Conflations and duplications in Wikidata items: causes, detection, solutions, and issues

Camillo Carlo Pellizzari di San Girolamo[1]

[1] *Scuola Normale Superiore, Pisa PI 56126, Italy*

Abstract

This paper analyzes the problems of incorrect disambiguation of entities in Wikidata items, both in general and focusing on items regarding humans. The problem of incorrect disambiguation is categorized into two types, i.e. conflations and duplications. The paper subsequently treats the causes of conflations and duplications, the methods available for detecting them, the solutions applicable to them and the issues that constitute an obstacle to the aforementioned solutions; three proposals are finally made to mitigate these issues.

Keywords

Wikidata, entity management, authority control

1. Introduction

Wikidata (WD) is a knowledge base containing, as of September 2023, more than 106 M entities[2]. The disambiguation of these entities is, or at least should be, obtained through the authority control, a key part of cataloguing [1]. The aim of authority control is creating entries that coincide exactly with the described entities; the incorrect disambiguation of an entry can result in conflations (i.e. many entities being described in the same entry) and duplications (i.e. many entries describing the same entity). The concept of authority control can be applied to the entries created by librarians (the authority records) as well as to the entries created by WD users (the WD items) [2].

This paper deals with the problem of conflations and duplications in WD items, both in general and with a specific focus on the 10+ M items regarding humans[3] (this estimate includes only individual real humans, excluding both groups of humans and fictional humans). The choice of limiting this research to humans is motivated by the standardized structure of these items, which usually contain the same core of data, i.e. name(s), birth/death dates and places, occupation(s); this makes the solution of the above problems easier than in the case of other types of items, e.g. organizations, places, or concepts.

Wikidata'23: Wikidata workshop at ISWC 2023
EMAIL: camillo.pellizzaridisangirolamo@sns.it
ORCID: 0000-0003-2699-1693

[2] Cf. https://www.wikidata.org/wiki/Special:Statistics.
[3] https://www.wikidata.org/wiki/Special:Search/haswbstatement:P31=Q5.

However, it should be noted that conflations and duplications affect significantly also other types of WD items. Discussions have aroused regarding e.g. duplications of geographical places[4] and conflations of buildings and organizations having their seat inside them[5]; these issues also affect the conceptual items composing WD ontology [3].

The following paragraphs analyze the most frequent causes of conflations and duplications in WD items, the ways of detecting these problems, the methods usually adopted in order to solve them, and the issues that presently affect this process; three proposals are finally made to mitigate these issues.

2. Causes

The causes of conflations and duplications are analyzed on three levels:

1. general causes;
2. causes specifically applicable to humans;
3. how they concretely originate in WD.

2.1. General causes

Conflations and duplications are both caused mainly by a non-bijective relationship between an entity and its name. In other words, if an entity has only one name and this name is used only by this entity, there is no risk of incorrect disambiguation.

However, incorrect disambiguation is typically a problem when an entity has many names (polyonymy) and when many entities use the same name (homonymy): a conflation happens when two entities using the same name are wrongly treated as just one entity, whilst duplication happens when two names used by the same entity are wrongly treated as two different entities.

2.2. Causes for entities regarding humans

Applying the previous description to the case of entities regarding humans, the risks of incorrect disambiguation can be described as follows.

Conflations usually affect homonyms, i.e. persons using the same name. Homonymies affect both very common combinations name-surname in a certain language (e.g. John Brown, Hans Meyer, Jean Martin, Mario Rossi etc.) and rarer ones, which could be more difficult to spot. Conflations are more probable when the homonym persons have been alive in the same period and in the same field of activity[6].

Duplications usually affect persons with many names. These names can have multiple causes: they can be multiple forms of the same name, with different degrees of completeness (i.e. including/excluding the second surname and/or the second name, or with second name represented by its initial)[7]; or they can be transliterations of the original name in different scripts[8];

---

[4] Cf. https://www.wikidata.org/wiki/Wikidata:Project_chat/Archive/2017/08#Dealing_with_our_second_planet and https://www.wikidata.org/wiki/Wikidata:Project_chat/Archive/2018/07#Another_cebwiki_flood?.

