Semantic Web -1 (2023) 1–14
DOI 10.3233/SW-233450
IOS Press

Psychiq and Wwwyzzerdd: Wikidata completion using Wikipedia

Daniel Erenrich
Etsy, Inc., Sunnyvale, CA, United States
E-mail: daniel@erenrich.net

Abstract. Despite its size, Wikidata remains incomplete and inaccurate in many areas. Hundreds of thousands of articles on English Wikipedia have zero or limited meaningful structure on Wikidata. Much work has been done in the literature to partially or fully automate the process of completing knowledge graphs, but little of it has been practically applied to Wikidata. This paper presents two interconnected practical approaches to speeding up the Wikidata completion task. The first is Wwwyzzerdd, a browser extension that allows users to quickly import statements from Wikipedia to Wikidata. Wwwyzzerdd has been used to make over 100 thousand edits to Wikidata. The second is Psychiq, a new model for predicting instance and subclass statements based on English Wikipedia articles. Psychiq's performance and characteristics make it well suited to solving a variety of problems for the Wikidata community. One initial use is integrating the Psychiq model into the Wwwyzzerdd browser extension.

Keywords: Wikidata, Wikipedia, browser extension, knowledge graph completion

1. Introduction

Wikidata is a collaborative knowledge base that is widely used as a source of structured data [19]. It is a multilingual, open-source database that is maintained by the Wikimedia Foundation and is closely linked to Wikipedia. Wikidata stores structured data about entities and links them to the Wikipedia versions in various languages about those entities.

Wikidata's ontology is community moderated and is primarily defined by two kinds of elements: items and properties. Wikidata items are abstract entities which represent conceptual or material entities. This includes people, places and ideas. Properties are relations between entities or other pieces of data. This includes relations, such as birthdate, father, place of birth or height. Properties and items are assigned codes of the form `P##` or `Q##` respectively.

Probably the most important properties in the Wikidata ontology are "instance of" (denoted `P31`) and "subclass of" (`P279`). These statements form the backbone of the Wikidata ontology and, as others have noted, are effectively equivalent to `rdf:type` and `rdfs:subClassOf` [6]. `P31` is used to express ideas, such as "Obama is a human" or "Rap is a music genre". `P279` is used to express ideas, such as "businesses are a kind of organization" or "film is a kind of artwork".

While Wikidata is a separate project from Wikipedia, a lot of cross pollination occurs between the two platforms and much of the new content on Wikidata is sourced from Wikipedia. Despite this, it is still the case that many high quality articles on Wikipedia have little or no information on Wikidata. As of November 2022, there are 359,581 items on Wikidata corresponding to English Wikipedia articles that have no `P31` or `P279` statements. Roughly 60 thousand of those items have no statements whatsoever and many more only have uninformative external IDs imported for them. This is an underestimate of the number of English Wikipedia articles that are not well represented in Wikidata because many articles do not even have a corresponding item. Because Wikipedia has generally stricter notability inclusion guidelines than Wikidata, these items Wikidata is lacking coverage of are especially important.

Quantifying the importance of the zero-context items can be approximated by considering the daily web traffic for English Wikipedia articles whose Wikidata items lack both `P31` and `P279` statements. On March 1, 2023 these main namespace articles together had over 6.1 million views[1] out of a total daily English Wikipedia traffic of 89.3 million.

Improving or automating the process of migrating information from Wikipedia to Wikidata would improve Wikidata greatly. Today manual human driven editing is already a minority of all work done on Wikidata. Nearly half of all edits to Wikidata are performed by bots[2] and a large fraction of the remaining human-driven edits are made using automated tooling. Despite this, millions of statements are likely missing from Wikidata [18]. While it has been found to be more up-to-date than other knowledge graphs [13], Wikidata lags behind English Wikipedia on many subjects.

Wikipedia is often able to absorb new facts or concepts before Wikidata does because of its larger and more active editor and user-base. English Wikipedia alone has over 3 times more active editors than Wikidata[3] and these statistics already overestimate the actual participation on Wikidata. This measure of “active editors” (five or more edits) greatly favors Wikidata’s editor count as some operations on Wikidata require many successive edits. Additionally, most Wikidata editors in June 2023 who used the Wikipedia or Wikidata user-interfaces only edited sitelink information. Sitelink information is useful in that it ties Wikidata items to Wikipedia articles but it does not contain any statements (the kind of structured information this paper is concerned with).

In a typical 24-hour period ending January 2, 2023, 967,292 edits were made to Wikidata but only 85,779 (8.9%) of those edits were made using one of the Wikidata web interfaces.[4] The rest were made either by bots or through various other tools. Most edits are either fully automated or semi-automated. Semi-automated edits are generally batch changes made by importing information or performing some clerical task. 526,297 (54%) edits in the 24-hour period were made using QuickStatements which is a tool for bulk editing.

To appreciably increase the speed of Wikidata progress even more (semi)automated tools are needed. Figure 1 shows that the ratio of Wikidata items to active users is increasing, indicating that the community’s size is not keeping up with the breadth of the Wikidata corpus.[5] While it is easy to just create more pages, curating a collection of this size is difficult. For a limited number of humans to curate a corpus of 100 million items better tooling is needed to more efficiently use their limited time.

The tools need to meet a number of requirements to be useful to the Wikimedia community. The most important of these requirements are: low latency, broad support for the most common Wikipedia patterns, flexibility to work with new Wikipedia content and high throughput of user actions. Low latency is critical because the tools are intended to be used in an online context and users will not tolerate a model that takes minutes to predict statements given an article. Broad support for Wikipedia patterns is important because the tool need to provide useful guidance in the most common cases. Flexibility is important because Wikipedia is constantly evolving and assumptions about, for example, the strict pattern of names of categories cannot be relied upon. For the tool to be high throughput it needs to require a minimal number of user interactions to perform common tasks.

