Fine-Grained Controversy Detection in Wikipedia

Siarhei Bykau [#], Flip Korn [+], Divesh Srivastava [*], Yannis Velegrakis [†]

[#] *Purdue University, USA* [+] *Google Research, USA* [*] *AT&T Labs-Research, USA* [†] *University of Trento, Italy*

sbykau@purdue.edu, flip@google.com, divesh@research.att.com, velgias@disi.unitn.eu

***Abstract*—The advent of Web 2.0 gave birth to a new kind of application where content is generated through the collaborative contribution of many different users. This form of content generation is believed to generate data of higher quality since the "wisdom of the crowds" makes its way into the data. However, a number of specific data quality issues appear within such collaboratively generated data. Apart from normal updates, there are cases of intentional harmful changes known as vandalism as well as naturally occurring disagreements on topics which don't have an agreed upon viewpoint, known as *controversies*. While much work has focused on identifying vandalism, there has been little prior work on detecting controversies, especially at a fine granularity. Knowing about controversies when processing user-generated content is essential to understand the quality of the data and the trust that should be given to them. Controversy detection is a challenging task, since in the highly dynamic context of user updates, one needs to differentiate among normal updates, vandalisms and actual controversies. We describe a novel technique that finds these controversial issues by analyzing the edits that have been performed on the data over time. We apply the developed technique on Wikipedia, the world's largest known collaboratively generated database and we show that our approach has higher precision and recall than baseline approaches as well as is capable of finding previously unknown controversies.**

I. Introduction

The advent of Web 2.0 and related social applications have transformed the average Internet user from a passive data consumer into an active data producer. This was a catalyst for the birth of collaborative content creation. Documents such as news blogs, computer source code and informational manuals increasingly rely on multiple users to contribute content. The value proposition of massive open collaboration, apart from the significant reduction in financial costs, is that it enables information to be kept up-to-date, sometimes within seconds of occuring events, and content to be more complete, informed and pluralistic. Of course, allowing a broad base including non-professionals to edit content opens the door to unreliable information. Fortunately, recent studies have shown that the "wisdom of crowds" can lead not to chaos but to surprisingly high quality data [1]. This occurs when competent and well-intentioned users outnumber ill-intentioned, careless, or misinformed ones, by correcting [2] and completing [3] information.

Wikipedia is heralded as a success story of collaboratively edited content. Nonetheless, a number of specific data quality issues arise in it. Apart from normal edits, reverts, deletions and so on, there are cases of intentionally harmful edits known as vandalism. Moreover, situations where there are incompatible viewpoints on the same issue may arise which in turn result in disputes or *controversies*. Ideally, a dispute will eventually converge to a community-accepted consensus after undergoing discussion between the disagreeing authors. However, disputes can sometimes degenerate into so-called *edit wars* [4] which disrupt progress towards improving the article.

Controversies in Wikipedia come in all shapes and sizes. Sensitive entries such as "Arab-Israeli Conflict" have, not surprisingly, been rife with controversial content involving the number of casualties, the proper calculation of Israeli per-capita GDP, etc. However, controversies often arise within articles not known to be sensitive, such as whether the word "the" should be uppercased in (the music band) "The Beatles" [5]. In many if not most instances, controversies are not reported and sometimes most authors and readers are not even aware of them.

Identifying controversies, and distinguishing them from vandalism, is important for several reasons: to give disputes a chance to be aired out rather than squelched; to set appropriate edit policies and raise the bar for supporting evidence when needed; and to provide the reader a more nuanced understanding of content. For Wikipedia specifically, controversy detection can play a significant role. It can help to maintain Wikipedia's neutral point of view policy, which strives for objectivity, meaning that articles should be fair to all sides (though not necessarily equal, in the case of minority views), and that controversial content requires more rigorous documentation.

While much work has focused on identifying vandalism (see, e.g., [2][6][7]) the only prior work on detecting controversies [8][9][10] focused on ranking pages according to their level of controversy. In contrast, in this work we focus on finding *fine-grained controversies on pages*. Towards the goal of better understanding the quality of Wikipedia articles and the trust that should be given to them, we study the problem of automatically identifying fine-grained controversies, based on the revision histories maintained as part of the Wikipedia database. The definition of a *controversy* is as follows: *a prolonged dispute by a number of different people on the same subject*.[1] Translating that definition into an operational technique needs to address several challenges. First, an edit may represent a signal of different kinds such as new information or a natural evolution in the existing

[1] http://en.wikipedia.org/wiki/Controversy

content, an act of vandalism where the author has an intention to destroy the integrity of content, or an alternate viewpoint on a topic which allows several incompatible perspectives. Second, tracking the same controversial topic across many revisions is difficult because often that topic can move around the page landscape (e.g. from the introduction to history sections). Third, there are cases of short-lived intense sparks of disagreement where it is difficult to differentiate between vandalism and controversies. Unfortunately, the textual data is very ambiguous, applying natural language processing in this context is a very difficult task and often it is hard to obtain highly accurate results. Fourth, controversy identification is a challenging computational problem due to the large number of versions in the history of a document. For example, at the time of writing, Wikipedia contained more than 100,000,000 revisions in 4,500,000 pages and was growing at a rate of 30,000 new articles per month.

We effectively address these challenges as follows.

- We provide the first operational notion of a *fine-grained controversy* in a collaboratively edited database based on the novel idea that the associated dispute is expressed as back-and-forth substitutions of content embedded within a similar context.
- To identify significant controversies over time, we propose to augment the substitution context with the support for the controversy, the duration of the controversy and the set of authors involved.
- We investigate two different models for characterizing evolving documents: one sees the document as (space-separated) "words"; the other sees the document as a sequence of the hyperlinks to other Wikipedia pages (which constitute the majority of all references) contained within a document. The latter exploits the labeling (and, therefore, normalization) of semantic concepts implicit in links and hence allows for more accurate results.
- We design an algorithm for controversy detection that uses these document models to cluster edits throughout the revision history of a page and identify controversies.
- We test our algorithms on the entire English Wikipedia dump (above 7TB) along with user studies conducted on Amazon Mechanical Turk and demonstrate that controversies can be effectively identified from a document's version history in a scalable manner.

Our experiments have shown that our approach has higher precision and recall than baseline approaches which are capable of finding fine-grained controversies. Moreover, our approach finds many previously unknown controversies (236% of the number of known controversies in the experiments) which are equally or more important than the already known ones. For example, in the Wikipedia page about Chopin our method detected not only the known controversy about his origin but also the controversies about his date of birth and his photograph by Louis-Auguste Bisson.

The paper has the following structure. We describe the related work in Section II. We provide the formal definition of the concepts and of the problem we solve in Section III and our solution in Section IV. Section V contains our extensive experimental results. We conclude in Section VI.

II. Related Work

The problem of data quality in collaboratively-edited databases has been studied a lot in the past. Mainly, there are four types of problems in such databases, namely vandalism detection, stability prediction, trust/reputation modeling and controversy detection.

