Reverts Revisited – Accurate Revert Detection in Wikipedia

Fabian Flöck
Institute AIFB
Karlsruhe Institute of Technology
Karlsruhe, Germany
fabian.floeck@kit.edu

Denny Vrandečić
Institute AIFB
Karlsruhe Institute of Technology
Karlsruhe, Germany
denny.vrandecic@kit.edu

Elena Simperl
Institute AIFB
Karlsruhe Institute of Technology
Karlsruhe, Germany
elena.simperl@kit.edu

ABSTRACT

Wikipedia is commonly used as a proving ground for research in collaborative systems. This is likely due to its popularity and scale, but also to the fact that large amounts of data about its formation and evolution are freely available to inform and validate theories and models of online collaboration. As part of the development of such approaches, revert detection is often performed as an important pre-processing step in tasks as diverse as the extraction of implicit networks of editors, the analysis of edit or editor features and the removal of noise when analyzing the emergence of the content of an article. The current state of the art in revert detection is based on a rather naïve approach, which identifies revision duplicates based on MD5 hash values. This is an efficient, but not very precise technique that forms the basis for the majority of research based on revert relations in Wikipedia. In this paper we prove that this method has a number of important drawbacks - it only detects a limited number of reverts, while simultaneously misclassifying too many edits as reverts, and not distinguishing between complete and partial reverts. This is very likely to hamper the accurate interpretation of the findings of revert-related research. We introduce an improved algorithm for the detection of reverts based on word tokens added or deleted to adresses these drawbacks. We report on the results of a user study and other tests demonstrating the considerable gains in accuracy and coverage by our method, and argue for a positive trade-off, in certain research scenarios, between these improvements and our algorithm's increased runtime.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative computing, Computer-supported cooperative work, Web-based interaction*; H.1 [**Models and Principles**]: User/Machine Systems—*Human factors*

Keywords

Wikipedia, revert detection, editing behavior, user modeling, collaboration systems, community-driven content creation, social dynamics

HT'12, June 25–28, 2012, Milwaukee, Wisconsin, USA.

1. INTRODUCTION

Wikipedia is a prime example of what can be achieved by mass collaboration on the Web. Its open, participatory policies supported by user-friendly collaboration technology have encouraged millions of Internet users to turn into Wikipedia editors, and dedicate themselves to the creation and continuous improvement of what has become the world's largest (online) encyclopedia. The same principles guide Wikipedia's handling of information, enabling and inspiring a great number of research studies about its content, community and processes, which lead to new insights in the fields of collaborative systems and user behavior. In addition to its primary content, Wikipedia grants free access to a huge amount of historical metadata concerning a large share of its editing activities.[1] This wealth of resources, including detailed usage and edit logs, are often used by scholars to inform and validate their theories and models of online collaboration.[2]

One of the basic tools when analyzing Wikipedia editor behavior is referred to as 'revert detection', which is typically defined as the task of finding edits that undo one or more previous edits (a full discussion about the notion of a revert is given in Section 2). In practice, and taking into account the very nature of the editing process in Wikipedia, revert detection forms a foundational step for many (more elaborated) research ideas, and its purposeful handling leads to a superior understanding of wiki-like systems of collaboration in general. It is used as input and filter for a variety of analysis tasks, including the extraction of implicit networks of editors, the analysis of edit or editor features that influence the chance of a given edit to be reverted, and the removal of noise when analyzing the emergence of the content of an article.

Despite its importance as a pre-processing step, most research work capitalizing on reverts-related information relies on a rather naïve revert-detection approach. In a nutshell, the approach discovers identical revisions of an article based on the MD5 hash values of the corresponding changes, and considers all edits lying between two consecutive revision duplicates as being reverted. Although its coarseness is discussed in some work (see Section 3), the extend of its oversimplification of revert behavior has not yet been acknowledged in its full range of implications. We argue that this approach is neither sound nor complete; it detects only a (in some cases very limited, see Section 5) share of the actual reverts - as they are defined and understood in Wikipedia editing practice, cf. Section 2 - and falsely classifies many co-occurring editing activities as reverts, thus leading to lower-quality analysis data which is

[1] A small number of edits is not made available due to legal reasons such as, for instance, copyright laws.

[2] The datasets are available at `http://download.wikimedia.org`.

very likely to hamper the accurate interpretation of the findings of Wikipedia-related research in areas as those mentioned earlier.

In this paper, we introduce an algorithm improving the detection of reverts in terms of accuracy and coverage. This is achieved by comparing edits based on the actions they perform, which are measured by means of the word tokens they add or delete.[3] To evaluate its performance against the naïve approach currently in use, we conducted a user study and other tests, which clearly provide evidence of significant gains in accuracy and coverage, in particular when considering full (as opposed to partial, see Section 2) reverts giving substance to in-depth studies of editing behavior at the level of individual articles.

The remainder of this paper is organized as follows. Section 2 first introduces the terminology used in the paper, elaborating on the definition of reverts and edits, and motivating the need for revisiting revert detection in Wikipedia. Section 3 gives an overview of the current state of the art in the field and discusses the deficiencies of existing approaches in more detail. Section 4 explains our revert-detection algorithm. Section 5 presents the results of the evaluation, including the positive trade-off between the benefits in accuracy and coverage achieved by the improved method and its potentially increased computational costs compared to the baseline approach. The paper is concluded by a summary of the findings and an outline of planned future work in Sections 6 and 7, respectively.

2. REVERTS AS BASIS FOR USER BEHAVIOR MODELING

In the official Wikipedia guidelines the definition of a 'revert' reads as follows: *"**Reverting** means undoing the effects of one or more edits, which normally results in the page being restored to a version that existed sometime previously. More broadly, reverting may also refer to any action that reverses the actions of other editors, in whole or in part."*[4] Text passages from other Wikipedia pages concur with this definition, see *"A revert means undoing the actions of another editor."* on the Wikipedia:Edit_warring page.[5] The notion of a 'revert' is further discussed and established in the following section. Furthermore, we would like to define a set of basic terms used in the remainder of this paper as follows:

- An 'edit' is the act of changing the content of an article followed by saving these changes to a new version of the article.
- A 'revision' is a version of the article created by an edit. That is, edit 1 creates revision 1, edit 2 creates revision 2, and so on.
- A 'reverting edit' is an edit that carries out a revert, i.e., reverts one or more other edits.
- A 'reverted edit' is an edit whose changes to an article are undone partly or completely by a reverting edit ex post. The article version that is changed by the reverting edit is the 'reverted revision'.

Within Wikipedia, reverting can be carried out in different ways:[6]

1. Manual reverting, which means deleting text from or adding text to an article by hand.
2. Activating the 'undo' button next to an edit in the article history dialog. It enables to undo the actions performed by only that specific edit.
3. Using the 'rollback' feature, which immediately reverts all top consequent edits made by the last editor, going back to a previous version of the corresponding article. It is available to administrators and editors who have been explicitly granted the right to use this function.

