify graph*

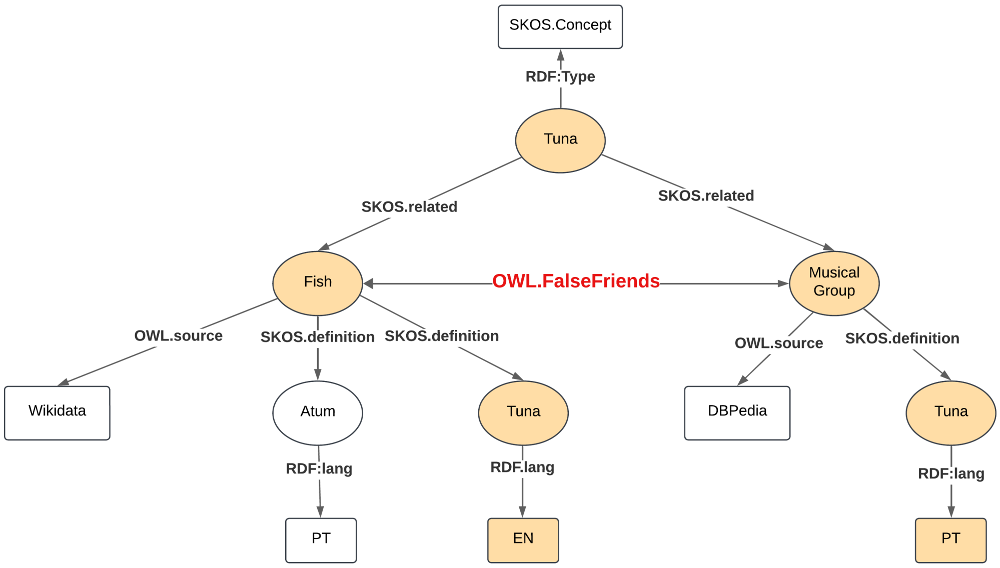


Figure 5: *Computing false friends using Wordify graph*

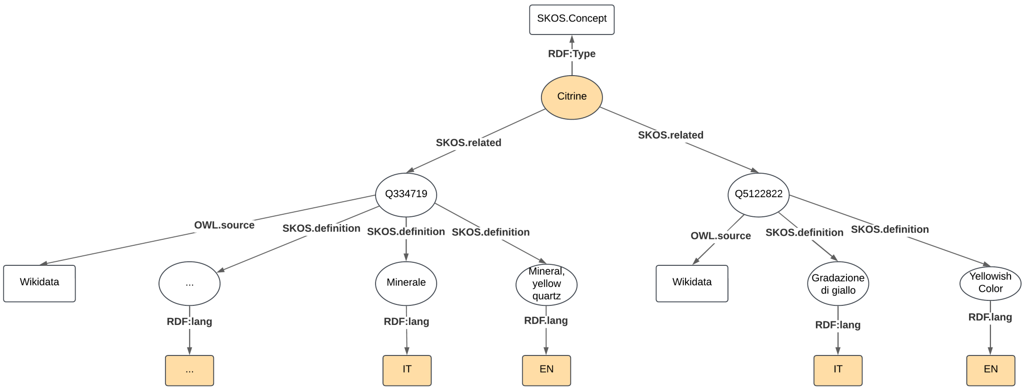


Figure 6: *Computing languages using Wordify graph*

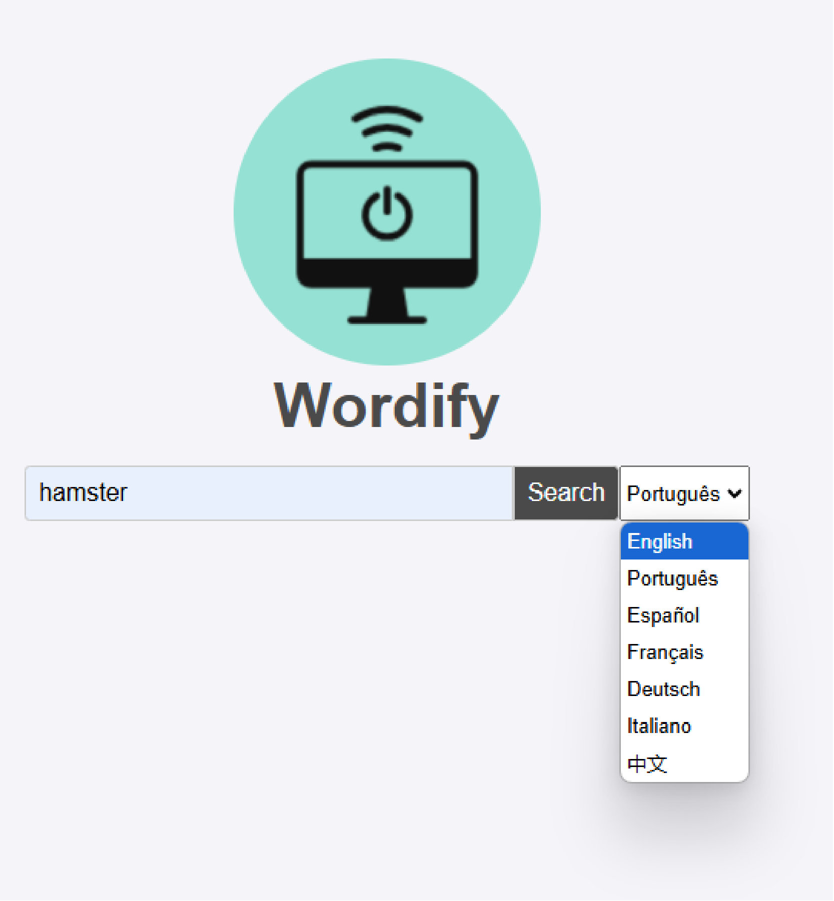
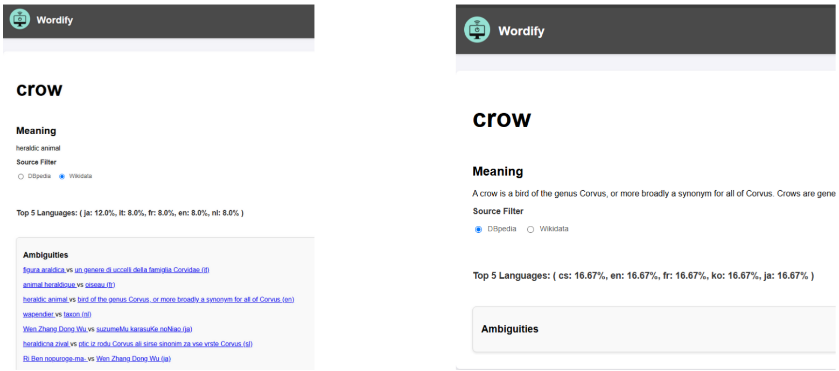