[5] Cf. https://www.wikidata.org/wiki/Wikidata:WikiProject_Performing_arts/Data_structure/Data_modelling_issues#Items_confounding_architectural_structures_and_organizations.

[6] E.g. https://www.wikidata.org/wiki/Q57906651 and https://www.wikidata.org/wiki/Q118183242, two geologists active in the same period and teaching in the same university.

[7] E.g. https://www.wikidata.org/wiki/Q111010598 (see aliases in Italian).

[8] E.g. https://www.wikidata.org/wiki/Q304890.

or they can be translations/adaptations of the original name in different languages[9]. Graphical variants (especially for premodern persons) can also generate multiple names.

2.3. Causes in WD

WD is edited both by humans and by bots. As of July 2023, most of WD edits have been made by bot accounts (55%), the others by humans (45%), either through semi-automatic tools or manually. Also considering item creations, bots (60%) prevail over humans (40%)[10]. Thus, it is relevant to consider how bot edits, semi-automated edits and manual edits can differently contribute to originating conflations and duplications, and to whom the responsibility of these incorrect disambiguations should be attributed in each case.

2.3.1. Causes in bot editing

Bot accounts (i.e. accounts possessing a bot flag) must comply with the bot policy[11]; according to it, each bot task has to be approved by the community and the bot "must stay within reasonable bounds of their approved tasks"; as of the 1st September 2023, in WD 360 accounts have the bot flag.

Duplication is an issue frequently discussed by the users commenting the requests of approval of new bot tasks, with the string *duplicat* currently occurring in more than one hundred request pages[12]: if the task involves importing new items, users commonly ask bot operators to demonstrate they are taking every possible measure to minimize the risk of creating duplicate items. In 2018 a proposal to add to the bot policy a maximum duplicate rate for bots creating new items was not approved by WD community[13]. To the contrary, conflation is almost never discussed (the string *conflat* currently occurs in only three request pages)[14].

Bots are often used also to add data to WD items using the external identifiers they contain as sources (of course, imported data should be in CC0 license[15]); e.g. if item X contains the ID Ψ, the bot can copy data from Ψ to X. However, if item X and ID Ψ are mostly about the same entity but ID Ψ also contains not-pertinent data (i.e. it is conflated), the bot importing data from Ψ to X will reproduce in X the same conflation affecting Ψ. In this case, the responsibility is mainly of the compilers of Ψ, although it can be argued that Ψ should not have been matched to X because of its being conflated; this argument is valid unless Ψ has become conflated after having been matched to X.

2.3.2. Causes in semi-automated editing

A variety of semi-automated tools can be used, without a bot account, to add statements to existing items and to create new items massively. Among the tools listed in the Bot requests

[9] E.g. https://www.wikidata.org/wiki/Q9438.
[10] https://wikidata.wikiscan.org/.
[11] https://www.wikidata.org/wiki/Wikidata:Bots.
[12] Cf. https://www.wikidata.org/w/index.php?title=Special:Search&prefix=Wikidata:Requests+for+permissions/Bot/&search=duplicat*.
[13] https://www.wikidata.org/wiki/Wikidata:Project_chat/Archive/2018/04#What_duplicate_rates_should_we_tolerate?.
[14] Cf. https://www.wikidata.org/w/index.php?title=Special:Search&prefix=Wikidata:Requests+for+permissions/Bot/&search=conflat*.
[15] Cf. https://www.wikidata.org/wiki/Wikidata:Data_Import_Guide.

page[16], the most used are QuickStatements[17] (QS) and OpenRefine[18] (OR). According to the statistics of the tool Wikidata Navel Gazer[19] (WNG), based on edit tags, as of the 1st of August 2023 a total of 390 581 099 edits have been made through QS, 16 439 295 edits through OR, 1 628 745 edits through Harvest Templates[20], 413 223 edits through WikibaseJS-cli[21]. QS is used to perform a broad range of edits in WD, whilst OR is used mainly for the reconciliation of external databases with WD. The batches of edits performed through semi-automated tools, unlike bot tasks, are *not* subject to a preliminary approval by WD community. However, batches containing a relevant percentage of mistakes can be reverted through the tool EditGroups[22]; usually reverts are decided through a community discussion[23].