Article editing is one of the most frequent user activities in Wikipedia. As of September 2011, 10.9 million edits are made on a monthly basis, with 3.5 million alone in the English Wikipedia.[7] The edits logged in the Wikipedia article-history dumps provide a rich footprint of previous and ongoing user activities, which can be leveraged in a variety of behavioral studies. The research reported here is based on data from the English Wikipedia. However, the method we present should be applicable to most other language Wikipedia's, provided a means to split Wikitext into word tokens, which are the unit on which our method operates.

Editing in Wikipedia refers to the actual encyclopedic articles, to the pages discussing other aspects of the overall collaborative project, as well as to the talk pages associated with the articles. In our work we focus on the first of the three Wikipedia-namespaces, due to its predominance in Wikipedia-related research.[8] For each edit the system logs a variety of metadata, including the name and ID of the article, the date- and timestamp, the username or IP of the editor, the content of the revision in wiki syntax, and a short comment. However, there is much more, potentially relevant insight to be gained by looking at indirectly gathered information; in particular, to analyze the social dynamics among editors, it is essential to study their (direct or indirect) inter-actions in terms of, for instance, specific user activities and their effects on the activities of other users. There are a few user actions that potentially inform such editor relations and can be derived from the editing logs available. One of the most prominent examples is probably 'co-editing', which refers to editing by multiple editors that content-wise steers an article in a similar direction: for example, two editors making edits that introduce similar statements into an article in different subsections. Nevertheless, measuring this type of activities assumes knowledge of the actual meaning of the edits undertaken, in order to identify those which are in agreement with the overall direction of the article. By contrast, reverts may reliably yield evidence of the relation of the actions (edits) of editors indirectly, even without having to take into account the semantics of the content. Let us illustrate how inferences about editor interrelations could be drawn based on rather shallow knowledge of the content of an edit through a simple example: If edit 1 establishes an article consisting only of the

[3] The algorithm is available for further usage by interested researchers at http://people.aifb.kit.edu/ffl/reverts/

[4] http://en.wikipedia.org/wiki/Wikipedia:Reverting, also compare http://en.wikipedia.org/wiki/Help:Reverting (accessed 13.09.11, bold from original, italics added).

[5] http://en.wikipedia.org/wiki/Wikipedia:Edit_warring (accessed 13.09.11, italics added).

[6] http://en.wikipedia.org/wiki/Help:Reverting, (accessed 13.09.11)

[7] http://stats.wikimedia.org/reportcard/RC_2011_09_detailed.html and http://stats.wikimedia.org/EN/TablesDatabaseEdits.htm (both accessed 03.11.2011).

[8] Reverts in the discussion namespace are rare as the goal is not to produce a single collaborative piece of content. For other namespaces of Wikipedia besides articles, the question of how to model reverts, in particular the applicability of an approach analogue to the one which is subject of this paper, remains to be investigated in future work.

word apple', edit 2 adds 'pie' after apple' and edit 3 deletes only the word 'pie', we can conclude intuitively that edit 3 deleted the content introduced by edit 2. If these were edits by two different editors, we can further assume that the editor of edit 3 wanted to delete the content introduced by the editor of edit 2 without considering the meaning conveyed by the words that are added and deleted in the process.

As briefly explained through the example above, reverts are relatively easy to extract and interpret compared to other types of editing activities, while providing an essential insight into the behavior of editors. A number of works on Wikipedia have been using reverts as a metric in their studies, be it on general trends involving reverts [16], correlation of the chances of getting reverted with specific editor or edit characteristics [7], using reverts as an indicator of damage repair and vandalism fighting [10, 13], or considering them in some other ways when analyzing editing behavior [8]. By looking into the user interaction modeled through the detected reverts and combining the reverts with additional data generated from the article history and other available sources, it is possible to extract the behavioral interrelations of editors in various ways, for instance by building a social network graph of editors inside an article, as it has been done by Kittur et al.[11]. In the case of social network modeling, it is especially important to not only detect who was reverted or who was reverting, but also *who was reverted by whom*, i.e., to model the antagonistic dynamics in an article on a detailed and accurate level to unveil the fine-grained conflicts and interrelations between users.

Data quality is always a relevant issue when it comes to interpreting reverts, but it is essential in making sense of the social dynamics a the level of individual articles, which can sometimes mean interpreting conflicts among only a handful of users who influence the direction of the entire article. We specifically aim to devise a revert detection method that is suited to deliver data to understand and model these intra-article socio-technical mechanisms in Wikipedia, as discussed in earlier work of ours [6]. Achieving this aim poses, however, higher demands on the quality of the method as is currently delivered by available techniques, as a large number of false-positives and -negatives in the results might lead to grave misinterpretations of editor relations. We give examples for this in Section 3. It is for these reasons necessary to be able to rely on the accuracy and completeness of the identification of edits as 'reverts'. This, in turn, requires a precise and purposeful notion of what a 'revert' is, and a revision of existing methods operationalizing revert detection to accommodate this theoretical understanding.

3. STATE-OF-THE-ART IN REVERT DETECTION

Wikipedia-related research using reverts as a metric [5, 7, 8, 11, 13, 16, 12] almost invariably deems only so called 'identity reverts' as an appropriate means to investigate revert behavior.[9] This approach relies on finding two revisions containing exactly the same content via MD5 hashes [15].[10] Subsequently, all edits lying between two identical revisions are considered as *reverted*, with the second identical revision as the *reverting* edit, and the first one as the one *reverted to*. As defined by Halfaker et al. in [7]: "A revert is a special kind of edit that restores the content of an article to a previous revision by removing the effects of intervening edits.". Table 1 shows an example of how reverts are detected in this manner.

Beside such work using identic revisions to detect reverts, there is research discussing types of reverts and actions to be considered reverts more in-depth, as well as the implications and complexity of reverts in Wikipedia (e.g. [14]). But those publications do not introduce an algorithm evaluated to detect reverts in a more accurate way. The work by Adler, Chatterjee and de Alfaro [1, 2] implements very elaborated approaches for keeping track of addition and removal of words in an article by different editors. But it is neither aimed at, nor evaluated in respect to precisely detecting revert relations between editors.

Where used in the research literature mentioned above, the basic identity revert detection method (SIRD) is never explicitly ex ante motivated by a theoretical concept of revert behavior or by any definition established by the Wikipedia editor community. Rather, the motivation for using SIRD as stated (representatively) by Halfaker et al. is that it "is computationally simple and determining exactly which editors' revisions were lost due to the revert is straightforward." [7]. The underlying (implicit) notion of what a revert is can be seen as a simplification of how Wikipedia defines the concept ("[...]which normally results in the page being restored to a version that existed sometimes previously.", see Section 2). It does not require the reverting edit to actually *undo the actions* of an edit identified as reverted (compare the indicated revert of edit 4 by edit 5 in Table 1). As such the revert detection method is also not able to make distinctions concerning the relationship between reverting and reverted edit: It is not possible to indicate if the reverting edit fully, partly or not at all undid the actions of the reverted edit (again, compare the example in Table 1). It also does not require the intention of the reverting edit to revert any other edit.