Figure 7: *The limited languages available for selection. The set of the languages were based on the most common languages in Europe, plus mandarin.*



crow

Meaning

heraldic animal

Source Filter

○ DBpedia ● Wikidata

Top 5 Languages: (ja: 12.0%, it: 8.0%, fr: 8.0%, en: 8.0%, nl: 8.0%)

Ambiguities

figura araldica vs un genere di uccelli della famiglia Corvidae (it)

animal heraldique vs oiseau (fr)

heraldic animal vs bird of the genus Corvus, or more broadly a synonym for all of Corvus (en)

wapendier vs taxon (nl)

Wen Zhang Dong Wu vs suzumeMu karasuKe noNiao (ja)

heraldicna zival vs ptic iz rodu Corvus ali sirse sinonim za vse vrste Corvus (sl)

Ri Ben nopuroge-ma- vs Wen Zhang Dong Wu (ja)

crow

Meaning

A crow is a bird of the genus Corvus, or more broadly a synonym for all of Corvus. Crows are gene

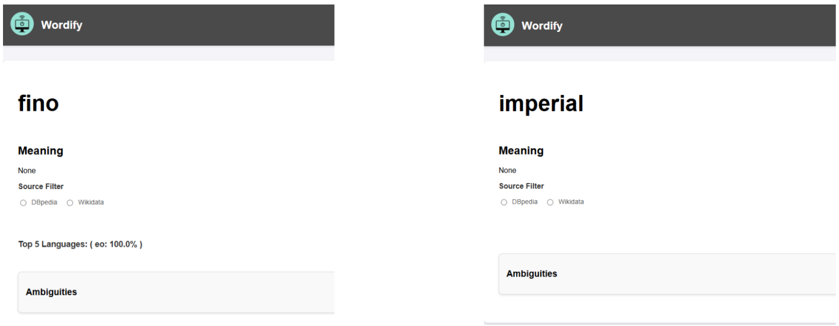
Source Filter

● DBpedia ○ Wikidata

Top 5 Languages: (cs: 16.67%, en: 16.67%, fr: 16.67%, ko: 16.67%, ja: 16.67%)

Ambiguities

Figure 8: *"Crow" example, exposing the incapability to cross data between the knowledge bases, as they show different results for the same entity.*



fino

Meaning

None

Source Filter

○ DBpedia ○ Wikidata

Top 5 Languages: (eo: 100.0%)

Ambiguities

imperial

Meaning

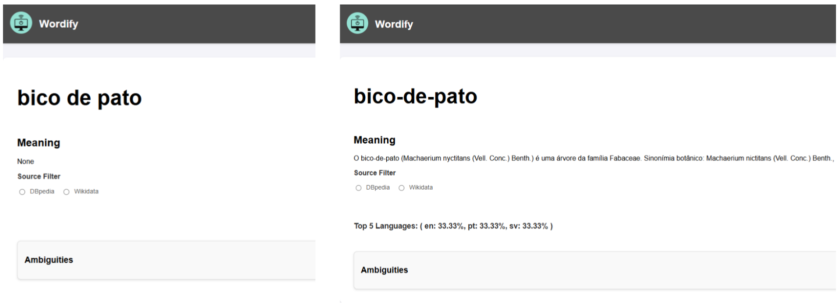
None

Source Filter

○ DBpedia ○ Wikidata

Ambiguities

Figure 9: *The "imperial" and "fino" example, exposing Wordify's incapacity to identify regionalisms.*



bico de pato

Meaning

None

Source Filter

○ DBpedia ○ Wikidata

Ambiguities

bico-de-pato

Meaning

O bico-de-pato (Machaerium nyctitans (Vell. Conc.) Benth.) é uma árvore da família Fabaceae. Sinonímia botânico: Machaerium nictitans (Vell. Conc.) Benth.,

Source Filter

○ DBpedia ○ Wikidata

Top 5 Languages: (en: 33.33%, pt: 33.33%, sv: 33.33%)

Ambiguities

Figure 10: *The "bico de pato" example, which shows that the system can only process single words.*