WikidataComplete – An easy-to-use method for rapid validation of text-extracted new facts applied to the Wikidata knowledge graph

Kunpeng Guo[1,2][0000−0002−0692−0057], Dhairya Khanna[3][0000−0002−4531−5351], Dennis Diefenbach[1,2][0000−0002−0046−2219], Aleksandr Perevalov[4,5][0000−0002−6803−3357], and Andreas Both[5,6][0000−0002−9177−5463]

[1] The QA Company, Saint-Etienne, France
[2] Laboratoire Hubert Curien, UMR CNRS 5516, Lyon, France
[3] Maharaja Agrasen Institute of Technology, New Delhi,India
[4] Anhalt University of Applied Sciences, Köthen, Germany
[5] Leipzig University of Applied Sciences, Leipzig, Germany
[6] DATEV eG, Nuremberg, Germany

Abstract. Wikidata is one of the most used knowledge graphs (KG) and it plays a vital role in the Semantic Web community. Many industries have integrated Wikidata into solutions dedicated to intelligent assistance, information retrieval, or knowledge integration. As one of the biggest KGs, Wikidata receives millions of edits every year. However, it is still far from complete. Generally, a natural workflow to ingest a new fact into Wikidata starts by searching the relative information in a free-text document collection (e.g., Wikipedia articles). This information is used to create a new fact (or update a fact) on Wikidata. The entire process is labor-intensive. In this paper, we present WikidataComplete, a plugin that facilitates Wikidata editors to contribute to the completion of the Wikidata KG. For the implementation of WikidataComplete, we integrated the latest question-answering (QA) technologies in order to extract the new facts. We embed our fact-ingestion workflow directly on the Wikidata entity page to make the insertion of facts smooth and efficient. Ultimately, WikidataComplete can be a handy tool for Wikidata contributors, and it has the potential to complete millions of missing facts in the Wikidata KG.

Keywords: WikidataComplete · Wikidata · Knowledge Graph Completion · Question Answering

1 Introduction

WikidataComplete tackles the task of *Knowledge Graph Completion* (KGC) which aims at adding missing relation for existing entities in a KG [5]. The task is particularly an important issue for community-based KGs such as Wikidata[7]. Despite the Wikidata KG receiving millions of edits each year[8] it is far from complete. For

[7] https://www.wikidata.org/

[8] https://www.wikidata.org/wiki/Wikidata:Statistics

instance, for the newspaper entities, Wikidata only lists the property *place of publication*[9] for 40% of them. There are two main paradigms to solve the task of KGC: (1) link prediction-based, which predicts missing relation based on the current graph structure [2]; (2) machine reading-comprehension technique that can leverage missing relations from free texts [3].

In this work, we present *WikidataComplete*, a Wikidata plugin that facilitates the process of fulfilling the missing facts in the Wikidata KG. The plugin follows the second paradigm to perform the KGC task. It includes a fact-verification step to have a human in the loop to guarantee the inserted fact is valid. Besides, the new ingested fact is accompanied automatically by evidence from where it was extracted, which is increasing the providence data quality too. Therefore, the new facts inserted by WikidataComplete into Wikidata KG will be both accurate and self-explanatory.

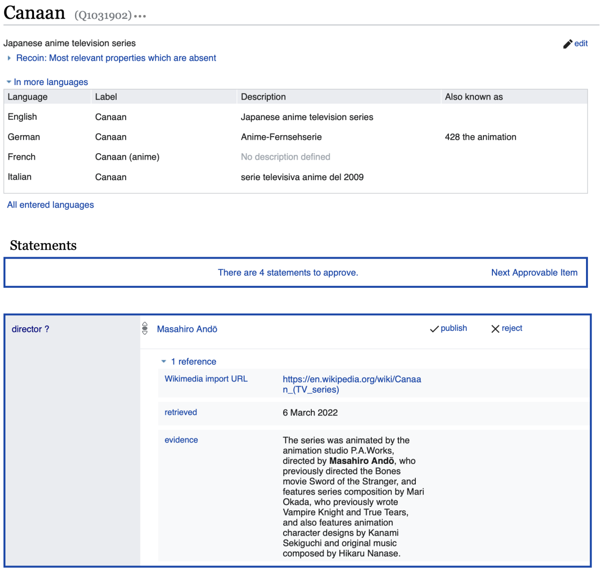
2 Demo

Finding and ingesting a new fact into Wikidata is a relatively labor-intensive process. For example, let us consider "Canaan", a "Japanese anime television series" (Wikidata URI: `https://www.wikidata.org/wiki/Q1031902`). By exploring the entity on Wikidata, it is not immediately clear if it is complete or not. Only after a more careful analysis (e.g., using the Recoin Plugin[10] [1]) one can detect that the property "director" (`P57`) is missing. A Wikidata editor who wants to improve the KG needs to find a source that contains this missing fact. One natural choice is to go to the Wikipedia and to check if the missing information is available in the text of an article. By reading the corresponding Wikipedia article, one will identify the following paragraph "The series was animated by the animation studio P.A. Works, *directed by Masahiro Andō*, who previously directed ....". In the next step, the editor can start completing the missing fact. Identifying the property (director) is not problematic. since this was its original intention. On the other hand, identifying the object entity by its label can be difficult if it is ambiguous. This is the case for the label *Masahiro Andō* which could correspond to the following URIs `Q9128134` (anime director), `Q11451348` (Japanese animator) or `Q1982546` (Japanese association football player). After having correctly disambiguated the object entity, the editor can finally insert the identified statement and improve the completeness of the Wikidata Knowledge Graph.