SIRD supposedly detects most of the existing reverts: Kittur et al. showed that by combining a method based on edit comments, (looking for the keywords 'revert' and 'rv') and SIRD, 95% of the *reverting edits* found as a result could be found using only identity reverts [11].[11] In a number of subsequent papers [7, 13, 16], this finding was used to conclude that the mere 5% additional reverting edits found by looking at comments do not justify the effort of using this additional source of information on top of the SIRD method. Still, to our knowledge, there was no dedicated investigation so far if other detection methods might find even more reverts,[12] as many users do not attach comments to their edits [11, 13, 16] and MD5 hashes cannot be used to find partial reverts that do not produce identical revisions [7].

In terms of a 'real-world-check' of the conceptualization underlying the SIRD method, there has not been any testing of the false-positive-rate of the delivered results, in the sense of evaluating it

[9] There is some other work that identifies reverts solely based on regular expressions (e.g., using keywords such as 'rv','revert') in edit comments, calling it a 'reasonable proxy' for revert detection [10] (also [9]). But this is to be neglected here as the work cited as source for this statement clearly states that "MD5 (identity) reverts actually capture more revisions than user-labeled (comment) reverts (3.7M vs. 2.4M), suggesting that a substantial number of reverts are not labeled as such" [11].

[10] The MD5 hash sum is commonly used to check if two file or text contents are identical.

[11] In the paper, the authors refer to such edit pairs as the found 'reverts' while actually they report the number of found reverting edits that either have an identical previous version or a comment mentioning the two keywords [11]. Where identified by comments, it cannot be in all cases concluded *what* revisions were actually reverted when there is no identical version (i.e., in the case of a partial revert) and no indicator in the comment (e.g., in the case of a comment consisting only of one of the keywords).

[12] Ekstrand et al. [5] compared cosine similarity and adoption coefficient approaches with the SIRD for finding revision history "trees". They come to the conclusion that the SIRD algorithm is a better solution for representing revision relationships than the other two approaches.

Table 1: Example of the result of the simple identity revert detection method

Edit number	Revision content (text)	Words deleted/added (actions taken) in the edit	MD5 hash (simplified)	Detected reverts
1	Zero	+'Zero'	Hash1	(Reverted-to by edit 5)
2	Zero Apple Banana	+'Apple' +'Banana'	Hash2	Reverted by edit 5
3	Zero Apple Banana Coconut Date	+'Coconut' +'Date'	Hash3	Reverted by edit 5
4	Zero Coconut Date	-'Apple' -'Banana'	Hash4	Reverted by edit 5
5	Zero	-'Coconut' -'Date'	Hash1	Reverting edits 2, 3, 4

against the Wikipedia definition of a revert or what is perceived as a revert by Wikipedia users. This means there was no evaluation, for instance, if the actions of identified *reverted* edits are really undone in subsequent *reverting* edits. This is a crucial issue especially in the light of the very simplistic, technology-driven definition of a revert the SIRD method implicitly builds upon. Although we know of at least one analysis toolkit (pyMWDat)[13] that extends to some degree the above described basic definition of reverting and reverted edits, we are so far not aware of an elaborate and working algorithm modeling revert behavior, which is designed to capture reverts based on a more realistic definition of how reverting and reverted edits are related.[14]

The inability of the SIRD method to detect reverts that do not create a duplicate revision is acknowledged by Priedhorsky et al., who state that beside the identity revert exists the "*effective revert*, where the effects of prior edits are removed (perhaps only partially)." [13]. Such cases cannot be fully detected using only MD5 hashes [7]. In Table 1, this is exemplified by the actions of revision 4, which deletes all words introduced by revision 2, while still generating a completely new revision content. Intuitively, one could say that revision 4 is *effectively* reverting revision 2, as it undoes all its actions; this interpretation conforms with the Wikipedia revert definition. The SIRD method, however, will not detect the revert relationship in this way, but instead, as shown in the example, detect revision 5 as reverting revision 2, because it (incidentally) lies between two identical revisions. In a scenario where revision 5 would be non-identical to revision 1, the method would not even detect any revert of revision 2. This illustrates the logical inconsistencies of the conceptual model on which SIRD implicitly operates. Note that this is only one of a number of many example scenarios in an edit history we found, where the SIRD method leads to a questionable result. Additional examples are given in Section 4.

The coarseness of SIRD is further discussed by Priedhorsky et al. [13], who note that understanding and taking into account the intention of a revert is challenging (and thus not feasible), while the method already covers one of the most common types of reverts (producing identical revisions) at an, arguably, sufficient level of quality. This statement is, however, neither proved, nor is it necessary an acceptable argument considering the current features of Wikipedia: The manual revert and the undo feature as accessible to common registered Wikipedians do not automatically generate an identical revision, but can, and presumably often do, generate partial reverts. This leads to the likely assumption that they make up the vast majority of non-bot reverts. The rollback feature, which undoes all edits after a specified revision, on the other hand, is available only to a few editors with the corresponding access rights, including administrators.

As a conclusion, when setting the Wikipedia revert definition as a benchmark for the understanding of revert behavior in current Wikipedia editing practice, the coverage (finding all actual reverts) and accuracy (finding only true-positives) of the SIRD method are suboptimal. Edits are always and only detected as reverted if they lie between two identical revisions for reasons which are not further taken into account. This has a number of important consequences:

- Edit pairs are detected as reverts that cannot be seen as reverts when compared to known edit behavior in Wikipedia and the general understanding of editing practice of the contributors;
- there might be many reverts still to be found by untested methods; and
- for those reverts that are found by the current approach, it cannot be distinguished to which extent a revert took place (full revert, 20%, 70% partial revert, and so on).

Partial reverts are more complicated to use in modeling social interactions than full reverts; for the latter, it is at least known that *all* actions of an editor have been undone, while for the former, the range of possible interpretations of the revert action is comparatively much wider.[15] When using the results of a revert analysis in scientific work, it should therefore at least be possible to in- or exclude partial reverts or select down to which degree of undoing the detected edits should be treated as reverts. This applies in particular to those scenarios where a thorough analysis based on a comparatively smaller data corpus is of interest, such as the editing behavior in one specific Wikipedia article. These scenarios motivated the development of the method presented in the next section.

4. AN IMPROVED REVERT DETECTION METHOD

4.1 Revert definition

The first step towards devising a more accurate revert detection method was to establish a clear conceptual foundation of what a revert is, followed by an algorithm that detects all and only those edits that fit the corresponding definition.