This paper discusses the creation of two related tools to improve the process of importing data from Wikipedia into Wikidata: Wwwyzzerdd and Psychiq. Wwwyzzerdd is a user-interface browser extension that allows users to annotate Wikipedia links with properties and thus add new statements to Wikidata. Psychiq is a machine learning

[1] Using the pageview methodology described in https://dumps.wikimedia.org/other/pageviews/readme.html.
[2] https://stats.wikimedia.org/#/wikidata.org/contributing/edits/normal|bar|2-year|editor_type~anonymous*group-bot*name-bot*user|monthly
[3] https://stats.wikimedia.org
[4] https://wikidata.wikiscan.org/hours/12/stats
[5] https://stats.wikimedia.org

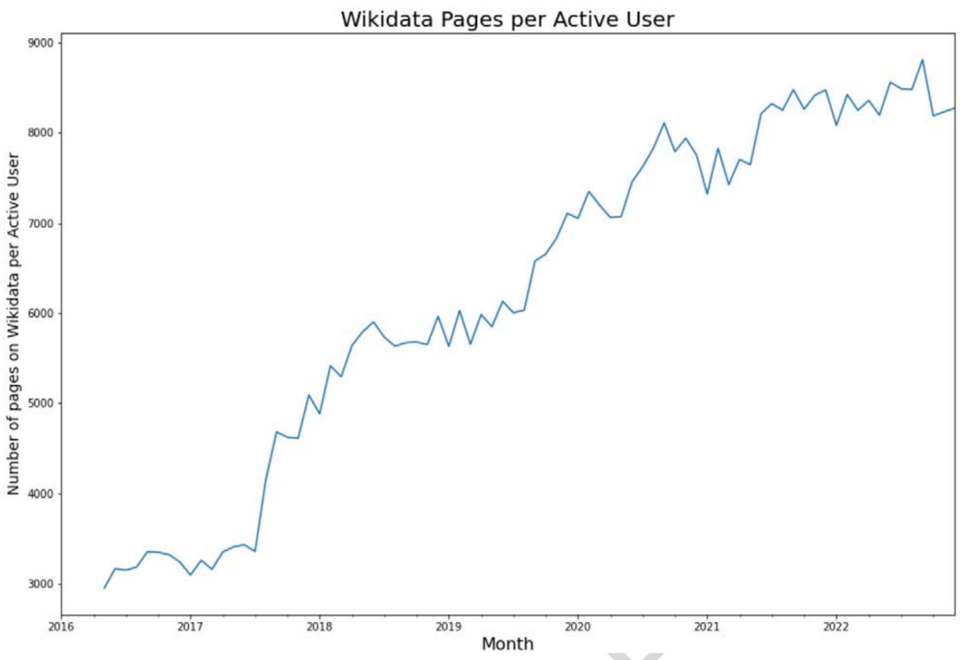


Fig. 1. The ratio of pages on Wikidata to active users has steadily increased, indicating that than the growth in articles has outpaced the growth in the active user base on Wikidata.

model that is integrated into the Wwwyzzerdd UI that suggests, on the basis of Wikipedia's content, new `P31` or `P279` statements to be added to Wikidata.

2. Related work

Tools in this space fall somewhere on an spectrum between exclusively human directed and fully automated. It has previously been noted that a combination of automated and human-directed edits together has a strong positive impact on item quality in Wikidata [14].

Many of these tools in the Wikidata ecosystem have no mentions in the published literature even though some are heavily relied upon.

2.1. Automated extraction

For fifteen years DBpedia has largely automated the process of structuring Wikipedia using semantic web and linked data technologies [8]. DBpedia primarily does this by having volunteers contribute to the "DBPedia Mappings Wiki"[6] which provides a mapping from fields on Wikipedia templates to ontology classes. While DBpedia's approach is effective for many Wikipedia articles it is fairly reliant on articles that use infoboxes. Many articles, especially less mature ones, do not have infoboxes for DBpedia to harvest.

Others have attempted bootstrapping fully automated extraction of semantic information from Wikipedia. KYLIN is one such effort [21]. KYLIN is similar to the work presented here on Psychiq in that it learns a classifier from article content to classes. This work differs from KYLIN in that Psychiq relies heavily on the Wikipedia category system while the paper for KYLIN describes the Wikipedia categories as, "so flat and quirky that utility is low". KYLIN also was only trained on four classes while Psychiq targets 1000.

[6]http://mappings.dbpedia.org/index.php/Main_Page

A much more ambitious effort is the Never Ending Language Learning (NELL) project [10]. The project aims to extract semantic meaning from the general web over an expanding number of learned relations and entities. The project appears to now be dormant and required training over 2,500 distinct learning tasks. NELL would be immensely overpowered for the use case considered here and may not meet Psychiq's latency requirements.

The Cat2Ax [4] system, like Psychiq, attempts to extract structured data from the category system that Wikipedia uses. Wikipedia has an extensive category system which is a hierarchy of classes that represent concepts, such as "Mountain ranges of Arizona". Cat2Ax differs from the work done for Psychiq as it targets the DBpedia ontology instead of the Wikidata ontology. Cat2Ax also attempts to infer both type axioms and relation axioms (in the parlance of Wikidata that is equivalent to targeting more than just `P31` and `P279` statements). It also does not leverage the articles' title information which is often critical for handling ambiguous cases and for which Cat2Ax employs a post-filtering step. One advantage Cat2Ax has is that in principle it is not limited to predicting high support types but for this paper's goal of prediction of types in the Wikidata ontology this is not particularly important. Cat2Ax and subsequent work were used to create a new knowledge graph rather than feeding into Wikidata [5].

Another automated attempt to populate statements is the Wikidata "NoclaimsBot". This community maintained bot looks at the templates used in a Wikipedia article and, based on heuristics created by editors, infers statements for the item. For example, if the template "Infobox river" is used then the bot infers that the item for that page is an instance of a river (`Q4022`). This bot's logic is simple and does not have mechanisms for resolving conflicts between multiple templates. The existence of this bot, however, impacts the quality of the training data in Wikidata. It means that much of the existing labels are being set using rigid rules. While it could, the bot does not set any references to indicate that the statements were inferred from templates used on Wikipedia. This means that there is no way to filter out automated statement additions from the training corpus. Since its inception in 2016, the NoclaimsBot has made 189,000 edits.[7]

In the same vein as the "NoclaimsBot" is the "Pi bot" [12]. While this bot does many kinds of tasks the relevant task for this paper is creating Wikidata items for new Wikipedia articles. Instead of being configured by the community, it is controlled by a series of heuristics encoded in python code. It uses logic such as "if the title includes 'list of' then the article is a Wikimedia list article (`Q13406463`)". It also has more advanced heuristics where it attempts to deduce if the page is a biography and if it is it attempts to populate attributes like birth date, death date, gender and occupation using even more complex heuristics. "Pi bot" has made more than 8 million edits though not all of those are using these heuristics.

Kian [16] is similar to Psychiq in that it is a neural network designed to predict Wikidata statements. It was written in 2015 and does not leverage pre-trained language models. Instead it relies on heuristic features over categories and trains a binary classifier for a given statement type. Kian was used to add 100,000 statements to Wikidata [17]. Kian was gamified into a human-in-the-loop labeling application using the Wikidata Distributed Game framework[8] to handle cases where the model was not sufficiently confident. This game, however, is no longer online and the tool is not maintained.