Through OR the entries of an external database are reconciled with WD items, i.e. either matched with existing items regarding the same entity or created as new items. A wrong reconciliation can originate conflations (when the entry is not imported into the existing item representing its same entity, but into another item) and duplications (when the entry is not imported into the existing item representing its same entity, but as a new item instead). These wrong matches can happen in two distinct scenarios.

In the first scenario, the matches by OR are based on a third ID (i.e. if entry A and WD item X both contain the ID Ψ, the entry A is matched with item X), so the responsibility for mistakes is not of the uploader, who merely executes the upload. In such cases, the blame is either of the compilers of A (who have added to it the non-pertinent ID Ψ) or of the compilers of X (for the same reason) or of the compilers of Ψ (who have conflated in it two distinct entities).

In the second scenario, OR suggests possible WD items matching with the entries of the external database and the uploader has to review manually these suggestions. In this case, obviously, the mistakes are caused by the inaccuracy of the uploader in the review of the matches proposed by OR.

The best possible reconciliation should also pay attention to the mistakes potentially affecting the external database. In particular, the uploader should always consider the possibility that the external database contains conflated and duplicate entries. Conflated entries should not be matched with WD, whilst duplicate entries should either be all matched with the same existing item or, if no item exists for the entity, be used to create one new item and all matched to it.

2.3.3. Causes in manual editing

When editing manually, users can generate new conflations and duplications for the reasons explained in the previous paragraph; so, in this case the responsibility for the incorrect disambiguation falls entirely on WD's community.

3. Detection

In general, the detection of cases of incorrect disambiguations is based on the following reasoning: one item containing multiple values for certain statements that are expected to have

[16] Cf. https://www.wikidata.org/wiki/Wikidata:Bot_requests.
[17] https://quickstatements.toolforge.org/.
[18] https://openrefine.org/.
[19] https://bambots.brucemyers.com/NavelGazer.php?property=P-5.
[20] https://pltools.toolforge.org/harvesttemplates/index.html.
[21] https://github.com/maxlath/wikibase-cli.
[22] https://editgroups.toolforge.org/.
[23] Cf. https://www.wikidata.org/wiki/Wikidata:Edit_groups; community discussions are automatically categorized in https://www.wikidata.org/wiki/Category:Edit_group_discussions.

one single value could be a conflation, whilst two items whose data show a high degree of similarity could be duplicates.

3.1. Use of property constraints

Property constraints[24], introduced in 2015 [4], can be applied to properties in order to clarify how they should be used. Constraint violations, which can be checked both through SPARQL queries and through bot-updated reports[25], help in discovering potentially problematic statements. The constraint violations, mainly of external-identifiers properties, could also be used in order to detect conflations and duplications; these incorrect disambiguations can happen both in WD and in the external database. Given a database P containing two entries A and B, the following reasoning can be applied:

- a single-value constraint violation (SVCV) in WD (i.e. one WD item X containing both IDs A and B) could be either a conflation in WD or a duplication in P;
- a unique-value constraint violation (UVCV) in WD (i.e. two WD items X and Y both containing ID A) could be a duplication in WD or a conflation in WD or a conflation in P.

3.2. Use of SPARQL queries

Whilst the detection method through constraint violations affecting identifier-statements could be applied to all kinds of items, some other detection methods, based on SPARQL queries, are available specifically for items regarding humans.

Considering that humans can only be born (and eventually died) in one place and in one moment, if a WD item X contains e.g. two birth dates or places, it could be a conflation. However, only truly-different values should be counted (i.e. if two values are just the same value with different precisions, like 1933 and 1st March 1933, or Brooklyn and New York, they should be counted as one, and the most precise should be ranked as preferred[26]), and sometimes for the same human different sources support different values[27].

Then, considering that different humans are rarely born (and/or died) in the same place and moment, if two WD items X and Y contain the same birth date and have the same label (or similar labels) in a given language, they could be duplicates. In this case, too, exceptions exist, and they should be marked with the property "different from" (P1889) in order to avoid incorrect merges, which would conflate two different humans.