The Wikipedia revert definition was used as a reference point, as it states what actions constitute a revert as a behavior of an editor, and as it is grounded in the common editing practice of the Wikipedia community. For assessing the results of our method versus SIRD, we gave priority to detecting edits that are

[13] Available at `http://code.google.com/p/pymwdat/` (accessed on 06.11.11)

[14] As noted in the documentation of the tool, pyMWDat works as the described by SIRD method, but differentiates between the revisions marked as 'reverted', in other words, those made by the first editor after the 'reverted-to' revision is marked as 'possible vandalism', while the remaining reverted edits are classified as a separate group of 'good-will edits'.

[15] To give an example, the removal of a word from a 600-word-entry could mean only a small correction, while adding the word 'not' in a certain position could change the meaning of the whole entry.

1. No false-positives, i.e., only reverts fitting the used definition; and

2. Full reverts, as only for those edits we can safely assume they were fully undone and thus unambiguously indicate an reverted-reverting relationship to be leveraged in social-graph-modeling and interpreting the results.[16]

According to these pre-requisites and taking into account what kind of data can be reliably used to identify reverts, the following definition was set up:

"An edit A is reverted if all of the actions of that edit are completely undone in one subsequent edit B. Edit B has then reverted edit A."

Note that this definition does not rule out that edit B performs other actions on top of undoing A's actions or the actions by a number of different edits (for an example see Table 2, where edit 8 is reverting edits 6 and 7). In addition, if A's actions have been undone only partially, this is not counted as a partial revert according to the used definition. Accordingly, if all of A's actions have been undone in a collective effort by many partly reverts, A is not counted as reverted (although all its actions were undone) An example is given in Table 2 (edits 9 to 11 are not reverting edit 8). This is due to the fact that, in this case, it would not be possible to assign a single reverting edit B, and thus not unambiguously determine the reverting and the reverted edit in every case. Those two restrictions do not comply with the Wikipedia concept of a revert, as it comprises partial reverts as well. Still, they were introduced to account for the aforementioned prerequisites. In doing so, our approach trades some of its coverage (of detecting all reverts as stated in the Wikipedia definition) for ease of computational and interpretational processing. Still, the approach is superior - assuming the Wikipedia definition - to the state of the art method in accuracy, coverage and the meaningfulness of the found reverts, as we will discuss in Section 5.

4.2 Algorithm

To operationalize the actions of the editors we use added and deleted word tokens, i.e., character chains separated by white spaces. We operate on the wiki syntax, not on the front-end article content.

Before taking a look at the specific word changes of an edit, we eliminate unchanged paragraphs from the preceding and current revision in order to reduce the amount of text that has to be collated via rather costly text difference comparisons (DIFFs). We first split up the syntax text into paragraphs via double line breaks. Each paragraph is assigned an MD5 hash value. When comparing the preceding and current revision, all paragraphs with matches in their hash values are discarded for the next step, as they were not altered by the edit leading to the current revision.[17] As most edits change no more than one or two paragraphs, this pre-filtering step proved to yield significant runtime improvement during testing of our algorithm.

To compare the remaining (edited) paragraphs, text difference comparisons are used.[18] The location of the word tokens within the article text is not taken into account, i.e., the revision content is treated as a bag-of-words. For every edit B_n in the article history, we check via text DIFFs if its previous i edits A_{n-1} to A_{n-i}, performed the exact opposite of a subset of actions of B_n, starting with A_{n-1}. The maximum scan-range i is in our example set to 20 previous revisions (we discuss the size of i in Section 5). If a negative matching subset is found in A_{n-1} we do not look for the same subset in the following A_{n-2} to A_{n-20}, as the action of A_{n-1} can only be undone once. In the next step, a content list for B is generated, containing the revision identifiers of all revisions still part of the article text. If no negated subset was found in the previous 20 edits, solely the ID of the revision produced by edit B is added to the content list. If one or more of the previous 20 edits contain a negated subset of B, then the revision ID of those edits A_{n-i} is removed from the content list for B. If B, on top of reverting other edits, added or deleted additional content that does not have a matching negative subset in the previous 20 edits, its revision ID is also added to the content list. Table 2 illustrates the procedure.

When performing the computation discussed above, we filter out so called 'blankers' from the list of possible reverting edits. Blankers are edits deleting the whole content of an article, which should not be treated as common reverting edits, as their behavior cannot be interpreted as an intentional act aimed at undoing the specific edits whose added content they delete. Rather, the vandalistic intention is aimed at harming the article as a whole.

Comparing revisions 1 to 5 in Table 2 with the example in Table 1 shows that, with our method and adhering to our definition, revision 5 is only reverting revision 3, while revision 4 is reverting revision 2. This indicates higher accuracy (according to Wikipedia's definition) in comparison to SIRD. With our method, we additionally detect the revert by revision 8 of revisions 6 and 7, where no duplicate revisions can be found. This means our method is potentially able to find more reverts than SIRD.

There are many other examples of reverts where our method can extract much more meaningful revert information compared to the baseline approach.[19] One very frequently occurring scenario is the repair of a vandalistic revision by several subsequent revisions, all trying to recreate the revision before the vandalism occurred. If the last one in this row of repair edits recreates the last vandalism-free revision, all other repair edits will be marked as reverted with the SIRD method, although each of the edits did only a partial revert. With the DIFF method and the definition introduced earlier in this section, we do not incorrectly assign reverts in this setting.[20]

Note as well that, if using a different definition (e.g., the full Wikipedia definition), our algorithm can be adapted to find partial reverts in addition to full reverts, covering scenarios as the revert of edit 8 in Table 2. As the DIFFs between revisions are already calculated in the current implementation of the method, it is only

[16] As noted earlier, the modeling of a social network at the level of individual articles was the main motivation for devising a new revert detection method.

[17] According to our definition, we do not count simple rearrangements of text as reverts.

[18] To find reverts adhering to our definition using DIFFs does suffice. We do not rule out adding other methods in a next step to increase the performance of our method further. We discuss this shortly in Section 7.

[19] We identified a large number of theoretical and practical cases by hand, where reverts do not produce identic revisions or where the simple SIRD approach is imprecise. They are not all listed here because of space constraints.

[20] An example: let the actions by edit 8 in Table 2 be considered vandalism. We then assume that edits 9, 10 and 11 are trying to repair damage caused by edit 8. While edits 9 and 10 overlooked some vandalistic effects, edit 11 eventually restores the last vandalism-free revision 7. In this case, the SIRD algorithm would have assigned edit 11 as the reverting edit of edits 8, 9 and 10, although, according to Wikipedia's definition, at least the actions of edits 9 and 10 were not undone. That means that we, on the one hand, do not detect the revert of edit 8 (false negative), but on the other hand, avoid detecting false-positives, i.e., the revert of 9 and 10 by edit 11. As vandalism reverting is a very common activity in Wikipedia, we assume the trade-off is positive.