While there is a lot of opportunity for automated extraction to improve Wikidata there isn't consensus within the community about how to do so. Wikidata has thus far chosen not to import statement en masse from DBpedia even though doing so would be relatively easy. Efforts to add AI/ML inferred statements have been met with some skepticism over concerns over quality while simple heuristic solution have been preferred. There is a fear in the community of adding a large number of false statements which the human reviewers would never be able to audit. It is an open question how accurate a model would need to be to add newly inferred statements fully autonomously.

2.2. *Human-guided extraction*

Sztakipedia [3] is a browser extension that utilizes DBpedia Spotlight [9] to suggest improvements to Wikipedia. While these suggestions don't explicitly "structure" the data or edit Wikidata the kinds of suggestions made are similar. Its design of modifying the user-interface of Wikipedia is also similar to Wwwyzzerdd.

[7]https://xtools.wmflabs.org/ec/wikidata.org/NoclaimsBot
[8]https://wikidata-game.toolforge.org/distributed

QuickStatements is community maintained tool that enable users to generate an offline batch of edits to Wikidata and execute them as a group via a web interface. While it has high usage numbers it serves as a convenient replacement for the Wikidata Python-API and does not aid the actual information extraction process. Users are left to do the data processing using some other tool.

PetScan is another community maintained tool that allows users to create filters for pages on a Wikipedia and then add statements on all matching items using QuickStatements. For example, a user could create a filter for all pages under the category "2018 films" which also use the template "Infobox film" and add the statement "instance of film" (`P31 Q11424`) to all the items. These batch workflows are human curated but it's possible that without appropriate supervision incorrectly added statements would get through and cause bad category to statement associations to be inferred by Psychiq.

IntKB [7] is a human-in-the-loop system that uses natural language processing to suggest statements to add to Wikidata based on the text of Wikipedia. IntKB also has a companion Wikidata gadget which augments the user-interface to allow for quick editing of Wikidata.[9] While the tool is effective at extracting certain types of relations, it is particularly poorly suited for the important kinds of relations Psychiq is designed to extract. This is because instance/subclass relations are often not explicitly addressed in the opening text of an article (e.g. articles don't call people "humans"). Also, while the tool is designed for human-in-the-loop work its community adoption is low with fewer than 1000 edits being made using the tool since its inception in September 2021. Of a random sample of 3657 of IntKB's suggested statements, it suggested statements using `P31` or `P279` zero times.

The "Wikidata browser extension"[10] is the most similar tool to Wwwyzzerdd. However its focus is on extracting information from non-Wikipedia websites. When using the extension on Wikipedia, it has a few issues including not resolving redirects, not suggesting properties to link and not indicating which items are already linked. It has a significant amount of usage (roughly 70% of the daily edit volume of Wwwyzzerdd).

2.3. Gaps in related work

Of the existing automated extraction tools none meet all of the desired requirements. DBpedia is incompatible with Wikidata's ontology. KYLIN supports only support four classes and so doesn't support the breadth of Wikipedia's content. Cat2Ax neither meets the latency or flexibility requirements as it is intended for offline batch analysis of Wikipedia dumps. Cat2Ax also only leverages the category information which reduces its accuracy. Kian is unmaintained and only works for a small number of classes. "NoclaimsBot" is inflexible and based on strict rules for category and template naming. "Pi bot" is entirely heuristic and supports a limited number of types.

Of the human-guided tools most are unsupported or solve a different problem. Sztakipedia no longer seems to be available to install. QuickStatements and PetScan don't solve the problem of extracting structured data from a single Wikipedia article. IntKB almost never suggests `P31` or `P279` statements. The "Wikidata browser extension" isn't optimized for extracting data from Wikipedia and so has higher latency and requires many more clicks to achieve the same thing as Wwwyzzerdd.

Given this, there is a need for new solutions to the problem.

3. Implementation

3.1. Wwwyzzerdd

Wwwyzzerdd is a browser extension for Firefox and Chrome that modifies the user-interface of Wikipedia. It is designed to make transcribing data from Wikipedia to Wikidata as efficient as possible. The main idea behind Wwwyzzerdd is that Wikipedia contains rich data about items that are not already present in Wikidata. If Wwwyzzerdd can make the process of structuring the data take just a few clicks then the coverage of Wikidata could be improved.

[9] https://github.com/WikidataComplete/Wikidata-Complete-Gadget
[10] https://github.com/fuddl/wd

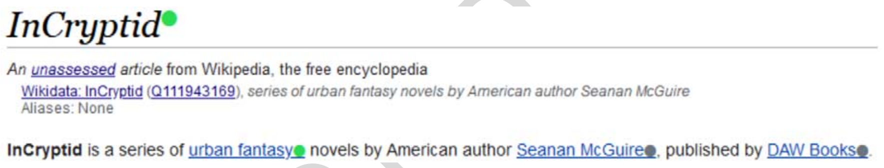
For example, a book might be properly tagged as a literary work but its authors and publishers may not be tagged in Wikidata. If the author is notable enough to have an English Wikipedia article, their name in the book's article will almost always be a link to the author's article. Wwwyzzerdd exploits the existence of this link.

Another common scenario is for the "NoClaimsBot" to populate a film's value for `P31` but not to import key people related to the works. Generally the director, producer, screenwriter and cast members of a film or TV series are notable enough to have their own pages on Wikipedia. So users could use Wwwyzzerdd to link those people to the item using the properties: `P161` (cast member), `P57` (director), `P58` (screenwriter) or `P162` (producer).

Wwwyzzerdd was implemented in Typescript [1] with the Parcel bundler[11] and using the yarn package manager.[12] The user-interface was built using React [13] with the Material UI component library.[14]

Using the Wikidata and Wikipedia APIs, Wwwyzzerdd interrogates all internal links in the current article and looks up the corresponding Wikidata items for them. It then renders an "orb" indicating if that item is already connected to the current item. If there is a property connecting the Wikidata item for the article to the item linked, the corresponding orb is green. Otherwise, the orb is gray. Figure 2 shows examples of both of these. Hovering over a green orb shows the list of properties linking the page to that item. If a user sees an entity which should be connected to the page's item they can click on the orb and select the relevant property. Figure 3 shows an example of this. Properties are either suggested to the user using Wikibase's integrated property suggestion algorithm (which suggests properties likely to be added the current item based on what's currently in Wikidata) or the user can search for a desired by property by typing in the search box. For repetitive workflows (e.g. repeatedly tagging different book items with the same author), Wwwyzzerdd remembers the last property an item was attached with and preferentially suggests it.