3.3. Statistics on conflations and duplications

No existing tool can be used to obtain statistics about conflations and duplications, since, as said above, there is no method allowing to discover them with full certainty. So the number of currently existing conflations and duplications is unknown (the same conclusion is reached by [5]).

However, considering the sum of SVCV and UVCV for a given external-identifier property it is possible to deduce the total number of conflations and duplications affecting either WD or the

[24] https://www.wikidata.org/wiki/Help:Property_constraints_portal.
[25] https://www.wikidata.org/wiki/Wikidata:Database_reports/Constraint_violations (these reports include also deprecated statements, which are ignored by constraints in the visualization of items and in SPARQL queries).
[26] Cf. https://www.wikidata.org/wiki/Help:Ranking.
[27] Cf. e.g. the different birth/death dates of https://www.wikidata.org/wiki/Q1698718.

considered external database. For the 5 most used external-identifier properties as of the 30th August 2023[28], the following list declares property ID, English label, number of items containing the property (defined as "items"), sum of SVCV and UVCV (defined as "disambiguation issues"):

- P698 (PubMed ID): 32 037 827 items, 58 018 disambiguation issues[29];
- P356 (DOI): 29 596 978 items, 39 934 disambiguation issues[30];
- P3083 (SIMBAD ID): 8 076 124 items, 12 013 disambiguation issues[31];
- P2671 (Google Knowledge Graph ID): 7 373 446 items, 27 236 disambiguation issues[32];
- P932 (PMCID): 6 577 473 items, 697 disambiguation issues[33];

The following list declares the same data for the 6 most used external-identifier properties whose subject type constraint allows usage in items regarding humans (P2671, being already present in the previous list, is not repeated in this one):

- P646 (Freebase ID): 4 417 648 items, 18 953 disambiguation issues[34];
- P214 (VIAF ID): 3 224 729 items, 70 566 disambiguation issues[35];
- P7859 (WorldCat Identities ID (superseded)): 1 893 721 items, 28 124 disambiguation issues[36];
- P227 (GND ID): 1 779 731 items, 14 360 disambiguation issues[37];
- P496 (ORCID ID): 1 788 906 items, 2 708 disambiguation issues[38];

As noted above, for these disambiguation issues it is impossible to disentangle automatically those depending from WD and those depending from each external database.

4. Solutions

This paragraph considers first how conflations and duplications are solved in single cases (i.e. splitting or merging items) and then how WD community generally tackles these problems, collecting relevant WikiProjects and guidelines.

4.1. Splitting items

The standard procedure for solving a conflation, after its detection, is splitting the conflated item into two items (or more, if necessary). Splitting an item consists in moving non-pertinent data from the conflated item either to an existing item (or items) or to a new item (or items). Non-pertinent data can be found in all the parts of the item:

1. labels, descriptions, and aliases;
2. statements (including identifiers);
3. sitelinks;
4. incoming links from other items (which are not, strictly speaking, a part of the item itself).

[28] Cf. https://www.wikidata.org/w/index.php?title=Template:Number_of_main_statements_by_property&oldid=1964859732.
[29] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P698&oldid=1965269162.
[30] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P356&oldid=1965284365.
[31] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P3083&oldid=1965220883.
[32] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P2671&oldid=1965225590.
[33] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P932&oldid=1965263192.
[34] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P646&oldid=1965269477.
[35] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P214&oldid=1965292956.
[36] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P7859&oldid=1965167996.
[37] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P227&oldid=1965291214.
[38] https://www.wikidata.org/w/index.php?title=Wikidata:Database_reports/Constraint_violations/P496&oldid=1965275301.

All these parts should be checked manually, in order to completely solve the conflation. Since the procedure is manual, no statistics are available about items splits.

4.2. Merging items

The standard procedure for solving a duplication, after its detection, is merging the duplicate items into one item (merges always happen between two items; in order to merge more than two items, a series of merges has to be performed[39]). Merges are performed mainly through the gadget Merge.js[40], which could be enabled by logged-in users in their Preferences; the special page MergeItems[41] could also be used. Once triggered, the merge procedure is completely automatic: one item (usually the newer item, but using Merge.js the choice is always made by the user) is redirected to the other and all its data are automatically transferred into the other. The merges made through Merge.js are tagged with the tag "gadget-merge"[42] and can be monitored through the special page RecentChanges[43].