Table 2: Example of the result of the improved revert detection method

Edit number	Revision content (text)	Words deleted/added (actions taken) in the edit	Content list (revision ##)	Detected reverts
1	Zero	+‘Zero’	1	
2	Zero Apple Banana	+‘Apple’ +‘Banana’	1;2	Reverted by 4
3	Zero Apple Banana Coconut Date	+‘Coconut’ +‘Date’	1;2;3	Reverted by 5
4	Zero Coconut Date	-‘Apple’ -‘Banana’	1;3	Reverting 2
5	Zero	-‘Coconut’ -‘Date’	1	Reverting 3
6	Zero Fig	+‘Fig’	1;6	Reverted by 8
7	Zero Fig Grape	+‘Grape’	1;6;7	Reverted by 8
8	Zero Huckleberry	-‘Fig’ -‘Grape’ +‘Huckleberry’	1;8	Reverting 6, 7
9	Zero Huckleberry Grape	+‘Grape’	1;6;8	
10	Zero Huckleberry Fig Grape	+‘Fig’	1;6;7;8	
11	Zero Fig Grape	-‘Huckleberry’	1;6;7	

necessary to build the content list on a word token basis instead of a revision ID basis.

4.3 Implementation

We implemented the SIRD approach and our new algorithm in Python.[21] To decrease runtime, we parallelized the algorithms using the Wikimedia Utilities by Aaron Halfaker, which provide a high-performance framework for analyzing large amounts of Wikipedia data.[22]

The input for both algorithms is a list of all revisions for a given article, consisting of the revision ID and the text of the revision. Both scripts also implement the detection of blankers, as noted earlier. For generating the text DIFFs, the “difflib” library of Python is used, as it correctly produces the needed revision differences.

Whereas the hashing algorithm as part of SIRD runs very efficiently (new content hashes are compared to previous ones), the new algorithm produces results much slower (it is quadratic over the number of words in the DIFFs). As in this work, we focused on showing the qualitative differences of the detected revert relations between the two methods, we optimized the runtime of our algorithm only to a certain extent. There are a number of pre-filtering techniques we are about to implement in future work, as pointed out in Section 7, which will potentially decrease the execution time further. Also, the algorithm needs to be executed only once for each revision and the resulting DIFFs can be saved and made accessible as metadata to the Wikipedia article revisions, so that repeated calculations (for example with a different revert definition) can be performed at much shorter runtimes, as the DIFF execution is the most computationally expensive part.

5. EVALUATION

We evaluated both the accuracy and the coverage of the reverts found by the DIFF method against the baseline SIRD approach.

5.1 Accuracy evaluation

For comparing the accuracy of the revert detection we set up a revert assessment survey with Wikipedia editors.[23] Only Wikipedia editors were chosen as we are interested in results that conform with what the Wikipedia community perceives as a revert. The survey was conducted for 11 days in October 2011. Participants were recruited through several internal Wikipedia outlets such as the Community Portal and the Village Pump.[24] We set up two samples of 20 assessment steps to be evaluated by the participants.[25]

The first sample (referred to as sample A) of 20 assessment steps consisted of 9 edit pairs detected as a revert only by the DIFF method, 9 detected only by the SIRD method and, as a control group, 2 pairs detected by both methods in the same way. 29 users completed this first sample and all assessment steps.[26] The second sample (sample B) consisted of 8 edit pairs detected as a revert solely by the DIFF method, 8 detected only by the SIRD method and 4 pairs detected by both methods. 16 participants, distinct from the assessors of the first sample, completed all steps in this sample.[27]

The samples were designed to include more SIRD- and DIFF-only edit pairs than results identified by both approaches, because the aim of the evaluation was to compare them against each other. The samples of 9, 9, 2 (A) and 8, 8, 4 (B) edit pairs, respectively, were randomly drawn from the pool of all unique revert-pairs detected by the two methods in five randomly selected Wikipedia articles. To generate the edit pairs, the number of i previous revisions to be scanned for reverted edits (as explained in Section 4.2) was set to 20.[28]

[21] Both scripts are available under an open-source license from `http://people.aifb.kit.edu/ffl/reverts/`

[22] `https://bitbucket.org/halfak/wikimedia-utilities/wiki/Home`

[23] All participants were asked if and how long they had been an editor. All were editors who performed reverts on a regular basis themselves. We asked for the length of their tenure as editor and their experience with reverts, which both had no significant impact on the answers. We also have no reason to believe the self-selection of participants introduced a bias in the answers.

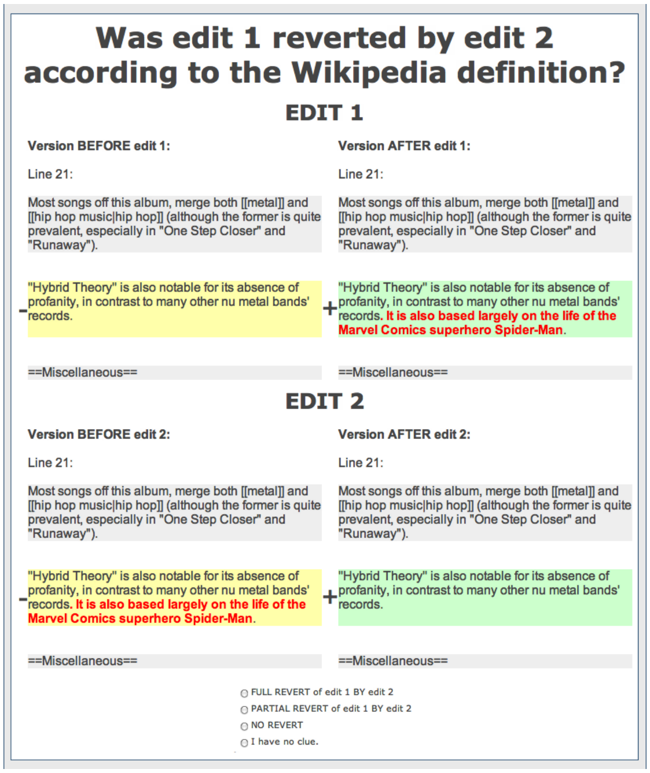
[24] Community Portal: `http://en.wikipedia.org/wiki/Wikipedia:Community_Portal`
Village Pump: `http://en.wikipedia.org/wiki/Wikipedia:Village_pump` (both accessed 25.10.11).

[25] We preferred to restrict the number of assessment steps to 20, as first tests showed that participants did otherwise abort the survey prematurely due to its perceived over-length.

[26] Nine users were excluded in total from the two samples, as they aborted the survey after only one or two questions. This was done to prevent a potential bias in more and possibly different answers for earlier questions.

[27] Unfortunately, no more than 16 participants volunteered to complete the assessment steps for sample B, which was set up after 29 users completed sample A. The number of edit pairs detected by both methods was raised slightly sample B to get assessments for a bigger sub-sample of these edits.