Vandalism detection [6][2][7] focuses on identifying those edits which are performed in order to intentionally destroy the content of a Wikipedia page. Chin et al. [2] identified different kinds of vandalism based on the type of action (delete, insert, change, revert), the type of change (format, content) and the scale of editing (mass deletion or mass insertion). Machine learning techniques were used to classify edits as either blanking, large-scale editing, graffiti and misinformation, using text features such as the number of known words, perplexity value, number of known bigrams and so on.

For *stability prediction*, Druck et al. [11] examine how different features affect the longevity of an edit including whether an author is registered or not, the total number of reverts done by the author, the addition of links, etc. Based on this, they train a classifier to label edits in one of three categories: revert, 6-hour longevity and 1-day longevity.

The *reputation and trust* of collaboratively generated content were studied in [12]. The authors proposed an algorithm to compute the reputation values of Wikipedia authors where each author increases her reputation if her edits are not changed by the subsequent edits. This is interpreted as a sign of the approval that the edit should remain on the page and thus the user who did it deserves that her reputation grows. In their subsequent work [13], the authors leverage the author reputations to compute trust levels for content, where the assumption is that a high reputation author is more likely to contribute trustable content. Trust is evaluated based on the algorithm's ability to predict stable content having longevity. This work led to an online system[2] which indicates the trust level of Wikipedia content using colors.

The problem of *controversy detection* has been studied only at the page level, i.e., ranking pages as controversial or not, whereas in this work we propose the first method which detects fine-grained controversies along with the information of when they appeared, what is the topic and who was involved in it. A machine learning approach was used by [8] in order to identify the amount of conflict on a page using the human-labeled controversy tags as a ground truth. They employed such features as the number of revisions, unique editors, page length and so on. A more statistical approach was proposed in [9]. It uses the statistics of deletions and insertions along with the Mutual Reinforcement Principle whereby frequently deleted content is considered more controversial if it appears on a page whose controversy level in the past was low, and if

[2]http://www.wikitrust.net/

its authors were involved in fewer past controversies. Formulas to compute the level of controversy of a page are derived based on these assumptions.

Other work focuses on revert statistics – the number of authors who revert an article back to a previous version – as a crude measure to detect controversies at the page level [14]. Another work is [15], which proposes a method to find pages with "arguments" based on bipolarities in the so-called "edit graph" of the (history of a) page in which nodes correspond to authors and edges correspond to pairs of authors where either one author deletes content written by another or restores the content that was written by another.

A number of approaches used visualization as a way to help administrators to find controversial pages. Viegas et al. [16] designed a tool to visualize the history of a page to identify vandalism and/or conflict. For each version in a page's history, the tool plots colored sequences of blocks, each block corresponding to a token, where each token is assigned a color that stays fixed over time. This allows the user to observe abnormal content changes between versions. [17] visualizes revision graphs which in turn helps to understand who are the dominant users and what are the conflicting topics of a page. Moreover, [17] allows for visual analysis of multipart conflicts, i.e. the conflicts where many opposing opinions exist. The visualization of controversies was also employed in [18] in order to study cross-linguistic controversies in Wikipedia.

III. Modeling Collaborative Data

Collaboratively generated content in the real world is a collection of entries created and edited by many editors of some application. These applications typically keep also the version history of the entries throughout the edits applied on them. The entries can be text, html pages, wiki pages, office documents, etc. To process these entries, we need to convert them to documents in our model. We call this process *tokenization*. A *token* is a string that is considered undivided. A *document* D is a non-empty sequence of tokens. There are different types of tokens that can be used. Each token type determines a *model*. The choice of the model is based on what best fits the task at hand. In this work, since our focus is mainly Wikipedia, we focus on two specific models: the *text* and the *link* models. The text model considers as tokens single words, meaning that every content entry is turned into a sequence of individual words. As we will show in the experiments, the text model has the advantage of capturing every change that may occur in the textual content entries (excluding images and other multimedia), since it does not disregard any part of them; however, it increases ambiguity. Note that even if the model considers single words, our algorithm is able to capture modifications involving more than one word, e.g. "New York" being replaced by "Connecticut".

The *link model* considers only the links (i.e., references) to other entries, pages or entities, meaning that during tokenization the original entry is stripped of everything that is not a link. In contrast to previous studies on Wikipedia content that have considered only the text model [9], [14], we also consider the link model, a fundamental novelty of our approach. The link model is a more semantic approach compared to the text model since the referenced entries are typically semantic concepts. In the case of Wikipedia, every entry describes unambiguously a concept or an entity, thus, the meaning of every reference[3] (link) in Wikipedia has clear semantics. At the same time, since Wikipedia has become one of the largest sets of collaboratively-edited data publicly available nowadays, almost every semantic concept has its respective entry in it, and with the Wikipedia language highly facilitating the creation of links, the density of links is such that it allows them to capture most of the page semantics in just a sequence of links.

Example 3.1: Consider the following Wikipedia content where the underline represents links. "A Caesar salad is a salad of romaine lettuce and croutons dressed with parmesan cheese, lemon juice, olive oil, egg, Worcestershire sauce, garlic, and black pepper. It is often prepared tableside. It is generally attributed to restaurateur Caesar Cardini, an Italian immigrant who operated restaurants in Mexico and the United States. Cardini was living in San Diego but also working in Tijuana where he avoided the restrictions of Prohibition." *When tokenized according to the text module it will become the sequence of words as seen in the previous sentences. When tokenized according to the link model, it will become the sequence:*⟨salad, romaine lettuce, croutons, parmesan cheese, olive oil, Worcestershire sauce, Caesar Cardini, Italian San Diego Tijuana Prohibition⟩.

The *position* of a token in a document is an index number identifying its location in the sequence of tokens that compose the document. Documents can in general be modified through a series of *primitive changes* on tokens including insertions, deletions, substitutions, reorderings, etc. We focus only on substitutions, where existing token(s) are replaced by others. This is predicated on the novel idea that controversies are typically expressed via substitutions, that is, authors having opposing viewpoints tend to argue back-and-forth over alternate content, rather than just inserting new content or deleting old content. Indeed, our data analysis in Section V supports this novel idea.

Definition 3.2: Given a document $D{=}w_1 w_2 \dots w_n$, a *substitution* of its sub-sequence of tokens $E_b{=}w_p w_{p+1} \dots w_{p+k}$ with the sequence of tokens $E_a{=}w'_1 w'_2 \dots w'_q$ (both E_a and E_b are not empty) is an action that converts the document D to the document $D'{=}w_1 w_2 \dots w_{p-1} w'_1 w'_2 \dots w'_q w_{p+k+1} \dots w_n$. The *subject* of the substitution is the pair $\langle E_b, E_a \rangle$, and its *position* is the number p. ∎

Changes are performed by authors at specific times. We assume set $\mathcal{A}$ of authors, and time domain $\mathcal{T}$, and we define an edit on a document to be the triple of the substitution change along the author that performed it and the time s/he did so.