4.2.1. Not mergeable duplicates

The following issues can hinder the merge of detected duplicate items:

- the two WD items can both contain a sitelink to a Wikimedia project having, mistakenly, two different articles for the same entity. The user can personally merge the articles (if they have a good knowledge of the language in which they are written, and enough time) or they can mark them as needing to be merged. Until the merge is performed, the two WD items cannot be merged and they are usually marked with the statement "instance of" (P31) "Wikimedia duplicate page" (Q17362920); as of the 24th July 2023, the items with the P31= Q17362920 statement are 10 538[44];
- the two WD items can both contain a sitelink to a Wikimedia project having, with a valid reason, two different article for the same entity (e.g. the two articles are written in two variants of the same language). As of now, these two items can never be merged and are marked with the statement "instance of" (P31) "Wikimedia permanent duplicate item" (Q21286738) and are interlinked through the property P2959 ("permanent duplicated item"); as of the 24th July 2023, the items with the P31=Q21286738 statement are 6 768[45] and the items using P2959 are 15 091[46].

4.2.2. Statistics on merges

Thanks to the standardized nature of merge-edits (in comparison with edits made in order to split items), some statistics are available about them through WNG: according to it, as of the 1st July 2023, 3 568 671 merges have been made by 52 435 distinct users. In June 2023, 27 561 merges have been made by 2 284 distinct users[47]; in July 2023, similarly, 27 392 merges have

[39] Cf. https://phabricator.wikimedia.org/T336192.
[40] https://www.wikidata.org/wiki/MediaWiki:Gadget-Merge.js.
[41] https://www.wikidata.org/wiki/Special:MergeItems.
[42] Cf. https://www.wikidata.org/wiki/Special:Tags.
[43] https://www.wikidata.org/w/index.php?title=Special:RecentChanges&tagfilter=gadget-merge.
[44] https://www.wikidata.org/wiki/Special:Search/haswbstatement:P31=Q17362920.
[45] https://www.wikidata.org/wiki/Special:Search/haswbstatement:P31=Q21286738.
[46] https://www.wikidata.org/w/index.php?title=Template:Number_of_main_statements_by_property&oldid=1939837531.
[47] https://web.archive.org/web/20230724151558/https://bambots.brucemyers.com/NavelGazer.php?property=P-5.

been made by 2 256 distinct users[48]. Statistics about merges can also be obtained counting the number of redirected WD items [5].

Statistics about redirected items (i.e. merged duplicates) can be obtained from the tool Wikiscan[49]: as of the 1st September 2023, WD has 109 531 235 items[50] and 4 112 295 redirected items[51]. Considering redirected items, 1 115 958 have been created by humans (specifically, 1 088 584 by registered users[52] and 27 374 by IPs[53]) and 2 996 337 by bots[54]. So, analyzing the total of merged duplicates, it seems that bots have created nearly the triple of merged duplicates in comparison with humans. But, taking into consideration the data available by year, bots have created more merged duplicates than humans each year from 2012 to 2019 (596 021 merged duplicates created by humans, 2 741 326 by bots), whilst the contrary is true each year from 2020 to 2023 (519 937 merged duplicates created by humans, 255 011 by bots). So, from 2020 humans (editing both with semi-automated tools and manually) are responsible for the creation of about two thirds of merged duplicates.

4.3. Approaches in WD community

The WD community has created specific WikiProjects[55] in order to deal with the problem of incorrect disambiguations: WikiProject Duplicates[56] was founded in 2016, whilst WikiProject Conflation[57] was founded in 2023. WikiProjects are meant to coordinate users involved in tackling a certain issue, or curating a certain group of items. Problematic disambiguations are frequently discussed also in thematic WikiProjects, as stated in [6].

Relevant guidelines about incorrect disambiguations include the practical ones, about splitting items[58] and merging items[59], and the theoretical ones, about conflations[60] and duplications[61]. Best practices have also been collected in a subpage of WikiProject Duplicates[62].