[28] The rationale behind this design decision was that an intentional revert targeting specific content is likely to happen within a limited window of edits after the original edit took place, as changes stay in focus of the community (and the change logs) for a limited amount



Was edit 1 reverted by edit 2 according to the Wikipedia definition?

EDIT 1

Version BEFORE edit 1:	Version AFTER edit 1:
Line 21:	Line 21:
Most songs off this album, merge both [[metal]] and [[hip hop music|hip hop]] (although the former is quite prevalent, especially in "One Step Closer" and "Runaway").	Most songs off this album, merge both [[metal]] and [[hip hop music|hip hop]] (although the former is quite prevalent, especially in "One Step Closer" and "Runaway").
- "Hybrid Theory" is also notable for its absence of profanity, in contrast to many other nu metal bands' records.	+ "Hybrid Theory" is also notable for its absence of profanity, in contrast to many other nu metal bands' records. **It is also based largely on the life of the Marvel Comics superhero Spider-Man.**
==Miscellaneous==	==Miscellaneous==

EDIT 2

Version BEFORE edit 2:	Version AFTER edit 2:
Line 21:	Line 21:
Most songs off this album, merge both [[metal]] and [[hip hop music|hip hop]] (although the former is quite prevalent, especially in "One Step Closer" and "Runaway").	Most songs off this album, merge both [[metal]] and [[hip hop music|hip hop]] (although the former is quite prevalent, especially in "One Step Closer" and "Runaway").
- "Hybrid Theory" is also notable for its absence of profanity, in contrast to many other nu metal bands' records. **It is also based largely on the life of the Marvel Comics superhero Spider-Man.**	+ "Hybrid Theory" is also notable for its absence of profanity, in contrast to many other nu metal bands' records.
==Miscellaneous==	==Miscellaneous==

○ FULL REVERT of edit 1 BY edit 2
○ PARTIAL REVERT of edit 1 BY edit 2
○ NO REVERT
○ I have no clue.

Figure 1: Screenshot of an assessment step in the survey

An assessment step consisted of two text DIFFs, exactly as known from the Wikipedia revision history feature.[29] Figure 1 depicts a sample assessment step. The first DIFF, shown on the top of the page, represented edit 1 (the DIFF showed what was changed by edit 1). The second DIFF depicted edit 2, which was a subsequent edit from the same article identified as the reverting edit of edit 1 by one of the revert detection methods. The use of colors and the "+/-" signs were adapted without changes from Wikipedia, as explained on the respective "Help:Diff" page.[29]

Participants were asked if, according to the Wikipedia definition of a revert, edit 2 had reverted edit 1. The answer options included 'Full revert', 'Partial revert', 'No revert' and 'I don't have a clue.'. The participants were provided with the Wikipedia revert definition in each assessment step (omitted in Figure 1) and were particularly asked to apply this definition in their assessment, and not their own definition, if different. At the end of the survey we asked the respondents if the Wikipedia definition conformed with their own definition of a revert, on a Likert-scale from 1 (no agreement) to 5 (full agreement). 17 editors answered this question, with 15 voicing full agreement and two agreeing only partly (scores 2 and 3).

For each of the two samples, containing 20 edit pairs each, the overall agreement of the participants that the corresponding pair was either a full revert, a partial revert or no revert, was computed. The assessed edit pairs received 29 (16) votes, distributed over the three revert types (or "I don't have a clue.").[30] Consequently, the revert types could achieve a score from 0 up to all 29 (16) participants agreeing in each assessment step. Figure 2 and Table 3 show the average agreement of participants over all edit pairs for each of the two methods and each of the three types of revert.[31] When asked if the displayed edit-pair was a full revert, in the mean 22.4 (12.6) of the participants expressed their agreement for the pairs found only by the DIFF method, while only 7.4 (3.8) did so for the edit pairs detected only by the SIRD method. This difference in the agreement score means was significant at $p < 0.01$ ($p < 0.01$).[32] When asked if an edit pair was a partial revert, the agreement was at a mean of 6.7 (3.9) for the SIRD method and at 1.2 (1.1) for the DIFF method. With $p < 0.1$, this difference was, however, not significant for sample A, but significant at $p < 0.01$ for sample B. When asked if an edit-pair was no revert at all, i.e., a false positive, a mean of 14.3 (8) participants agreed for the edit pairs detected only by the SIRD, while only a mean of 5 (2.3) said so for the pairs detected solely by our DIFF method. This difference was significant, at $p < 0.05$ ($p < 0.05$). The means of agreement for the control group of the edit pairs found by both methods were generally aligned with those of the new-method-only edit pairs in both samples, as they revealed no significant differences (therefore excluded from the summary of results in Table 3). They were not aligned with the means of the SIRD-only sample, as can be observed in Table 3. For sample A, only the difference in the agreement scores

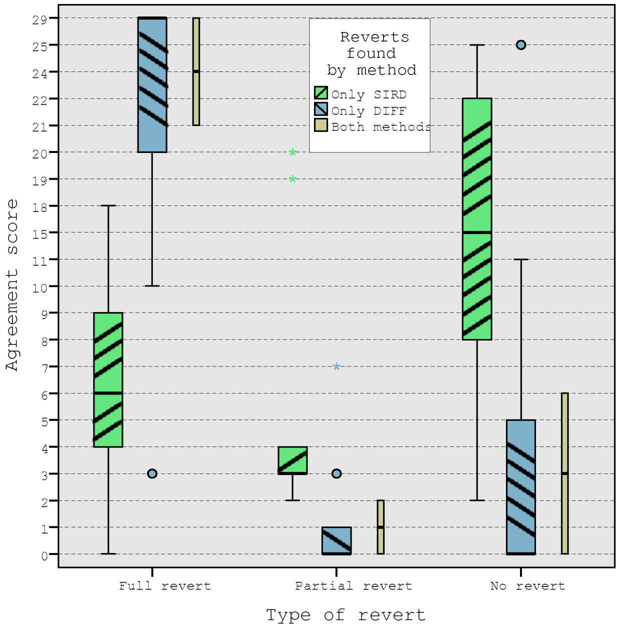


Figure 2: Boxplot comparison of the means of agreement score between all three methods, grouped by indicated type of revert for sample A (25th and 75th percentiles as box, 1.5x interquartile range (IQR) as whiskers, outliers > 1.5*IQR, extremes > 3*IQR)

of edits. This was confirmed through additional evidence collected by manual assessment we conducted. From this assessment, we had good reason to assume maximum accuracy at $i = 20$.

[29] http://en.wikipedia.org/wiki/Help:Diff (accessed 01.11.11).

[30] Numbers of sample B are put in brackets, henceforth.

[31] The raw data of the survey answers can be found at http://people.aifb.kit.edu/ffl/reverts/ The box plot for sample B was omitted as it showed very similar agreement scores to sample A, thus yielding no additional information other than confirming our previous findings.