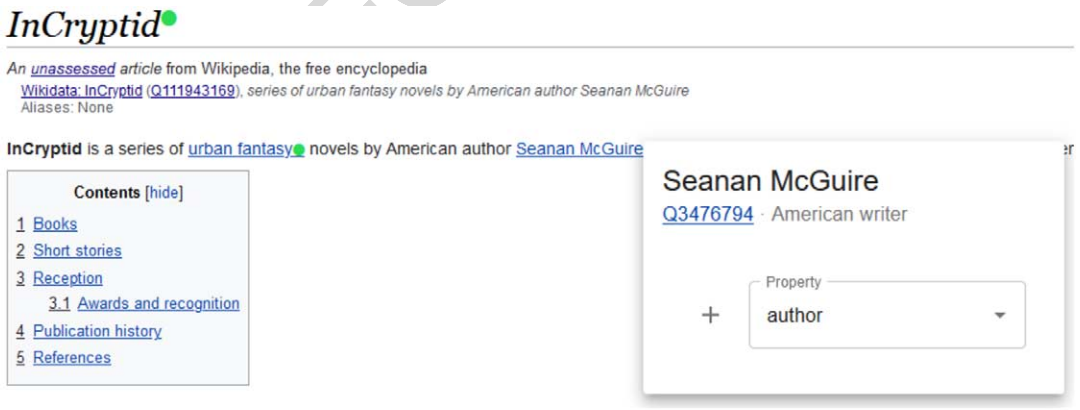
By relying on the already populated links, Wwwyzzerdd avoids the well-known problem of entity-linkage between the Wikipedia text and Wikidata. This can sometimes be problematic as Wikipedia editors will sometimes


InCryptid●

An unassessed article from Wikipedia, the free encyclopedia
Wikidata: InCryptid (Q111943169), *series of urban fantasy novels by American author Seanan McGuire*
Aliases: None

InCryptid is a series of urban fantasy● novels by American author Seanan McGuire●, published by DAW Books●.

Fig. 2. The Wwwyzzerdd user-interface showing linked entities (e.g. genre) in green and unliked entities (e.g. publisher and author) in gray.


InCryptid●

An unassessed article from Wikipedia, the free encyclopedia
Wikidata: InCryptid (Q111943169), *series of urban fantasy novels by American author Seanan McGuire*
Aliases: None

InCryptid is a series of urban fantasy● novels by American author Seanan McGuire

Contents [hide]
1 Books
2 Short stories
3 Reception
3.1 Awards and recognition
4 Publication history
5 References

Seanan McGuire
Q3476794 · American writer

Fig. 3. The Wwwyzzerdd user-interface shows a pop-up which allows the user to add a statement connecting the article with the target through an arbitrary Wikidata property.

[11] https://parceljs.org/
[12] https://yarnpkg.com/
[13] https://react.dev/
[14] https://mui.com/

not link common terms, such as the "United States". Adding support for entity linkage techniques will likely be added later.

It is important that Wwwyzzerdd shows contextual information, such as the description and label of the linked item. This is because it is very common for the text of the link and the target of the link to not match. For example, Wikipedia articles often will say something similar to "Susan is an American author" where American is a hyperlink. Sometimes that hyperlink goes to the page for the "United State of America" and other times it will go to the article for the "American" ethnic group. Which value is linked determines the correct property to use.

Wwwyzzerdd also checks all external links in the Wikipedia article. External identifier properties in Wikidata are tagged with URL match patterns (`P8966`). These patterns are regular expressions which match URLs for the identifiers source. So for example, `orcid.org/(0000-{4}-{4}-{3}.)` matches ORCID iD URLs and extracts the underlying identifier. Wwwyzzerdd attempts to match all the Wikipedia external links on the URL match patterns. If a match is found then an orb is shown offering to let these user attach the identifier to the Wikidata item using the corresponding property.

To use the extension the user clicks on one of the orbs as shown in Fig. 3. The extension uses the Wikidata API to suggest reasonable properties or the user can type to search. Once the user decides on a property and hits the plus button, Wwwyzzerdd makes two edits to Wikidata. First, it links the page's item to the target item through the selected property. Then it adds a reference to that statement indicating that the information was inferred by examining the current version of the Wikipedia article.

Wwwyzzerdd does not currently support adding qualifiers to statements but this is mostly needed for specific properties, such as "relative" (which should be qualified with "kinship to subject"). This may be solved in the future by adding custom support for relationships that require qualifiers (e.g. "uncle") which would automatically add the correct statements and qualifiers.

3.2. Psychiq

Psychiq's focus is exclusively on populating missing instance of (`P31`) and subclass of (`P279`) statements. This limiting decision was made for a number of reasons. First, as mentioned previously, `P31` and `P279` are the backbone of the Wikidata ontology and many other properties have constraints that require these values be populated.

Second, populating `P31` and `P279` statements is particularly difficult to do using Wwwyzzerdd alone. This is because it is rare for articles to link to items for their containing classes. For example, articles about people almost never link to the item for human (`Q5`) which is the correct Wikidata class. Many common classes for entities in Wikidata do not even have corresponding English Wikipedia articles. For example popular music songs are generally coded as `Q105543609` but there is no corresponding English Wikipedia article for this concept.

Third, once the values of `P31` or `P279` are known Wikidata's property suggestion mechanism becomes much more capable. If Wikidata knows that the item is about a human then it intelligently suggests the properties gender, occupation, country of citizenship and date of birth. If instead the item is about a business it suggests headquarters location, country, inception and website. The property suggester used on Wikidata only has this level of intelligence when `P31` or `P279` are set.[15]

To simplify the machine learning process Psychiq only considers the title of an article and its categories. English Wikipedia has an extensive category system which alone allows for strong predictive performance. It is important to consider the title because some articles cannot be correctly classified only using categories. For example "General aviation in the United Kingdom" is impossible to classify as "aspect in a geographic region" (`Q74817647`) by only considering its categories of "Aviation in the United Kingdom" and "General aviation". In the future the first sentence of the article, the full-text of the article or category hierarchy information may also be added as a feature to the model. For now this information is excluded because of a desire to keep the inference latency and cost low. The full-text also cannot be included because the Distilbert model used in Psychiq does not support context lengths that can span the length of an entire Wikipedia article.