5. Issues and possible approaches

This paragraph proposes some measures which could have positive effects in mitigating the issues of incorrect disambiguation in Wikidata, with a specific focus on their prevention and on the efficiency of the procedure for solving them.

5.1. Prevention

[48] https://web.archive.org/web/20230831104225/https://bambots.brucemyers.com/NavelGazer.php?property=P-5.
[49] https://wikidata.wikiscan.org/.
[50] https://wikidata.wikiscan.org/?menu=tables&submenu=creation&filter=creation_noredir.
[51] https://wikidata.wikiscan.org/?menu=tables&submenu=creation&filter=creation_redir.
[52] https://wikidata.wikiscan.org/?menu=tables&submenu=creation&filter=creation_redir&type=user.
[53] https://wikidata.wikiscan.org/?menu=tables&submenu=creation&filter=creation_redir&type=ip.
[54] https://wikidata.wikiscan.org/?menu=tables&submenu=creation&filter=creation_redir&type=bot.
[55] Cf. https://www.wikidata.org/wiki/Wikidata:WikiProjects.
[56] https://www.wikidata.org/wiki/Wikidata:WikiProject_Duplicates.
[57] https://www.wikidata.org/wiki/Wikidata:WikiProject_Conflation.
[58] https://www.wikidata.org/wiki/Help:Split_an_item.
[59] https://www.wikidata.org/wiki/Help:Merge.
[60] https://www.wikidata.org/wiki/Help:Conflation (see also, for humans, https://www.wikidata.org/wiki/Help:Conflation_of_two_people).
[61] https://www.wikidata.org/wiki/Help:Deduplication.
[62] https://www.wikidata.org/wiki/Wikidata:WikiProject_Duplicates/VIAF_members.

With respect to the prevention of these issues, it has been noted that bot tasks need to be approved before running and the users discussing them usually require the issue of duplication to be tackled thoroughly. To the contrary, semi-automated batches run by users using OR, QS, or other tools are not subject to any approval procedure, but can only be criticized and eventually undone afterwards. The presence, in batch editing, of a relevant percentage of incorrect disambiguations can be considered disruptive editing, and thus can fall under the blocking policy[63] and can be a valid reason for undoing the whole batch itself, but there is no policy specifically dealing with conflations and duplications.

Approving a policy containing precise standards of quality for semi-automated batches, including norms regarding incorrect disambiguations, could have positive effects both in encouraging the users running the batches to care more about data quality and in providing a clear reference point for judging if the mistakes affecting a batch are serious enough to justify undoing it. This policy could allow to users having run problematic batches to choose between undoing them or fixing their mistakes in a reasonably brief span of time.

5.2. Data round-tripping

With respect to the detection of the issues, constraint violations and SPARQL queries are already an effective mean to find a relevant amount of disambiguation issues. However, the lists of items obtained through these means mix disambiguation issues affecting WD items and disambiguation issues affecting the external databases to which WD items link. Whilst the first ones, once solved, disappear from the lists, the second ones cannot be solved in WD, but only in the external database itself, and so can remain in the lists for long time, wasting the time of the users stumbling upon them. This is a major issue in the workflow of users interested in dealing with disambiguation issues in WD.

Data round-tripping[64], i.e. the synchronization of WD data with the external database's data (implying also the correction of mistakes on each side), surely benefits the quality of both WD items and the external database's entries [7]. Thus, each external database should be interested in receiving mistake reports from, among others, WD users. Some databases effectively provide a contact method (web form, e-mail, phone number etc.; each WD property can indicate through P10923 the error-report method used by the database) and, when contacted, solve the reported mistakes on a regular basis, but others never answer reports (or explicitly refuse them[65]), and a few ones do not even provide any contact method. National authority files, whose IDs are widely used in Wikidata, are also affected by these issues, as shown in [8], and often lack effective ways of mistake report.