[3]Note that we consider only internal links which constitute the majority of all references in Wikipedia.

Definition 3.3: An *edit* e is a tuple $\langle c_e, a_e, t_e \rangle$, where c_e is the *substitution*, $a_e \in \mathcal{A}$ is the *author*, and $t_e \in \mathcal{T}$ is the *time* of the edit. The *subject* and *position* of the edit are denoted as $subj(e)$ and $pos(e)$, respectively. ∎

We make the natural assumption that no more than one author can perform changes on a document at the same time, but we do allow multiple changes to happen at the same time from the same author. The rationale behind this is that authors are typically performing a number of changes in the document and then submit them all together (i.e., by clicking on the save button). We denote by $D_b \xrightarrow{e} D_a$ the fact that an edit e applied to a document D_b results in the document D_a. For a sequence of edits $E = e_1, e_2, \ldots, e_m$, having $time(e_1) = time(e_2) = \ldots = time(e_m) = t$, the notation $D_p \xrightarrow{E} D_a$ is a shorthand for $D_p \xrightarrow{e_1} D_1 \xrightarrow{e_2} D_2 \xrightarrow{e_3} \ldots \xrightarrow{e_m} D_a$. By abuse of notation, $time(D_a)$ and $time(E)$ will refer to the time t.

Definition 3.4: A *version history* h is the sequence of documents D_0, D_1, …, D_m such that $D_0 \xrightarrow{E_1} D_1 \xrightarrow{E_2} D_2 \xrightarrow{E_3} \ldots \xrightarrow{E_m} D_m$. E_1, E_2, …, E_m is an inferred progression of edit sequences such that: (i) for every two edits $e, e' \in E_i$, $time(e) = time(e')$; and (ii) for every two edits $e \in E_j$ and $e' \in E_i$, with $j < i$, $time(e) < time(e')$, for $i = 2..m$ and $j = 1..(m-1)$. The *edits of the page history* is the set $E_1 \cup E_2 \cup \ldots \cup E_m$ ∎

Definition 3.5: A *collaboratively edited database*, or *database* for short, is a set of version histories $\{h_1, h_2, \ldots, h_k\}$. Each version history h_i is referred to as a *page* P_i, and the documents in h_i are the *versions* of the page P_i. ∎

Note that given page P, there may be more than one set of sequences that can serve as edits of the page history. For many practical applications, the exact edits may not be important, or may not be possible to know. For instance, in Wikipedia, the authors edit the pages and commit to the system the edited version without any extra information. It is up to the system to compare the previous version with the one committed by the author and generate a set of edits, if it is needed, that when applied to the former generates the latter.

There are many definitions and interpretations of what a controversy is. A widely accepted definition is the one provided by Wikipedia itself which states that a *controversy* is a prolonged dispute by a number of different people on the same subject. Since this is a semantic definition, to discover controversies there is a need for an operational interpretation. In collaboratively generated databases like Wikipedia, such disputes can be observed through the edits that the different authors have performed on the pages. Thus, the problem of identifying controversies boils down to the identification of groups of edits that represent such disputes.

Identifying edits on the same subject matter is a challenging task since there is no way to uniquely identify topics in a document that will allow them to be tracked over time. A straightforward idea is to use the positions of edits as a reference; unfortunately, positions from one version to another may change significantly. A second idea is to use only the subject of the edit, that is, the change that actually took place. Unfortunately, this is not a viable solution, since the same content may appear in different places in a document with very different meanings. For example, the fact that "Polish" was replaced by "French" in two different versions of a page does not mean that the two users that performed these two edits were referring to the same thing. The first may refer to, say, the nationality of a person while the second may refer to the language in which a manuscript is written. The only way to understand this is to also consider surrounding content to take the right semantics into account. We refer to this surrounding content as the *context* of the edit. We posit that two edits with the same or very similar context likely refer to the same topic. An important issue related to context is to decide how much surrounding content should be considered. One may think that the more tokens considered as context the better. However, too many tokens increase the risk of including tokens modified by other edits, reducing the similarity of context of the edit to others on the same subject matter. Therefore, we introduce the notion of the *radius* of a context.

Definition 3.6: The *context* of radius r of an edit e with the substitution of size $k+1$ on a document D with tokens $w_1 w_2 \ldots w_n$, $ctx(e)$, is the sequence $w_{p-r} w_{p-(r-1)} \ldots w_{p-1} w_{p+k+1} \ldots w_{p+k+(r-1)} w_{p+k+r}$ of tokens, where $p = pos(e)$. ∎

Note that since the context does not contain the subject of the edit, it can successfully identify edits about the same subject matter even in cases in which traditional semantic similarity techniques may fail. For instance, a substitution of the expression "his birthday is unknown" with "born in March 1821" and of the expression "born in circa 1800" with "lived from March of 1821" are essentially saying the same thing. A direct matching of the two edits will most likely conclude incorrectly that they are not related,

For measuring the similarity of the contexts of edits we use the well-known Jaccard similarity [19], and we use it as a metric of similarity between edits.

Definition 3.7: The *similarity* between two edits e_1 and e_2 is $\frac{|S_{ctx(e_1)} \cap S_{ctx(e_2)}|}{|S_{ctx(e_1)} \cup S_{ctx(e_2)}|}$, where S_σ denotes the elements of sequence σ represented as a set. ∎

Example 3.8: Suppose that on the 7th of May of 2008, the editor Mogism replaced in the Wikipedia entry from Example 3.1 the text Caesar Cardini *with* Julius Caesar. *This modification is modeled by the edit* $e = \langle \langle Caesar\ Cardini, Julius\ Caesar \rangle, Mogism, 2008/05/07 \rangle$. *The subject of the edit is the pair* $\langle Caesar\ Cardini, Julius\ Caesar \rangle$. *Assuming context radius is equal to 2, for the text model the context of the above edit is* $\langle to, restaurateur, an, Italian \rangle$ *while for the link model it will be* ⟨olive_oil, Worcestershire_sauce, Italian, San_Diego⟩.