WikidataComplete automatically addresses this workflow. While exploring Wikidata entities, editors will be directly be pointed to incomplete ones and new facts are ready to be reviewed. For the example, WikidataComplete directly proposes the fact depicted in Figure 1. One can see that it has identified that the entity is incomplete concerning to a certain relation. Thereafter, it identifies a text segment in Wikipedia containing the relevant information (see "evidence" statement), provides a disambiguated entity, and asks the editor only for publishing or rejection. To allow the editor to judge the fact, a source and evidence are provided.The source redirects the user to the Wikipedia page containing the answer in highlighted text. In case

[9] https://www.wikidata.org/wiki/Property:P291

[10] https://www.wikidata.org/wiki/Wikidata:Recoin



Canaan (Q1031902)

Japanese anime television series

Language	Label	Description	Also known as
English	Canaan	Japanese anime television series	
German	Canaan	Anime-Fernsehserie	428 the animation
French	Canaan (anime)	No description defined	
Italian	Canaan	serie televisiva anime del 2009	

There are 4 statements to approve.

director ? Masahiro Andō

Wikimedia import URL	https://en.wikipedia.org/wiki/Canaan_(TV_series)
retrieved	6 March 2022
evidence	The series was animated by the animation studio P.A.Works, directed by **Masahiro Andō**, who previously directed the Bones movie Sword of the Stranger, and features series composition by Mari Okada, who previously wrote Vampire Knight and True Tears, and also features animation character designs by Kanami Sekiguchi and original music composed by Hikaru Nanase.

Fig. 1. Screenshot of WikidataComplete showing a new fact for "Canaan"

of approval, both are inserted into the graph. This enables other editors to better trace back where the knowledge is coming from.

The code with instructions to activate the plugin can be found on GitHub.

3 Process

The workflow of WikidataComplete contains three main modules: *(1)* Incomplete Triplet Curation, *(2)* Dedicated Relation Extraction, *(3)* Target Entity Linking.
Incomplete Triplet Curation: A triple in a KG is composed of three main elements: Subject Entity, Property, and Object Entity. A triple is incomplete if for one subject entity (e.g., *"Canaan (Japanese anime television series)"*), the property *"director"* is missing from the Wikidata KG. WikidataComplete collects the missing triples via the process consisting of 3 steps:

- *Fix a class*: The plugin first fixes a class (i.e., *"anime television series"*) and collect the subject entities that match to it.
- *Find Most Frequent Properties*: The plugin analyzes the subject entities that belong to the *fixed* class to get the most frequent properties that exist among all of them. The plugin tries to complete the entities of this class with respect to these properties.
- *Resource Availability Verification*: The plugin fixes the class and the property and conducts a collection of missing triplets *(Subject - Property - ?)*. Before moving forward to *(2) relation extraction*, the triplets are filtered out if the subject does not have a corresponding Wikipedia page.

Dedicated Relation Extraction: This module first receives a collection of missing triplets. The problem of finding the missing object entity is solved with the task of question answering (QA) over free-text. The QA task takes a question and a paragraph and tries to identify in the paragraph the corresponding answer. The plugin downloads the Wikipedia article of the subject entity and treats it as our target passage. Then it constructs the questions by putting together the subject entity label and property entity label (e.g., "*Canaan director?*"). The QA model will indicate a set of candidates in the form of text spans for the object entity.

Object Entity Linking: The output of the *(2) Relation Extraction* module is a list of candidates for the object entity in the textual span format. For the fact ingestion, we need to link the spans to their corresponding entity in the Wikidata KG. This is achieved by a pre-trained KG linker model [4]. This module can help to filter out unreasonable answer choices made by the *(2) Relation Extraction* module.

Finally, the new triples together with snippets (pieces of evidence) from where they were extracted are obtained. This is presented in the UI of WikidataComplete as in Figure 1 for user approval or rejection.

4 Conclusion

In this paper, we introduced WikidataComplete, a Wikibase plugin that integrates Question-Answering technologies and human-in-loop verification strategy to help complete the Wikidata Knowledge Graph. The proposed workflow has the potential to add millions of missing facts in the Wikidata KG by extracting them from textual resources and reducing the required time investments for the Wikidata editors. We plan in the future to generalize our approach to other sources than Wikipedia, to increase its precision, and to apply it to other domains.

This work was partially funded by the Google Summer of Code program 2021.

References

1. Balaraman, V., Razniewski, S., Nutt, W.: Recoin: relative completeness in wikidata. In: Companion Proceedings of the The Web Conference 2018. pp. 1787–1792 (2018)
2. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge graph completion: A review. Ieee Access **8**, 192435–192456 (2020)
3. Han, X., Liu, Z., Sun, M.: Joint representation learning of text and knowledge for knowledge graph completion. arXiv preprint arXiv:1611.04125 (2016)
4. Kratzwald, B., Kunpeng, G., Feuerriegel, S., Diefenbach, D.: IntKB: A verifiable interactive framework for knowledge base completion. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 5591–5603. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020)
5. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence (2015)