Until the mistake is solved in the external database, WD items have to keep conflated and duplicate external IDs. Conflated IDs are usually ranked as deprecated and qualified with P2241 ("reason for deprecated rank") Q14946528 ("conflation") – this qualifier has 10 341 occurrences, as of the 25th July 2023[66] –, whilst duplicate IDs do not usually receive any specific marker, since in most cases all these IDs are equally valid. The presence of these problematic IDs in WD items, as said, has the main negative effect of flooding the lists of constraint violations with a relevant amount of false positives (i.e. incorrect disambiguations which cannot be solved in WD, but need to be solved elsewhere).

[63] https://www.wikidata.org/wiki/Wikidata:Blocking_policy.
[64] Cf. https://www.wikidata.org/wiki/Wikidata:Data_round-tripping.
[65] Cf. https://web.archive.org/web/20230621075514/https://www.wikidata.org/wiki/User:CaféBuzz/BNF.
[66] https://w.wiki/77bn.

The creation of a tool simplifying both the reporting of mistakes in external databases for WD users and the management of such reports by the curators of these databases could mitigate this issue[67].

5.3. Management of conflations

The disambiguation issues, after being detected, follow different paths of solutions: splitting a conflated item is fully manual operation, requiring to evaluate to which item each datum needs to be assigned, whilst merging two duplicate items is an automated operation requiring just a few seconds.

Although, obviously, disentangling two mixed entities necessarily requires human judgement in order to decide the exact boundaries of each entity, it is nonetheless evident that the present procedure for splitting items has some drawbacks: it is very time-expensive and there is a high risk, for the user, to forget checking some parts of the conflation, thus not solving it completely. These issues can also discourage less experienced users from trying to solve conflations when they find them. Moreover, item splits are presently impossible to monitor, since they are not performed through a gadget (which could assign them a specific tag).

A related issue regards the solution of conflations deriving from incorrect merges[68]: the merge, after having been proven wrong, has to be undone separately in both the involved items, and then all incoming links have to be checked (and corrected, whenever necessary) manually.

Introducing a gadget designed to help users in solving conflations could solve the above problems. The gadget should present the user a panoramic view of the two items he is managing, distinguishing their four parts (labels, descriptions, and aliases; statements; sitelinks; incoming links), and provide the user a simple interface for moving these parts from one item to the other. It could integrate two already existing gadgets, Move[69] (used for moving sitelinks) and moveClaim[70] (used for moving statements). This new gadget would facilitate the whole process and in this way it would both encourage users to solve conflations[71] and make some statistics about item splits available.

6. Related work

The issue of entity identification and disambiguation has been discussed in semantic web literature. The surveyed literature is mainly concerned with the development of automatic (or semi-automatic) methods and tools for the disambiguation of entities. The problem of preventing database contributors from adding new conflations and duplications, which is typical of user-generated databases such as Wikidata, appears not to be the subject of dedicated publications.

Among the surveyed publications, the following are of particular interest. [9] provides a survey of techniques and tools used in entity management systems for the semantic web. [10] deals with the issue of author disambiguation in bibliographic databases. [11] describes the clusterization process used by VIAF and how it deals with ambiguities. [12] proposes a method for identifying duplicate entries for people and companies in a given dataset. [13] describes the deduplication procedure used in the database ScholarlyData[72].

[67] Cf. https://phabricator.wikimedia.org/T312718.
[68] See https://phabricator.wikimedia.org/T237262.
[69] https://www.wikidata.org/wiki/MediaWiki:Gadget-Move.js.
[70] https://www.wikidata.org/wiki/MediaWiki:Gadget-moveClaim.js.
[71] Cf. https://www.wikidata.org/wiki/Wikidata_talk:Ontology_issues_prioritization.
[72] http://www.scholarlydata.org/.

7. Conclusions

Disambiguation issues are among the problems affecting the data quality of WD items: two entities having the same name can be conflated into one item, one entity with two names can be duplicated into two items. These issues are generated in WD items by bots, by humans using semi-automated tools and by humans editing manually. Statistics about item merges (i.e. including only duplicates which have already been detected and merged) suggest that until 2019 most duplicates have been created by bots, whilst from 2020 most duplicates have been created by humans.