To quantify the time span of a set of edits E, we define the *duration* of the set to be the time $time(e_m) - time(e_1)$, where

Algorithm 1: Controversy Detection Algorithm (CDA)

Input: h: A page history
Output: Set $\mathcal{C}$ of controversies

(1) // Edit extraction
(2) $E \leftarrow \emptyset$
(3) **foreach** i=1..($|h|-1$)
(4) $t \leftarrow$ Time $h[i+1]$ was created
(5) $u \leftarrow$ User that created $h[i+1]$
(6) $\mathcal{E} \leftarrow$ MYERSALG($h[i]$,$h[i+1]$)
(7) **foreach** j=1..($|\mathcal{E}|-3$)
(8) **if** $\mathcal{E}[j]$ is an $eq \wedge \mathcal{E}[j+1]$ is a $del \wedge$
(9) $\mathcal{E}[j+2]$ is an $ins \wedge \mathcal{E}[j+3]$ is an eq
(10) *Let* $del(E_d)$ be the $\mathcal{E}[j+1]$
(11) *Let* $ins(E_i)$ be the $\mathcal{E}[j+2]$
(12) **if** $|E_i| < 5$ and $|E_d| < 5$
(13) $E \leftarrow E \cup \langle\langle E_d, E_i\rangle, t, u\rangle$

(15) // Eliminate edits with low user support
(16) **foreach** $e \in E$
(17) $E_{eq} \leftarrow \{e' | e' \in E \wedge subj(e) = subj(e')\}$
(18) $U_e \leftarrow \{user(e') | e' \in E_{eq}\}$
(19) $U_e \leftarrow$ ELIMINATEDUPLICATES(U_e)
(20) **if** $|U_e| < k_{thrshld}$
(21) $E \leftarrow E$ - E_{eq}

(23) // Cluster edits based on context and using as similarity
(24) // metric among the edits the similarity of their contexts
(25) $\mathcal{E} \leftarrow$ CLUSTER(E, "$context$", $r_{thrshld}, cutoff^{ctx}_{thrshld}$)

(27) // Cluster & merge the sets of edits based on the
(28) // subject of their edits
(29) $\mathcal{C} \leftarrow \emptyset$
(30) $\mathcal{M} \leftarrow$ CLUSTER($\mathcal{E}$, "$subject$", $cutoff^{sub}_{thrshld}$)
(31) **foreach** cluster $M \in \mathcal{M}$
(32) $\mathcal{C} \leftarrow \mathcal{C} \cup flatten(M)$
(33) $\mathcal{C} \leftarrow top_k(\mathcal{C})$

e_1 and e_m are the earliest and latest edits in E, respectively. Similarly, we introduce the notion of *plurality* of a set of edits as the number of distinct users that have performed these edits, and the *cardinality* as the number of edits in E.

IV. DETECTING CONTROVERSIES

To identify controversies, we need to look for edits in the history of a page that are about the same subject matter, have taken place in a period of a certain duration and have been performed by at least a certain number of users. Each group of edits that has these characteristics is an indication of a controversy. Thus, our controversy detection algorithm returns a set of sets of edits.

Although one can look at the pool of all the edits of the pages of a database, in this work we focus on controversies identified within a specific document. Thus, we restrict our analysis on the sets of edits in the history of each specific page independently. Algorithm 1 (CDA for short) provides an overview of the steps we follow. The variables in the algorithm with the subscript *thrshld* are configuration parameters the values of which are specified offline.

Edit Extraction. The first step that needs to be done is to identify the edits that have taken place in the history of a page. Recall that we typically have the documents in the history and not the edits. The edits are discovered by comparing consecutive documents in the history. For each pair of such consecutive documents (i.e., versions) Myers' algorithm [20] is executed. The output of the algorithm is an edit script. The elements in the script can be of three different types: (i) $eq(E)$ indicating that a set of tokens E remained unchanged between the two page versions that were compared; (ii) $del(E)$ indicating that the sequence of tokens E of the first version was deleted; and (iii) $ins(E)$ indicating that a sequence of tokens E was inserted in the second version. In the generated edit script we are looking for patterns of a del followed by an ins that are located between two eq. This pattern indicates a substitution which is what is of interest to us. From the discovered patterns of this type, we do not consider those where the number of tokens in the ins or the del operation is more than 5. The intuition behind this which is based on a number of experiments that we have performed is that edits involving large pieces of text do not indicate controversies but are often vandalisms. The remaining patterns are turned into substitution edits. (lines 1-13)

Eliminate Edits with Low User Support. From the edits generated in the previous step, we eliminate those that have not been repeated by at least $k_{thrshld}$ users. The idea is that if an edit does not enjoy any broad support, i.e., has been made by only one (or extremely few) users, it is some personal opinion and not a point of view reflected by a considerable set of users. In counting the number of times that an edit has been repeated by a user, we consider only the subject of the edit and not its context, since the same user may perform the same edit in different places and in different versions. A typical value we consider for $k_{thrshld}$ is 2, i.e., eliminating single-user edits. (lines 15-21)

Cluster Edits Based on Context. The remaining edits go through a process that tries to group together edits that are about the same subject matter. For this task a clustering algorithm is employed. The choice of the clustering algorithm is an orthogonal issue. CDA uses it as a black-box. However, for our implementation we have chosen the well-known DBSCAN [21] algorithm. The algorithm requires a metric for the similarity among the elements it needs to cluster. This similarity is based on the context of the edits and is measured using the Jaccard similarity as described in the previous section. The step is configurable using the context radius $r_{thrshld}$ and the threshold level $cutoff^{ctx}_{thrshld}$ that the DBSCAN algorithm uses to decide on whether two elements are similar enough to be in the same cluster. The outcome of

parameter	range	default value
model	`link`, `text`	`link`
$r_{thrshld}$	2,4,6,8	8
$cutoff^{ctx}_{thrshld}$	[0 ... 1]	.75
$cutoff^{sub}_{thrshld}$	[0 ... 1]	.8
$k_{thrshld}$	1,2,3,4,5	2

TABLE I
CDA CONFIGURATIONS PARAMETERS

this step is a set of groups of edits. Each one of these groups is considered to represent a controversy. (lines 23-25)

Cluster and Merge the Sets of Edits Based on the Subject. Since the clustering performed in the previous step was based on the context of the edits and not on their subject, it may be the case that a controversy about a topic that appears in different parts of the document may end up being recognized more than once, i.e., we may end up with different groups of edits that talk about the same subject matter. To reconcile these groups into one, so that every controversy is represented by one group of edits, the clustering algorithm is executed once again, but this time on the sets of edits and not on the individual edits themselves. The similarity function used among the groups is again the Jaccard similarity but this time applied on the sets of edits that each group contains. The similarity function finds the edits that are common to two different groups by a comparison on their subject. (lines 27-30)

Before they are returned to the user, the found sets of edits are ranked based on the level of controversy they describe. This level of controversy can be determined by many factors. In our implementation we have used a number of methods such as the cardinality, duration and plurality because the assumption is that a controversy is stronger when it involves more users, lasts longer and is repeated more times. (line 31-33)

V. EXPERIMENTAL EVALUATION

For the experimental evaluation we use the Wikipedia dump from December 2013. It consists of a set of entries in Wikitext (or Wiki-markup) format each one with all its historical versions. For each version the information on the time and the author that submitted the version is also available. We refer to this dataset as the *full dump*.

We have implemented CDA (see Algorithm 1) in Java and we conducted experiments on a 4GHz CPU PC with 4GB memory running Ubuntu 12.04. For the discovery of links in the Wikimedia content of the Wikipedia entries we use the JWPL Wikimedia parser[4]. The CDA parameter configuration is summarized in Table I.