[32] The mean agreement for the pairs found by both methods was 25 (14.8). It was significantly different from the SIRD-method-only sample at $p < 0.01$, but not significantly different from the new-method-only sample.

for full reverts differed significantly ($p < 0.01$), but for sample B, all differences were significant. The means of agreement for the answer “no clue” are not listed here as they were very low (< 0.6) for both methods and samples and all revert types, and there were no significant differences.

Table 3: Means of agreement scores for different methods and revert types, for both survey samples

Sample A	n	Full revert	Partial revert	No revert
DIFF only	9	22.4	1.2	5
Difference		p < 0.01		p < 0.05
SIRD only	9	7.4	6.7	14.3
Difference		p < 0.01		
Both methods	2	25	1	3

Sample B	n	Full revert	Partial revert	No revert
DIFF only	8	12.6	1.1	2.3
Difference		p < 0.01	p < 0.05	p < 0.05
SIRD only	8	3.8	3.9	8
Difference		p < 0.01	p < 0.05	p < 0.01
Both methods	4	14.8	0.8	0.3

Differences tested via Student’s T-test. P-values only shown where significant with $p < 0.05$ and always depicting the differences between means below and above. n stands for the number of assessment steps/ edit pairs.

5.2 Coverage of revert detection

To evaluate the performance of our method against SIRD with respect to the number of reverts detected, we analyzed a sample of 5000 randomly selected articles, which were no redirect pages. They contained a total of 392724 edits, which equals 77.116 billion possible combinations of edit pairs. The results were generated via two scripts implementing the two methods under evaluation, as described in Section 4.3. First, we report on the results produced with a maximum scan-range of $i = 20$ previous revisions for identic hashes and, respectively, corresponding DIFFs, as used for the accuracy evaluation in Section 5.1.

In total, 75278 unique edit pairs were detected as reverts in the articles by the two methods. Of those reverts, 39816 were found by both methods in the exact same manner, 14495 were found only by the SIRD method, and 20976 were found only by the new method. Table 4 gives an overview of the results. Thus, for 27% of the reverts found by the SIRD, the new method found different reverts. On top, the new method found about 12% more reverts than the SIRD method. The runtime on a 2.80 GHz Intel Xeon, 4 core, Debian Squeeze 64bit system with 8GB RAM, running with four threads and measured with the Unix ‘time’ command was 4 min 16 sec for SIRD and 51 min 52 sec for the new method, respectively.

When carried out with a maximum scan-range of $i = 50$ and $i = 100$, the number of reverts found by both methods as well as the difference between the two methods grew considerably. The new method found 19% and 24% more reverts, respectively, as can be seen in Table 4. The table also shows the increase in runtime for a larger size of the edit window.

5.3 Discussion of the evaluation results

Regarding accuracy, participants of the revert assessment agreed that in the mean, for the new method, the found edit pairs are (1) less often false positives, i.e., no actual reverts and (2) more often full reverts than those pairs found only by SIRD.

When we consider the basic definition of a revert SIRD operates with, and the scenarios in which this oversimplification can lead to suboptimal revert classification, as demonstrated in Section 4, the results make a strong case for the more accurate revert detection offered by our method. The explanatory power of the results might be to some extent impaired by the relatively small sample of assessment steps and survey participants. Nevertheless, the high significance of the key findings, and the fact that two distinct groups of Wikipedians assessed two distinct sets of edit pairs in an almost identical fashion speaks for the generalizability of the observations. The survey results for the new DIFF method appear to be in accordance with the part of the detected reverts which both methods are able to find. It becomes clear from the answers of the editors that the share of reverts that is solely identified by the SIRD method is in the mean significantly more often wrong and finds considerably fewer full reverts compared to the total number of reverts detected.

Looking at this number, the new method is able to detect from 12% (at $i = 20$) up to 24% (at $i = 100$) more reverts than SIRD. When operating on larger editing windows the accuracy might decrease because the new method will more likely match negative subsets of word tokens that were not meant to be reverted by the potential reverting edit. An example would be the deletion of the common word ‘a’ in such a case. If ‘a’ was introduced 6 edits ago, it is more likely that this is the same ‘a’ that was deleted. If the addition of ‘a’ is found, say, 99 edits ago, the likelihood of being a revert target of the current edit is comparatively lower, as this token could, for instance, be from a completely other part of the text, could have been missed in a previous step of the DIFF calculation, and so on. This means that decreasing the window size in this example would increase the likelihood to find an actual revert.

Nevertheless, at least for $i = 20$, we can postulate the following based on the analysis of the evaluation: Given the result of a revert analysis with the SIRD method on a set of edits, our method is able to detect different revert-pairs that are significantly less likely to be false-positives and more likely to be full reverts for 27% of the revert-pairs detected by the baseline approach. In addition, the new method finds 12% more revert-pairs of the same, higher accuracy than does the SIRD method. Note that we excluded partial reverts from the detection for this evaluation. The resulting number of detected reverts would increase considerably when including these.

Another aspect that speaks in favor of the new method is the following: SIRD gives preference to detecting reverts that produce an identical revision. This is more likely to happen when the rollback function (see Section 2) is used, as rollbacks invariably return an article to a pre-existing revision. It is less likely to happen for undo-based and manual reverting. As the rollback function is available only to administrators and editors with special rights, the majority of editors has to make use of the remaining two procedures in order to revert. It is therefore plausible that reverts detected via identical revisions were conducted by a disproportionately high number of users with special rights and administrators. In turn, the reverts left undiscovered by SIRD are more prone to being carried out by ‘common’ editors. In this manner, SIRD introduces a bias towards a special user group. So, even given a theoretical high accuracy in detecting reverts that can be identified with identical hashes, this bias would exists when relying solely on this method.

Looking at these results, it must be concluded that Wikipedia research work that bases its inferences on data derived via the SIRD method runs the danger of being missguided. Not fully acknowledging these impairments or dismissing them as ignorable noise in the data is more than tenuous; at least if the aim is to model the complex dynamics in specific articles on a detailed and accurate level.

Table 4: Number of detected reverts (edit pairs) in article sample by methods, with runtimes, for different levels of *i*

i		Pairs detected by SIRD	 DIFF	Gain by DIFF	Total unique detected pairs	Detected by both methods	Det. only by SIRD	Det. only by DIFF	Runtime SIRD	Runtime DIFF
20	Absolute	54311	60783	6472	75278	39816	14495	20967	4m 16s	61m 52s
	%	-	-	12	100	52.2	19.3	27.9		
50	Absolute	55647	66115	10468	81714	40048	15599	26067	4m 27s	74m 22s
	%	-	-	19	100	49.0	19.1	31.9		
100	Absolute	56101	69549	13448	85604	40076	16025	29503	4m 45s	79m 01s
	%	-	-	24	100	46.8	18.7	34.5		

6. CONCLUSION

In this paper, we provided new substantial evidence that simple revert detection via MD5 hash values is not sufficient to accurately capture all relevant revert relationships between Wikipedia editors and that these shortcomings seem to be more grave than generally suspected in the research work that applies this method[33]. We presented a new method for the detection of reverts in Wikipedia which compares edits based on the actions they undertake at the level of word tokens added or deleted. Our method relies on a revert notion which is closer to the official Wikipedia guidelines, and to the general understanding of the Wikipedia community with respect to reverting behavior.