As was done for Cat2Ax, Psychiq filters the set of categories that are considered when making predictions. Categories were filtered out if they contained strings:

[15] https://github.com/wikimedia/mediawiki-extensions-PropertySuggester

- Short description
- Articles with
- All stub articles
- Wikidata
- Noindexed pages
- Redirects
- All articles
- dates
- Wikipedia articles
- wayback links
- Pages containing
- Articles containing
- Articles using
- Articles needing

These are largely administrative categories that do not contain substantial semantic information. In contrast with Cat2Ax though, categories containing the string “stub” are preserved as they contain important topical context. For example, the category “`Swedish handball biography stubs`” indicates that the article is likely about a person.

A primary goal of Psychiq is that it be fast enough to use in interactive contexts. While it’s unlikely that the model will be small enough to embed and run directly in the Wwwyzzerdd extension it must be small enough to host cheaply. To that end a small and fast model was selected: Distilbert-base-uncased.

Distilbert-base-uncased [15] was selected because of its size, latency and capacity. Distilbert is small given its performance, just a couple hundred MBs of weights, and was trained over the Wikipedia English corpus which makes it a good fit for this application. It is pared down version of the BERT model [2] which, while no longer state-of-the-art, has become a standard NLP baseline. Distilbert and BERT both leverage transformers models. Distilbert was trained by “distilling” BERT to a similarly shaped model with fewer layers.

In practice the model is hosted on huggingface.com and the model runs on an average input in 35 ms once the instance is spun-up. Distilbert is trained exclusively on English so fine-tuning training/evaluation data for Psychiq is restricted to English Wikipedia.

Psychiq is restricted to predicting amongst the top 1000 most frequent `P31` or `P279` statements. Of all the English Wikipedia articles with `P31` or `P279` statements on the linked Wikidata items 93% of them use one of these top 1000 most frequent items. The one-thousandth most frequent statement occurred 356 times. Many of the items that do not use one of the 1000 most common statements would still be well covered by a top-1000 statements. For example, instances of “village of Japan” (`Q4174776`) would also be well represented by “village” (`Q532`).

The Hugging Face Transformers library [20] was used to train a sequence classification model for 1001 classes. This model feeds the text corresponding to a Wikipedia page through a BERT-like neural-network to produce an embedding of the text. This embedding is then fed into a final dense linear layer with a number of neurons equal to the number of classes and with a cross-entropy loss function.

As input to the model, we generate a text document for each Wikipedia page composed of a listing of all the categories separated by newlines followed by the title of the article. The order of the categories is left as-is in the article because Wikipedia assigns some small meaning to the ordering of the categories.[16] A sample document for the article “Bandaje Falls” would be:

Waterfalls of Karnataka
Tourist attractions in Dakshina Kannada district
Geography of Dakshina Kannada district
Bandaje Falls

[16] https://en.wikipedia.org/wiki/Wikipedia:Categorization#General_conventions

The corpus data processing was done using BigQuery. Freely available dumps of Wikidata and English Wikipedia were loaded into Google Cloud Storage and then imported into BigQuery. In particular, the corpus of “enwiki-categorylinks”[17] was used to map English Wikipedia articles to categories. The “wikidata-nt-all”[18] corpus was downloaded and split into two. One dataset was a mapping of Wikidata items to statements and one was a mapping from Wikidata items to English Wikipedia articles.

The 1000 most common values for the property of `P31` or `P279` for items with corresponding English Wikipedia articles were computed and each was assigned a unique label value. A single “example” was generated for each English Wikipedia article, excluding redirects, with a specific value for `P31` or `P279`. These examples were joined with the labels with a special 1001 label value introduced for statements outside of the top 1000.

The training and testing corpus of these documents and the accompanying labels is available on Hugging Face.[19]

The model is then asked to predict one of the top 1000 statements or none-of-the-above based on these documents. The correct prediction for that sample document would be “waterfall” (`Q34038`). Training was performed on Google Collab for a single epoch, due to budgetary constraints. Over 5.6 million examples were in the training set leaving another 10% out as a test set. The hyper-parameters for learning were a learning rate of 2e-5, a batch size of 32 and a weight decay of 0.01. The trained model is available for download on Hugging Face.[20] Scikit-learn was used to compute performance metrics [11]. Source code for downloading/preparing data and training the model is on GitHub[21]

Table 1 shows the key test set performance indicators. Qualitatively the performance is good enough for its intended human-in-the-loop use. This is especially true when considering the top-5 scenario where it is used in Wwwyzzerdd (users are shown the 5-top predictions from the model). Figure 4 shows the model’s accuracy in various top-K scenarios. The “guessing” baseline is the accuracy of always selecting the top-K most common labels. The confusion matrix indicates which pairs of statements the model is worst at distinguishing. The most confused pairs are presented in Table 2. Most of the pairs are self-explanatory and are the result of either very closely coupled concepts (a song and the single it was released on) or nested concepts (literary works are a kind of written work).

As noted by Shenoy [18], Wikidata often has trouble distinguishing between “subclass of” and “instance of” statements. This is corroborated by Psychiq’s confusion between instances of genes (DNA) and subclasses of protein-coding gene. An example of this is the gene the gene DYNC1I1 (`Q248215`). While it is often sensible for an item to be both an instance and subclass of other items (e.g. rap is both a subclass of music and an instance of music genre) it does not make sense for an item to be an instance and subclass of the same thing (in this case kinds of genes). Areas where the ontology could be improved can be found by examining Psychiq’s errors.

Another common issue occurs when Wikidata and Wikipedia disagree. For example, for many of the Wikidata items about genes the corresponding English Wikipedia article is actually about the protein that the gene codes for. The Wikidata item for the gene DYNC1I1 lists itself for the property “encoded by” (`P702`) which can’t be right because the item is a subclass of protein-coding gene and not a protein. When Wikidata and Wikipedia disagree what the topic is you cannot safely infer Wikidata statements from the Wikipedia article. This is a limitation of this approach.