A. Sources of Ground Truth

To study the effectiveness of CDA, it is required to know the ground truth, i.e. which Wikipedia entries, or what part of the entries are actually controversial. To obtain this information we used three different sources: (i) the list of textual descriptions of controversies provided directly by Wikipedia; (ii) the controversy related templates as inserted by Wikipedia authors wherever it is believed that a controversy exists; and (iii) the controversies identified by users in the Amazon Mechanical Turk (AMT) evaluation experiments we have conducted.

[Wikipedia Provided Controversies (WPC)] Wikipedia provides a list[5] of controversies from which we manually extracted all Wikipedia entries (263) with controversies. As a result we produced a list of pairs ⟨entry, description⟩, with the *entry* being the Wikipedia entry and *description* being the textual description of the controversy in the specific entry. As an example, one such pair is about the Wikipedia entry of `Freddie Mercury` and the respective description explains that there is a controversy about the ancestry of the famous singer on whether he is the most famous Iranian, Indian, Parsi or Azeri rock star.

The Wikipedia provided controversies are well-known and broadly accepted controversies, nevertheless the list is by no means complete.

[Template-indicated] Another source of controversies that has been used in other works [9] is based on the dispute template messages[6] that editors of the Wikipedia entries leave in the text to notify other editors and readers about an issue related to the page, a paragraph or some words. There are many types of templates. Among them, we consider only those directly related to disputes, i.e., the `contradict`, `contradict-other`, `disputed`, `dispute about`, `disputed-category`, `pov`[7], `pov-check`, `need-consensus`, `disputed-section`, `pov-section`, `pov-intro` and `pov-statement`.

For a Wikipedia page with controversy templates, we need to quantify how controversial the page is. To do so, we measure the number of controversial templates in all the versions in the history of the page. In particular, we use the *Article Tag Count* (ATC) [9] metric:

$$ATC = \sum_{i=1}^{n} c_i \quad (1)$$

where n is the number of versions of the entry and c_i is the number of controversy related tags in version i.

Templates offer many more controversies than the list of known Wikipedia controversies since they are flexible and easy to use. Nevertheless, despite their flexibility, a recent study has shown [9] that they are not extensively used, thus, the information they provide on the controversies is not complete either.

[User-specified Controversies] A third form of controversy source is the set of users employed to conduct the user-based evaluation of CDA. We refer to the set of controversies

[4]http://code.google.com/p/jwpl/

[5]http://en.wikipedia.org/wiki/Wikipedia:Lamest_edit_wars

[6]http://en.wikipedia.org/wiki/Wikipedia:Template_messages/Disputes

[7]Point of View.

provided in that way as the user specified controversies. The user-specified controversies have the advantage that they can provide more details on whether and why a Wikipedia entry (or part of it) is controversial, and on whether the entry is more controversial than another. The set of pages which the users of our evaluation can judge is considerably smaller then the set of pages of Wikipedia. Practically it is difficult to run a large scale evaluation since there are thousands of controversies in Wikipedia entries. Furthermore, users have to be domain experts in order to provide a well-founded and trustworthy opinion on whether a piece of a Wikipedia entry that seems to have been edited extensively is controversial or not.

In this controversy source we used the Amazon Mechanical Turk[8]. We built a GUI which worked in two steps. First, we asked the users to match the controversies found by CDA and the respective textual descriptions of the known controversies. Second, for every edit in every set that CDA has returned, the users are asked to provide a tag (topic) indicating what they believe is the controversy that the edit is about, or provide the tag *unspecified* if they do not find any (see a GUI screenshot for that case in Figure 1). Those tagged with the

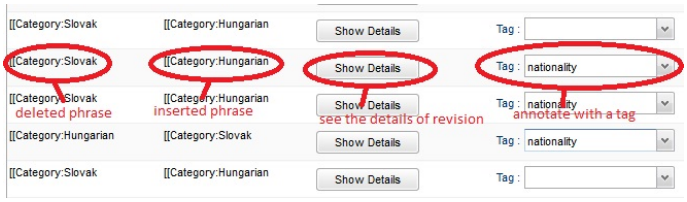

Fig. 1. GUI for user-based evaluation

tag *unspecified* are discarded and the rest are re-grouped in a way that all the edits with the same tag end up in the same group and no two edits with different tags end up in the same group. The formed groups then play the role of ground truth. We refer to this ground truth as $ideal(C)$.

For every task we asked three distinct users (totally, 12 distinct AMT users participated in this user study) to provide their feedback. We report the average values of measured metrics (e.g. precision, Rand index, and so on). In the majority of evaluations, the users were in agreement with each other (the differences of metrics are less than 10%). However, for the final results we eliminated a small number of outliers (less than 1% of total evaluations).

B. Measuring CDA Success

To evaluate the results of CDA, we need to compare them with the ground truth. For this we use a number of different metrics to measure its different quality aspects.

Given a controversy, i.e., a set of edits, c found by CDA, and the set of tags that the user has provided to the sets of edits in all the found controversies, we define the noise of c, denoted as $noise(c)$, to be the set of edits tagged with the *unspecified* tag. On the other hand, $clean(c)$ denotes the set of edits tagged with some topic specific tag, i.e. $clean(c) = c \setminus noise(c)$. The controversy c is said to be *clean* if $noise(c) = \emptyset$ and *noisy* otherwise.

[Noise/signal ratio] To realize the amount of edits among those returned by CDA that were not categorized to any topic, i.e., those tagged with the *unspecified* tag, we use the noise/signal ratio:

$$noise/signal = \frac{\sum_{i=1}^{k} \mid noise(c_i) \mid}{\sum_{i=1}^{k} \mid c_i \mid} \tag{2}$$

where k indicates the number of controversies returned by CDA. A noise/signal ratio of 1 indicates that there are no edits in topK results that could be successfully tagged with a tag describing a topic.

[AvePr] To measure the precision of controversies found by CDA, we use the average precision [22]. In this experiment, a controversy is said to be *relevant* if it has at least one edit which is annotated by a topic tag. Note that the average precision accounts both for the number of relevant controversies and their order, i.e. the higher the relevant controversies in the result list, higher the average precision.

$$AvePr = \frac{\sum_{i=1}^{k} P(i) rel(i)}{\#\ of\ relevant\ controversies} \tag{3}$$

where k is the number of retrieved controversies, $P(i)$ is the precision at position i and $rel(i)$ is a function which takes 1 if the ith-result is a relevant controversy and 0 otherwise. Note, that the denominator is the number of relevant controversies in topK results and not the total number of relevant controversies in the dataset.

[Rand index] To compute the distance between the retrieved controversy C and the ideal $ideal(C)$ we employ a clustering evaluation metric, namely the Rand index [23]. Informally, the Rand index measures the accuracy of the retrieved controversies according the user provided ground truth.

$$Rand = \frac{a+b}{\binom{\sum_{i=1}^{k} \mid clean(c_i) \mid}{2}} \tag{4}$$

where $\sum_{i=1}^{k} \mid clean(c_i) \mid$ is the total number of topic edits in the top k controversies, a is the number of edit pairs that are in the same group in $clean(C)$ and in the same group in $ideal(C)$ and b is the number of edit pairs that are in different groups in $clean(C)$ and in different groups in $ideal(C)$. For example, the Rand index of 1 indicates that all edits in $clean(C)$ form the same clusters as in $ideal(C)$.