As revealed by a user study, our method, without implementing a very complex algorithm, is more accurate in identifying full reverts as understood by Wikipedia editors. More importantly, our method detects significantly fewer false positives than the SIRD method; this is due to the simplified revert model the SIRD method operates on, which does not perform optimally in practice when extracting revert data for realistically modeling editor behavior in Wikipedia. A limiting factor for these encouraging results is the fact that the assessed samples of edit pairs and editors were by no means large. However, the answers of two distinct groups of Wikipedians on two distinct samples of edit pairs showed almost identical assessments. Given these oberservations, combined with the key findings being highly significant and an algorithm built on a solid theoretical foundation, rooted in the Wikipedia community's revert definition, we are confident that our results can be further generalized. Concerning the number of identified reverts, we found that an average 27% of the revert pairs detected by SIRD are not accurate in the above regard and that the DIFF method we developed can not only detect the same amount of reverts with better accuracy, but on top finds 12% more revert pairs than the baseline approach.

While outperforming the baseline approach in terms of quality of results, the major drawback of our algorithm is the increased computational cost. As it is quadratic over the number of words in the DIFFs, in its current implementation it might not be the tool of choice if larger amounts of articles are to be analyzed; especially in the case of complete history dumps of the large Wikipedias, e.g., English, German or Spanish.[34] For this purpose, SIRD has the clear advantage of a shorter runtime. When used to compare revert trends over time for a complete Wikipedia history dump or large batches of articles to each other, the lack of accuracy may for some research scenarios prove negligible in favor of computational efficiency.[35] Still, when calculating relations between editors at the article level and dealing with other tasks that require a in-depth look on reverts, using SIRD introduces the risk of misinterpreting and wrongly modeling revert actions. In particular, the editorial social system of an article, to be studied via social network analysis or visualization, requires an accurate and complete capturing and depiction of what is happening among key editors and editor camps. We are confident that research relying on SIRD for that purpose due to the lack of alternatives [11] or proposing similar modeling [6] will profit from the accuracy and coverage gain of our method.

We believe that our findings can be transferred to any other wiki system that shows editor characteristics and functionalities similar to Wikipedia, given the Wikitext can be split into word tokens. This, of course, has to be subject of further research.

7. FUTURE WORK

In order to improve the efficiency of our algorithm, we are experimenting with additional pre-filtering techniques such as W-Shingling[4] and Bloom filters[3], which we yet have to evaluate regarding the quality of the results.

We are planning to continue our experiments investigating the effects of an increased maximum scan-range of revisions i (see Section 4) on accuracy and coverage. Preliminary insights revealed that choosing an edit frame size of 20 is likely to yield optimal results.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257790, project RENDER.[36] Special thanks go to Andriy Rodchenko and Maribel Acosta of the KIT Karlsruhe for their efforts regarding the implementation work.

8. REFERENCES

[1] T. Adler and L. Alfaro. A content-driven reputation system for the Wikipedia. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 261–270, 2007.

[2] T. Adler, K. Chatterjee, L. Alfaro, M. Faella, I. Pye, and V. Raman. Assigning trust to Wikipedia content. In *International Symposium on Wikis*, 2008.

[33] It of course depends on the specific research work if and what actual effect these shortcomings have on the eventual outcome of that research.

[34] Of course, the calculated DIFFs (which require the majority of the computational effort) can be reused once generated. But this is of little practical implication as they are not available for many articles yet.

[35] Given the generation of identical revisions through reverts stays almost the same over time, which is not necessary the case when new revert-features are introduced or abolished, as it is the case with the rollback feature. In inter-article comparison, one has to assume an equal production rate of identity reverts per revert in all articles, in turn.

[36] `http://www.render-project.eu`

[3] F. Bonomi, M. Mitzenmacher, R. Panigrahy, S. Singh, and G. Varghese. An improved construction for counting bloom filters. In *Algorithms – ESA 2006*, volume 4168 of *Lecture Notes in Computer Science*, pages 684–695, 2006.

[4] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8–13):1157–1166, 9 1997.

[5] M. D. Ekstrand and J. T. Riedl. rv you're dumb: Identifying discarded work in wiki article history. In *The Fifth International Symposium on Wiki's and Open Collaboration*, Orlando, FL, Oct. 2009.

[6] F. Flöck, D. Vrandečić, and E. Simperl. Towards a diversity-minded Wikipedia. In E. Munson, D. W. Hall, D. de Roure, and S. Staab, editors, *Proceedings of the ACM 3rd International Conference on Web Science 2011*, 06 2011.

[7] A. Halfaker, A. Kittur, R. Kraut, and J. Riedl. A jury of your peers: quality, experience and ownership in Wikipedia. In D. Riehle and A. Bruckman, editors, *Int. Sym. Wikis*. ACM, 2009.

[8] A. Kittur, E. H. Chi, B. A. Pendleton, B. Suh, and T. Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. In *Proceedings of the 25th Annual ACM Conference on Human Factors in Computing Systems (CHI 2007)*, 2007.

[9] A. Kittur and R. E. Kraut. Beyond wikipedia: coordination and conflict in online production groups. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 215–224, New York, NY, USA, 2010. ACM.

[10] A. Kittur, B. A. Pendleton, and R. E. Kraut. Herding the cats: the influence of groups in coordinating peer production. In D. Riehle and A. Bruckman, editors, *Int. Symposium on Wikis '09*. ACM, 2009.

[11] A. Kittur, B. Suh, B. A. Pendleton, and E. H. Chi. He says, she says: conflict and coordination in Wikipedia. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 453–462, New York, NY, USA, 2007. ACM.

[12] S. K. Lam, A. Uduwage, Z. Dong, S. Sen, D. R. Musicant, L. Terveen, and J. Riedl. Wp:clubhouse? an exploration of wikipedia's gender imbalance. In *WikiSym 2011*, Mountain View, CA, 10/2011 2011. ACM, ACM.

[13] R. Priedhorsky, J. Chen, S. T. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in Wikipedia. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268, New York, NY, USA, 2007. ACM.

[14] R. Priedhorsky and L. Terveen. Wiki grows up: arbitrary data models, access control, and beyond. In *7th International Symposium on Wikis and Open Collaboration*, pages 63–71, Mountain View, CA, 10/2011 2011. Association for Computing Machinery (ACM).

[15] R. Rivest. The MD5 Message-Digest Algorithm, 1992.

[16] B. Suh, G. Convertino, E. H. Chi, and P. Pirolli. The singularity is not near: slowing growth of Wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, WikiSym '09, pages 8:1–8:10, New York, NY, USA, 2009. ACM.