Table 1

Psychiq test set performance metrics

Metric	Value
Accuracy	84.6%
Precision (weighted)	83.2%
Recall (weighted)	84.6%
ROC AUC (weighted, one-vs-rest)	0.996

[17] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-categorylinks.sql.gz
[18] https://dumps.wikimedia.org/wikidatawiki/entities/latest-all.nt.bz2
[19] https://huggingface.co/datasets/derenrich/psychiq2-dataset
[20] https://huggingface.co/derenrich/psychiq2
[21] https://github.com/derenrich/psychiq

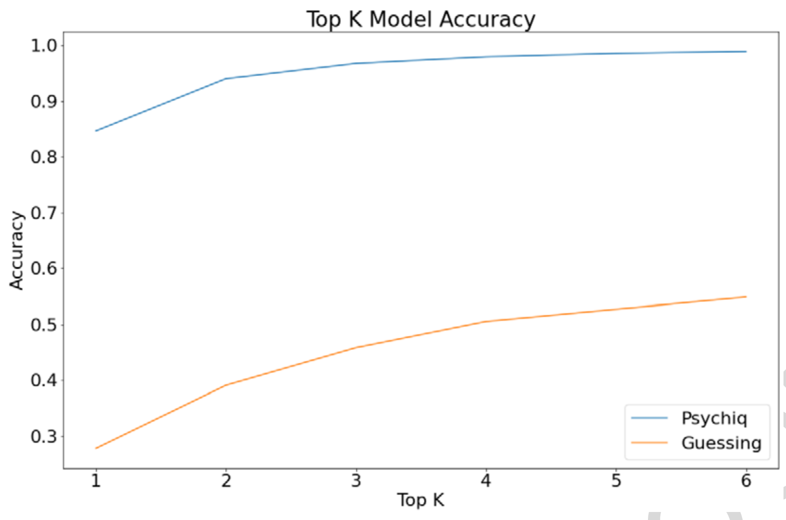


Fig. 4. Psychiq's accuracy is reliably above 80%, especially in the top-5 accuracy context where its used in Wwwyzzerdd. The chart compares Psychiq's performance against suggesting the most frequent statements (i.e. guessing).

Table 2
Most confused pairs of statements

Statement 1	Statement 2	Error frequency
instance of disambiguation page (`Q4167410`)	instance of human name disambiguation page (`Q22808320`)	1131
instance of written work (`Q47461344`)	instance of literary work (`Q7725634`)	1087
instance of gene (`Q7187`)	subclass of protein-coding gene (`Q20747295`)	1031
instance of musical work/composition (`Q105543609`)	instance of single (`Q134556`)	1087
unknown	instance of business (`Q4830453`)	976
instance of civil parish (`Q1115575`)	instance of village (`Q532`)	941

As shown in Fig. 5 Psychiq was integrated into the Wwwyzzerdd user-interface by adding a drop-down next to the Wikipedia article's title which shows the user the top few suggested statements for that article. The orb is green if one of the suggested statements is already present on the item.

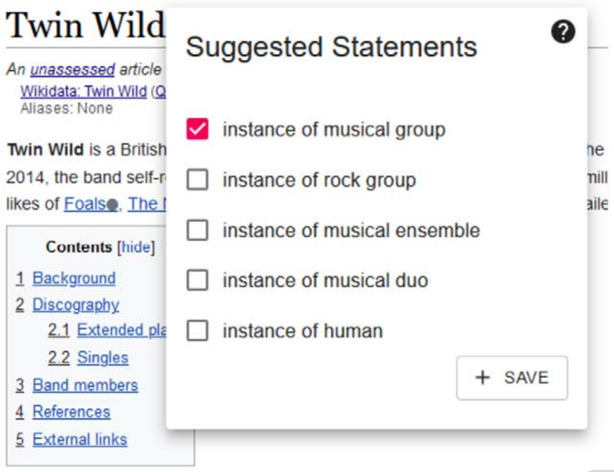
4. Discussion

4.1. *Wwwyzzerdd*

Wwwyzzerdd has seen significant adoption. 111,655 edits were made using the tool in 2022 after edit logging began in March. Figure 6 shows a trend of increasing adoption over time. 69 distinct users tried the tool. The Chrome Web Store reports 58 installs[22] and the Mozilla Add-Ons website reports 25 users.[23]

While it is difficult to judge the quality or impact of the edits being made, one measure is the fraction of edits made that end up reverted. Good or productive edits tend not to be reverted. In total 0.69% of edits made using the tool were reverted. For comparison, on a typical day, January 1 2023, about 1.3% of all edits made using the Wikidata web UI were reverted. Investigating the reverted Wwwyzzerdd edits individually suggests that a significant fraction are users reverting themselves after making an error. A possible explanation for the lower revert rate of Wwwyzzerdd users is that the userbase of Wwwyzzerdd is self selected to the most active and knowledgeable users.

[22]https://chrome.google.com/webstore/detail/wwwyzzerdd-for-wikidata/gfidggfngdnaalpihbdjnfbkfiniookc
[23]https://addons.mozilla.org/en-CA/firefox/addon/wwwyzzerdd-for-wikidata/


Suggested Statements

- [✓] instance of musical group
- [ ] instance of rock group
- [ ] instance of musical ensemble
- [ ] instance of musical duo
- [ ] instance of human

Fig. 5. Psychiq is integrated into Wwwyzzerdd as a drop-down that lets the user select statements to add to Wikidata.

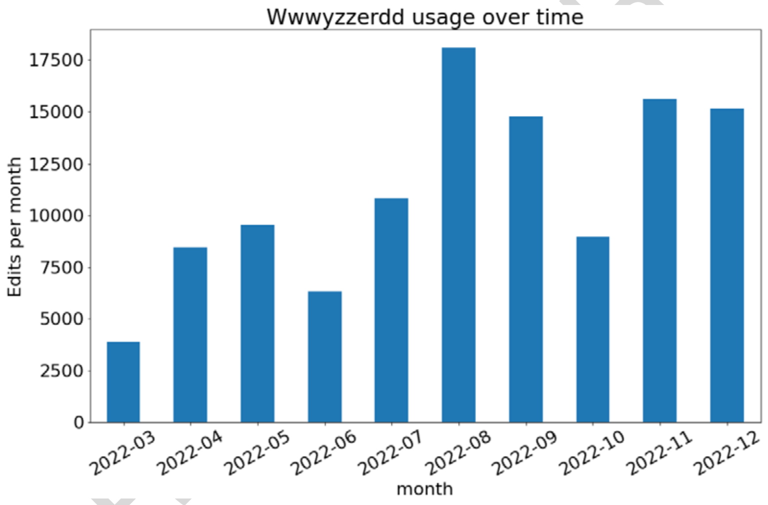


Fig. 6. Wwwyzzerdd has seen strong and growing usage since logging began in March 2022.

Evaluation of the Wwwyzzerdd edits using standard linked data quality measures [22] was not performed. The revert measure used above is an informal measure of quality and further study should be conducted to quantify the quality of the edits made using this tool.

While the tool was originally only available in English it works on all language editions of Wikipedia and has seen adoption across non-English Wikipedias. 43% of edits were made from the English Wikipedia, 36% were from the Japanese Wikipedia and, in descending order, most of the rest of the edits were from French, Belarusian, Russian, Chinese and Dutch Wikipedias.