[# of distinct controversies] For a set of controversies, C, we compute the total number of topics which can be found in it, i.e.

$$\#\ of\ distinct\ controversies = \mid ideal(C) \mid \tag{5}$$

The next two metrics are computed using the user feedback on a controversy and not edits. Thus the users either specify whether the controversy matches the given textual description (*Recall on WPC*) or the controversy corresponds to some controversial topic (*Precision on a set of pages*).

[8]https://www.mturk.com/mturk/welcome

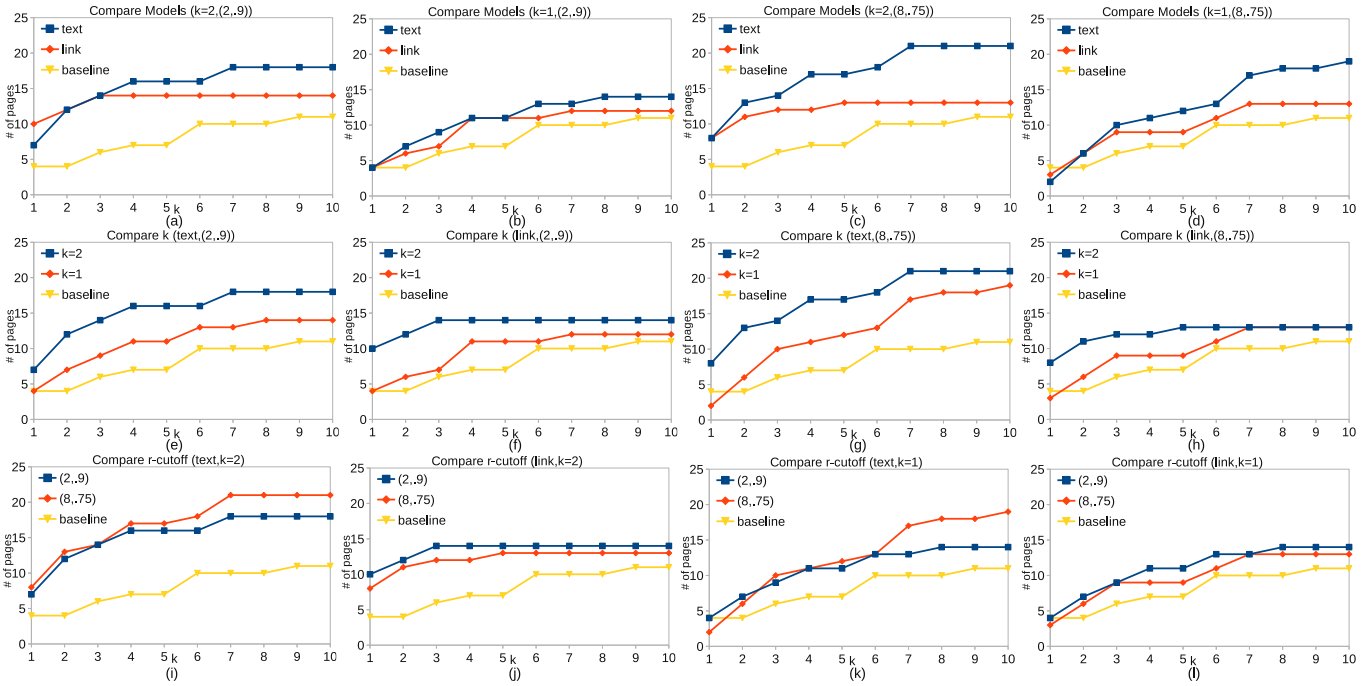

Fig. 2. The recall experiment on WPC where the first row (a,b,c,d) compares models, the second row (e,f,g,h) compares $k_{thrshld} = 1$ and $k_{thrshld} = 2$, and the third row (i,j,k,l) compares the `(2,.9)` and `(8,.75)` ($r_{thrshld}$,$cutoff^{ctx}_{thrshld}$) parameter pairs.

[Recall on WPC] To compute the recall when we know one controversy per page, i.e. the case of the list of WPC, for a given number of retrieved controversies k we count the number of pages which have the corresponding controversy in its first k controversies. Hence, the recall is computed as follows:

$$Re = \frac{\#\ of\ pages\ with\ controversy\ in\ topk\ results}{n} \quad (6)$$

where n is the number of pages we know controversies in.

[Precision on a set of pages] For a given number of retrieved controversies k and a ranking function f (for possible ranking functions see Section IV) we measure the precision on a set of n pages as follows. First we rank the controversies within a page using f and pick the top1 controversy. Second we rank pages using f of top1 controversies of pages. Finally, the precision is the ratio between the number of relevant controversies for topK pages and k.

$$Pr = \frac{\#\ of\ pages\ for\ which\ top1\ is\ relevant}{k} \quad (7)$$

Even though we have both precision and recall we do not report F-measure since those metrics use different ground truth sources, i.e. for recall we use the list of WPC and for precision we conduct the user study. Moreover, the recall is computed with the assumption that for every page we know only one controversy (its textual description) which makes recall completeness tied to WPC.

C. CDA Experimental Results

We study the recall of the algorithm on the set of pages with WPC (Section V-C1). The precision of our algorithm is measured both on the set of pages from WPC (Section V-C2) where we use user feedback as the ground truth and on the full dump of Wikipedia pages (Section V-C3). Finally, we present the results where we experiment with edits as a way to rank pages by the controversy level (Section V-C4). As a baseline we consider the technique which was widely applied to detect controversies at the page level (e.g. [9]), namely use the number of revisions as an indicator of controversies. We translated this idea to the individual controversy level. More specifically, we collect all text edits in all revisions of a page, then we cluster those edits by their content and rank those clusters by the number of revisions the respective edits appeared in. Note, that other page level approaches (e.g. [8][10]) can not be used for fine-grained controversy detection and adapting their ideas is considered as future work.

1) Recall of CDA: We are first interested in measuring the recall of the CDA, i.e. what fraction of WPC the algorithm is able to retrieve.

As ground truth we use the list of WPC where we select 25 pages for which our algorithm produces at least one match in top 10 results under at least some parameter setting (for details on the parameter settings see below). We run the algorithm on those 25 pages and retrieve top 10 controversies per page ranked by the cardinality of controversy. Then every controversy of a page is validated by the user who checks whether the corresponding textual description of the page matches the retrieved controversy. The reason for using only 25 pages is that the validation is an extremely time consuming process and in this experiment we do not compare with

external approaches but rather we aim at comparing CDA under different parameter values.