Conflations and duplications are detected through constraint violations and SPARQL queries: if an item contains two values for the same datum, it could be a conflation; if two items contain the same value for the same datum, it could be a duplication (or one of the two items is conflated). The considered datum is typically an external identifier; for this reason, the detected disambiguation issue can lie either in WD items or in the considered external database. The issues affecting WD items can be solved directly by WD users, whilst the issues affecting external databases need to be solved by these databases.

Solving a conflation implies splitting the conflated item, an operation which is performed manually, checking each piece of the item. To the contrary, the solution of a duplication, i.e. merging the duplicate items, is a fully automated operation, which can also be monitored through statistics.

As outlined, three main proposals are advanced in order to mitigate the problem of incorrect disambiguation, both improving its prevention and facilitating its solution. Firstly, in order to reduce the number of mistakes introduced by semi-automated batches, it is proposed to introduce precise standards of quality for these batches, including norms regarding incorrect disambiguations. Secondly, a more efficient data round-tripping procedure is needed in order to make the detection and solution of incorrect disambiguations more efficient: a dedicated tool could simplify the communication between WD users and external databases' curators, encouraging the second ones to receive and answer efficiently the mistake reports coming from the first ones. Thirdly, a new gadget helping users in solving conflations could make items splits faster and help users in performing them without forgetting some parts of the conflation itself, besides making possible to obtain some statistics, not available as of now.

8. References

[1] R. A. Wiederhold, G. F. Reeve, Authority Control Today: Principles, Practices, and Trends, Cataloging & Classification Quarterly 59.2-3 (2021) 129–158. doi:10.1080/01639374.2021.1881009.

[2] C. Bianchini, L. Sardo, Wikidata: a new perspective towards universal bibliographic control, JLIS.it 13.1 (2022) 293–311. doi:10.4403/jlis.it-12725.

[3] D. Ammalainen, Wikidata Ontology Issues. Suggestions for prioritisation based on the perceived frequency of occurrence and the severity of impact on data re-use, 2023. URL: https://commons.wikimedia.org/wiki/File:Wikidata_ontology_issues_%E2%80%94_suggestions_for_prioritisation_2023.pdf.

[4] D. Vrandečić, L. Pintscher, M. Krötzsch, Wikidata: The Making Of, in: WWW '23 Companion: Companion Proceedings of the ACM Web Conference 2023, Association for Computing Machinery, New York, NY, 2023, pp. 615–624. doi:10.1145/3543873.3585579.

[5] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe, P. Szekely, A study of the quality of Wikidata, Journal of Web Semantics, 72 (2022). doi:10.1016/j.websem.2021.100679.
[6] T. Kanke, Knowledge curation work in Wikidata WikiProject discussions, Library High Tech 39.1 (2021) 64–79. doi:10.1108/LHT-04-2019-0087.
[7] S. Fauconnier, Data Roundtripping: a new frontier for GLAM-Wiki collaborations, 2019. URL: https://diff.wikimedia.org/2019/12/13/data-roundtripping-a-new-frontier-for-glam-wiki-collaborations/.
[8] C. Bianchini, S. Bargioni, C. C. Pellizzari di San Girolamo, Beyond VIAF: Wikidata as a Complementary Tool for Authority Control in Libraries, Information Technology and Libraries 40.2 (2021). doi:10.6017/ital.v40i2.12959.
[9] A. Morris, Y. Velegrakis, P. Bouquet, Entity Identification on the Semantic Web, in: Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP2008), Rome, Italy, December 15-17, 2008.
[10] C. Hedeler, B. Parsia, B. Mathiak, Using the semantic web for author disambiguation-are we there yet?, in: ISWC 2014 Posters & Demonstrations Track, pp. 449–452.
[11] T. B. Hickey, J. A. Toves, Managing Ambiguity In VIAF, D-Lib Magazine 20.7/8 (2014). doi:10.1045/july2014-hickey.
[12] M. Holub, O. Proksa, M. Bieliková, Detecting Identical Entities in the Semantic Web Data, in: SOFSEM 2015: Theory and Practice of Computer Science. 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24-29, 2015, Proceedings.
[13] Z. Zhang, A. G. Nuzzolese, A. L. Gentile, Entity deduplication on ScholarlyData, in: The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14, pp. 85–100.