Wwwyzzerdd has been used to add statements for a diverse set of properties. The most common property added using Wwwyzzerdd was `P19` (place of birth) at 7.8% of all edits. This is followed by `P31` (instance of) at 5.4% (this includes edits made using the Psychiq integration), `P69` (educated at) at 5.3% and `P161` (cast member) at 4.9%. 14.2% of statements added involved external identifiers and the most commonly added external identifier was `P2002` (Twitter username).

A fundamental limitation of Wwwyzzerdd and Psychiq is that they rely on the accuracy and reliability of (English) Wikipedia. The sources that Wwwyzzerdd extracts from Wikipedia are not to the underlying sources used in

Wikipedia but to Wikipedia itself. This reliance on secondary sources can be problematic especially if the data is to be reused within Wikipedia itself (creating a referential loop).

Referencing Wikipedia is convenient to the editor and better than providing no reference but is not as good as it could be. A future version of the software could allow users to import primary sources from Wikipedia directly into Wikidata. For example, a future version could allow users to drag-and-drop a reference footnote onto an orb to export that reference directly to Wikidata.

Another limitation of Wwwyzzerdd is that subsequent updates to Wikipedia are not automatically imported to Wikidata. This means that errors corrected in one place are not corrected in the other. A fully automated text to structured data system would allow updates to flow immediately from one to the other, however it is not clear such a system would be accepted by the Wikidata community.

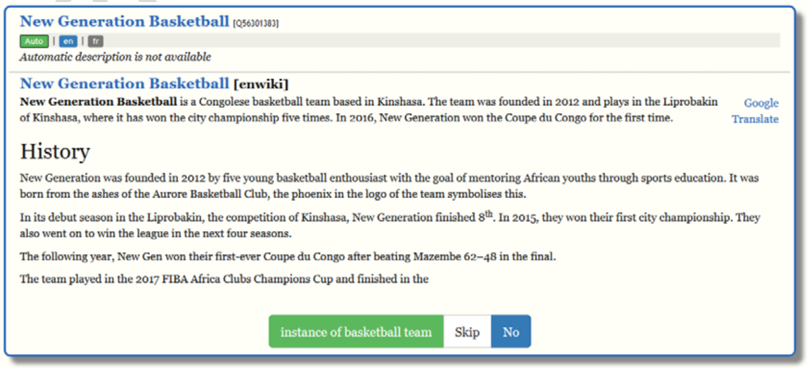
Without user studies under controlled conditions it is impossible to fully assess whether users like the experience of using Wwwyzzerdd and are made more productive by it. However, informal polling of the most frequent users suggests enthusiasm for the browser extension and a desire for it to take on more capabilities. The growth in monthly edit counts since launch also suggests that Wwwyzzerdd is successfully meeting a niche need in the Wikidata community.

In the future Wwwyzzerdd may be extended to work on non-Wikipedia websites. For that to happen a named entity extraction and linkage system would need to be integrated to allow users to select mentions of entities in general Web text (this isn't needed on Wikipedia as text is already linked to "entities" through intra-wiki hyperlinks). Users would validate the linkage back to Wikidata and draw edges between entities and select properties to connect them. The URL of the current web page could then be used as the reference for the added statement.

4.2. *Psychiq*

Comparing Psychiq to past scholarly work is difficult as most of the past work has been evaluated against DBpedia. Cat2Ax reports a 95.7% precision for the equivalent "type assertion" task. This is considerably higher than 83.2% reported in Table 1, however their performance metric is computed differently. Instead of checking for strict label equality Cat2Ax used human evaluators to judge the output as "correct or incorrect". It is very common for Psychiq to assign a label that doesn't strictly match what it is in Wikidata but that a human evaluator would judge as correct (indeed Table 2 shows that the most common errors are of this type). So the precision of Psychiq is likely underestimated under Cat2Ax's metric.

To get a better understanding of Psychiq's model performance a separate experiment was performed. In addition to being integrated into Wwwyzzerdd, Psychiq was converted into a simple "game-like" web interface.[24] Figure 7 shows what the user interface for this game looks like. The tool shows users a Wikidata item, its corresponding


New Generation Basketball [Q56301383]

Auto | en | fr

Automatic description is not available

New Generation Basketball [enwiki]

New Generation Basketball is a Congolese basketball team based in Kinshasa. The team was founded in 2012 and plays in the Liprobakin of Kinshasa, where it has won the city championship five times. In 2016, New Generation won the Coupe du Congo for the first time.

History

New Generation was founded in 2012 by five young basketball enthousiast with the goal of mentoring African youths through sports education. It was born from the ashes of the Aurore Basketball Club, the phoenix in the logo of the team symbolises this.

In its debut season in the Liprobakin, the competition of Kinshasa, New Generation finished 8th. In 2015, they won their first city championship. They also went on to win the league in the next four seasons.

The following year, New Gen won their first-ever Coupe du Congo after beating Mazembe 62–48 in the final.

The team played in the 2017 FIBA Africa Clubs Champions Cup and finished in the

Fig. 7. Using the Wikidata distributed game framework, Psychiq's top predictions are "gamified" so that users can easily validate and execute its suggestions.

[24]https://wikidata-game.toolforge.org/distributed/#game=84

English Wikipedia article and a suggested statement. The player can then accept the statement as correct, mark the statement as incorrect or skip the statement if they are unsure. The game was seeded with the 30 thousand proposed statements where Psychiq was most confident.[25] After 10,595 statements were evaluated by human volunteers, over 97.7% of suggested statements were accepted. Initial human feedback on this tool suggested the model was performing well.

The Psychiq model is robust to missing information. Newly created articles often lack categories and only have titles so a system like Cat2Ax would not be applicable. As an example, for the article "University of Vechta" without categories the model still accurately suggests that it may be a university. The model, however, will not work with only category information because the model assumes the name of the category is the name of the article. For example, just the category "Mosques in Jordan" is predicted to be an instance of "aspect in a geographic region" (`Q74817647`) when it should have predicted "mosque" (`Q32815`). Fortunately, it is not possible for an article to have categories but no title.

Another use case of Psychiq is identifying errors in Wikidata. By looking for instances where the model is very confident and disagrees with what is currently stated in Wikidata, Psychiq can find probable errors. Unfortunately doing this also produces many uninteresting conflicts where, for instance, the model identifies the item as a road but it's labeled as a highway. By filtering out uninformative pairs based on their frequency, one is left with more actionable discrepancies. Of the remaining top fifty cases where Psychiq is most confident Wikidata is in error[26] manual review found all but one to be actual errors in Wikidata. The most probable detected errors were caused by English Wikipedia pages being converted from disambiguation pages to articles about specific people without updating the corresponding Wikidata item.