For the recall measurement we test both the CDA with text (`text`) and link (`link`) models and the `baseline` which is a grouping of all edits of a page ordered by the number of revisions they occurred in. The parameters of CDA are chosen as follows: $k_{thrshld}$ of 1 and 2, $r_{thrshld}$ of 2 and 8, and $cutoff^{ctx}_{thrshld}$ of .9 and .75. The above choices of $r_{thrshld}$ and $cutoff^{ctx}_{thrshld}$ are based on the following empirical observations. By varying the context $r_{thrshld}$ from 2 to 8 and $cutoff^{ctx}_{thrshld}$ from .1 to 1 we observe that the highest recall is detected at two pairs of values. First, when $r_{thrshld}$ is very small and the $cutoff^{ctx}_{thrshld}$ is large `(2,.9)`. That is the case when the edits are clustered by considering only a few neighboring tokens but requiring them to be almost the same. Second, when $r_{thrshld}$ is large but $cutoff^{ctx}_{thrshld}$ is relaxed `(8,.75)` which means that we use more tokens around but we are not very strict to have the tokens be exactly the same.

The CDA recall for varying k under different parameter values is shown in Figure 2. Each column represents the comparison of two values of one parameter. Thus the first row (Figure 2 (a,b,c,d)) compares the `text` and `link` models under all combinations of the $k_{thrshld}$ and $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ parameters. In the second row (Figure 2 (e,f,g,h)) we report the comparison of the recall values with two values of $k_{thrshld}$: 1 and 2. Finally, the comparison of `(2,.9)` and `(8,.75)` $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ is in Figure 2 (i,j,k,l).

In the above experiments, the `text` model outperforms the `link` model which means that `text` is able to detect a bigger fraction of WPC. Clearly, we observe that $k_{thrshld} = 2$ improves the recall for any model and $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ pair. Hence, $k_{thrshld}$ allows CDA to eliminate many noisy edits. Regarding the $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ pair, `(8,.75)` produces a slightly better recall (for the `(link,k=2)` the difference is only 1 page). In all parameter settings both the CDA `text` and `link` models significantly outperform `baseline`. The difference is higher when $k = 2$ and the data model is `text`.

In the next experiment we measure the recall on the entire set of pages with WPC (263 pages). We vary the number of retrieved controversies k from 1 to 10 and measure the recall as we did it in the previous experiment. As parameter values, we use the ones which show the highest recall values (see Figure 2), i.e. the `text` model , $k_{thrshld} = 2$ and `(8,.75)` $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ pair. The results are shown in Figure 3(a) where we report the recall for the clean/noisy controversies of CDA along with `baseline`.

From the full recall experiment on WPC, we conclude that CDA is able to retrieve a large portion of WPC (e.g. 117 out of 263 pages have the corresponding controversy as the top10 result). Surprisingly, the results show a small constant difference between the clean and noisy recalls which means that the algorithm mainly retrieves the controversies which have topic edits. The full recall of both clean and noisy CDA controversies are higher than `baseline`: for top10 the difference is 53 known controversies (more than 20%). Note that the recall of 45% (for $k = 10$) is computed on WPC and therefore it does not count the controversies which are not in WPC (e.g. there are many other controversies which are not documented by Wikipedia). In the next experiments we discuss those controversies in detail.

metric	link	text	link ins/del	text ins/del	baseline
noise/signal	**0.19**	0.25	0.64	0.57	0.75
AvePr	**0.91**	0.89	0.62	0.63	0.34
Rand	0.7	**0.74**	0.44	0.31	0.3
# of dist contr	65	**80**	29	25	17

TABLE II
NOISE/SIGNAL, AVEPR, RAND AND # OF DISTINCT CONTROVERSIES FOR THE `TEXT`,`LINK` DATA MODELS AND FOR THE TEXT (`TEXT INS/DEL`) AND LINK (`LINK INS/DEL`) MODELS WITH INSERTIONS AND DELETIONS, AND FOR `BASELINE` ON A SET OF 25 PAGES WITH WPC

2) Accuracy of CDA: As a second step, we aim at measuring the accuracy of CDA. As a dataset we use a set of 25 pages with WPC for which our algorithm produces at least one match in top 10 results under at least some parameter setting (the same set of pages that are used in the Section V-C1). As the source of controversies we conducted the user study where every edit is annotated with a topic or the unspecified tag (see Section V-A for details) by three different AMT workers. The reason for using only 25 pages is that manual annotation is an extremely time consuming process.

For the parameters we compare the effectiveness of the `text` and `link` models, where $k_{thrshld}$ and $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ parameters are assigned the values which maximize the recall in the experiment in Section V-C1, namely $k_{thrshld} = 2$ and `(8,.75)`, respectively. In addition, we also run the above experiment for insertions and deletions along with substitutions (`text ins/del`, `link ins/del`) to validate our assumption about substitutions as a good means to detect controversies (see Section III for details). Finally, we also present the results of our `baseline`, i.e. the grouping of all kinds of edits which are then ordered by the number of revisions they appeared in.

In this experiment, for every page we measure 4 metrics which address different aspect of the effectiveness of CDA: the noise/signal ratio (noise/signal), the average precision (AvePr), the Rand index (Rand) and the number of distinct controversies (# of distinct controversies). In Table II we report the average values across 25 pages of the noise/signal ratio, AvePr and Rand index and for # of distinct controversies we show the sum of all distinct controversies of those pages.

As a result of the experiment, we observe that the `text` model is able to retrieve more distinct controversies (80 vs 65). However, the `text` model is more noisy (0.25 vs 0.19 noise/signal ratio). The AvePr of `link` model is higher than that of the `text` model (0.91 vs 0.89). Finally, the `text` model is less blurred (0.74 vs 0.7 Rand index).

Moreover, in this experiment we observed that our technique finds new important controversies which were not captured in WPC. Specifically, using the `text` model CDA found 59

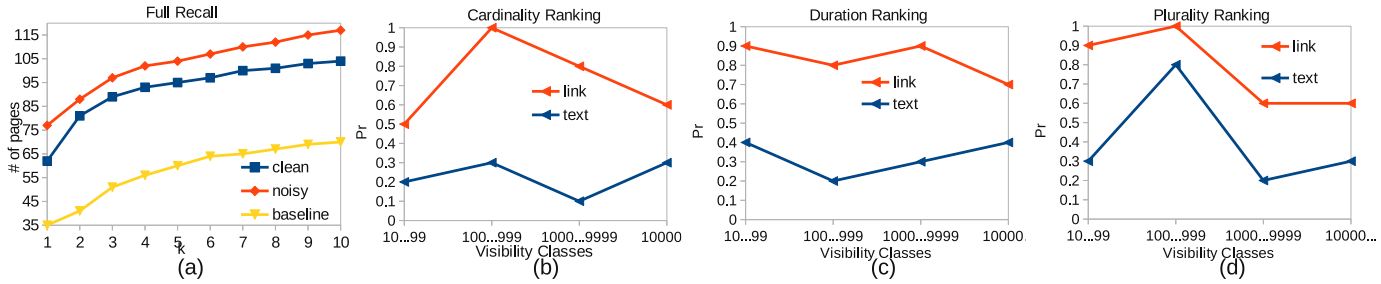

Fig. 3. (a) the recall on the entire set of WPC of clean/noisy controversies and the baseline; the cardinality (b), duration (c) and plurality (d) ranking for 4 levels of visibility of `link` and `text`.