Wwwyzzerdd already works in multiple languages, and it would not be difficult to also make Psychiq work for multiple languages. All that would be required is expanding the training set to multiple language Wikipedias and using a language model trained on a multi-lingual corpus. Using Wwwyzzerdd in a multi-lingual context does introduce problems if the different linked Wikipedia articles are about subtly different subjects.

A major difficulty in Wikidata is establishing and communicating the desired ontology for various classes of information. It is not obvious to new users that "Beat It" (`Q210218`) should be classed as an instance of "musical work/composition" (`Q105543609`) and not the more natural sounding "song" (`Q7366`). Another common confusion is between "bibliography" (`Q1631107`) and "bibliography" (`Q134995`) which are two different concepts with the exact same name. Today this information is communicated through Wikiprojects which write up documentation[27] or through model items[28] which are maintained examples of the desired way to model information. Instead, with Psychiq, the correct way to model an item can be suggested using machine learning. If the community wants to change how to represent something, they just needed to update all the existing items to the new ontology and then the next time the model is retrained the suggestions will automatically reflect the new ontological model.

While Psychiq's performance numbers look encouraging it is likely that selection bias makes the number appear more positive than they may otherwise be. Psychiq is only trained on Wikidata items that have already been categorized and items that are difficult to categorize are less likely to have been categorized. Thus the dataset that Psychiq is trained on is unrepresentatively easy to classify.

Psychiq's performance could likely be greatly expanded by using a more powerful and modern large-language model. Additionally the model could be given access to the entire article's text instead of just the title and categories. For now the cost and latency of such an approach makes it prohibitive.

5. Conclusion

This paper has discussed two related approaches for improving the user-experience of Wikidata completion using the textual content of Wikipedia. This work represents a step beyond the current dominant tools in the Wikidata

[25] Given the limited human volunteer time it does not make sense to have people evaluate statements of lower confidence even though doing so would provide a more comprehensive view on Psychiq's accuracy.

[26] https://www.wikidata.org/wiki/User:BrokenSegue/PsychiqConflicts

[27] https://www.wikidata.org/wiki/Wikidata:WikiProject_Music

[28] https://www.wikidata.org/wiki/Wikidata:Model_items

ecosystem. The state of the art of tooling in the Wikidata automation space still remains behind the work of the competition. DBpedia's approaches to automation fifteen years ago remain more advanced than many of the approaches being actively used in Wikidata. This is partly due to a difference in philosophy over how contributions should be made and partly a lack of investment. Hopefully this and similar parallel efforts will move Wikidata towards the state-of-the-art.

References

[1] G. Bierman, M. Abadi and M. Torgersen, Understanding typescript, in: *European Conference on Object-Oriented Programming*, Springer, 2014, pp. 257–281.

[2] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, Minneapolis, MN, USA, June 2–7, 2019, S. Papers, J. Burstein, C. Doran and T. Solorio, eds, Long and Short Papers, Vol. 1, Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/n19-1423.

[3] M. Héder and P. Mendes, *Round-Trip Semantics with Sztakipedia and DBpedia Spotlight*, 2012. doi:10.1145/2187980.2188048.

[4] N. Heist and H. Paulheim, Uncovering the semantics of Wikipedia categories, in: *The Semantic Web – ISWC 2019*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois and F. Gandon, eds, Springer International Publishing, Cham, 2019, pp. 219–236. ISBN 978-3-030-30793-6. doi:10.1007/978-3-030-30793-6_13.

[5] N. Heist and H. Paulheim, Information extraction from co-occurring similar entities, in: *Proceedings of the Web Conference 2021*, 2021, pp. 3999–4009. doi:10.1145/3442381.3449836.

[6] A. Ismayilov, D. Kontokostas, S. Auer, J. Lehmann, S. Hellmann et al., Wikidata through the eyes of DBpedia, *Semantic Web* **9**(4) (2018), 493–503. doi:10.3233/SW-170277.

[7] B. Kratzwald, G. Kunpeng, S. Feuerriegel and D. Diefenbach, IntKB: A verifiable interactive framework for knowledge base completion, in: *Proceedings of the 28th International Conference on Computational Linguistics (Online)*, International Committee on Computational Linguistics, Barcelona, Spain, 2020, pp. 5591–5603, https://aclanthology.org/2020.coling-main.490. doi:10.18653/v1/2020.coling-main.490.

[8] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer et al., DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic web* **6**(2) (2015), 167–195. doi:10.3233/SW-140134.

[9] P.N. Mendes, M. Jakob, A. Garcia-Silva and C. Bizer, *DBpedia Spotlight: Shedding Light on the Web of Documents, in: Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.

[10] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves and J. Welling, Never-ending learning, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* **12** (2011), 2825–2830.

[12] M. Peel, Pi bot, https://github.com/mpeel/wikicode.

[13] S.G. Pillai, L.-K. Soon and S.-C. Haw, Comparing DBpedia, Wikidata, and YAGO for web information retrieval, in: *Intelligent and Interactive Computing*, V. Piuri, V.E. Balas, S. Borah and S.S. Syed Ahmad, eds, Springer, Singapore, 2019, pp. 525–535. ISBN 978-981-13-6031-2. doi:10.1007/978-981-13-6031-2_40.

[14] A. Piscopo, C. Phethean and E. Simperl, What makes a good collaborative knowledge graph: Group composition and quality in Wikidata, in: *International Conference on Social Informatics*, Springer, 2017, pp. 305–322. doi:10.1007/978-3-319-67217-5_19.

[15] V. Sanh, L. Debut, J. Chaumond and T. Wolf, DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, 2019, arXiv, arXiv:1910.01108.

[16] A. Sarabadani, https://github.com/Ladsgroup/Kian.

[17] A. Sarabadani and D. Erenrich, Re: Kian, 2023.

[18] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe and P. Szekely, A study of the quality of Wikidata, *Journal of Web Semantics* **72** (2022), 100679. doi:10.1016/j.websem.2021.100679.

[19] D. Vrandečić and M. Krötzsch, Wikidata: A free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85. doi:10.1145/2629489.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest and A.M. Rush, Transformers: State-of-the-art natural language processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2020, pp. 38–45, https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[21] F. Wu and D.S. Weld, Autonomously semantifying Wikipedia, in: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, 2007, pp. 41–50. doi:10.1145/1321440.1321449.

[22] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, Quality assessment for linked data: A survey, *Semantic Web* **7**(1) (2016), 63–93. doi:10.3233/SW-150175.