page	WPC	New controversy
Chopin	nationality	birthday, photo, name
Avril Lavigne	song spelling	music genre, birthplace, religion
Bolzano	name spelling	language
Futurama	verb spelling	TV seasons, channel
Freddie Mercury	origin	name spelling, image

TABLE III
NEW PREVIOUSLY UNKNOWN CONTROVERSIES FOUND BY CDA

parameter	link model	text model
# of pages with edits (% of total)	2.4M (24%)	7.1M (71%)
total # of edits	16M	295M
average # of edits per page	1.6	29.5
max # of edits of a page	8031	111894

TABLE IV
THE ENGLISH WIKIPEDIA (A DUMP OF DEC 2013 WITH 10M PAGES) STATISTICS.

(236% of the number of WPC) new controversies which were not mentioned in WPC and 35 of them were ranked higher than the ones from WPC. Table III shows some of the examples of such controversies.

Comparing the cases when we use only substitutions (`text`, `link`) versus insertions, deletions plus substitutions (`text ins/del`, `link ins/del`) we observed a significant drop in all metrics and both data models, for the latter case. This confirms our assumption to use only substitutions to detect controversies (see Section III for details).

Finally, we observe that CDA (both `text` and `link`) significantly outperforms `baseline` which illustrates that the CDA technical contributions allow us to retrieve very clean results (0.19 vs 0.75 noise/signal ratio improvement), to have high precision (0.91 vs 0.34), to provide more homogeneous groups of edits (0.74 vs 0.3 Rand index) and, finally, to find more controversies (80 vs 17).

3) Accuracy of CDA at Large Scale: In this experiment we study the effectiveness of CDA at the scale of millions of Wikipedia pages. For that purpose we use the dataset of all the Wikipedia pages including meta pages like talks and templates (10M).

We conduct experiments with the `text` and `link` models and $k_{thrshld}$ and $r_{thrshld}$,$cutoff^{ctx}_{thrshld}$ parameters fixed to the values which maximize the recall in the experiment in Section V-C1, namely $k_{thrshld} = 2$ and `(8,.75)`, respectively.

The obtained edit and controversy statistics are shown in Table IV. The statistics indicate that the `text` model provides almost 3 times more edits which results in a large portion of pages (7.1M vs 2.4M) which can be potentially analyzed by CDA.

During our experiments, we observed that pages which are more popular (have a large number of revisions) are likely to be attacked by vandals. To address this fact, we introduce the notion of visibility which is determined by the number of revisions of a page. In the next experiments, we study the precision of CDA for each class of visibility individually. For that purpose, we bin the pages based on the number of revisions on exponential scale, i.e. 10-99, 100-999, 1000-9999, and 10000-....

As a ground truth in this experiment, we use the feedback from users, i.e. for every group of edits we ask the users to specify whether it is a controversy or not.

According to the definition of controversy we discussed in Section III, there are three properties which characterize controversial content, namely the number of edits of a controversy, the duration of edits and the number of distinct authors (plurality). In the precision experiment, we aim at testing all these three properties and therefore we experiment with the following ranking functions: the cardinality of controversy (`cardinality`), the duration of controversy (`duration`), the plurality of controversy (`plurality`). In addition, in order to see whether our ranking functions are truly correlated with controversies we also use a random ranking (`random`). A ranking function f is applied as follows: first we rank the controversies within a page using f and pick the top1 controversy, second we rank pages within the same visibility class using f of top1 controversies of pages.

In Figure 3(b,c,d) we report the precision which is computed on the set of Wikipedia pages for top 10 retrieved controversies per page. Moreover, we do it for 4 classes of visibility individually, i.e. only pages from the same visibility class are ranked. Three ranking functions are used: the `cardinality`

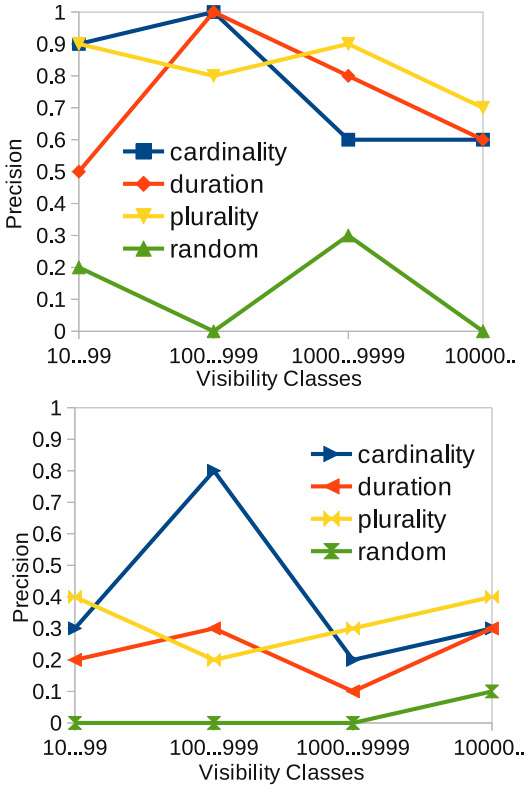

Fig. 4. `Cardinality`, `duration` and `plurality` for the `link` (a) and `text` (b) models.

(b), `duration` (c) and `plurality` (d).

The results show that the `link` model results in considerably higher precision than using the `text` one, for each of `cardinality`, `duration` and `plurality`.

In the next experiment, using the same setting we study which ranking function leads to higher precision and also compare the proposed ranking functions with a random ordering. The results for the `link` and `text` models are shown in Figure 4. Both for the `link` and `text` models, the proposed ranking functions are able to detect controversial content because they lead to significantly higher precision in comparison with the random ordering. Interestingly, we observe a decline in precision for the `link` data model with an increasing level of visibility (Figure 4(a)).

4) Ranking pages with Edits: In this experiment we use edits (i.e. substitutions of `text` and `link` tokens) to rank pages according to the level of their controversiality. The main goal of this experiment is to show that edits are highly correlated with the known controversial page level metrics (the revision number, the number of unique authors and the approaches from [9]).

The effectiveness at the page level is measured by precision, recall and Normalized Discounted Cumulative Gain (NDCG) (see [24] for details) of retrieved controversial pages. As a dataset for that experiment we use the full dump ($10M$ pages). As the ground truth of the level of controversy of a page we use ATC, i.e. the number of controversy templates in the revision history of a page. For the experimental details see [9].

As competing algorithms we use the number of revisions of a page (*revN*), the number of unique contributors of a page (*uniqContr*), the age-aware CR Average (*ageAvg*) and the age-aware CR Product (*ageProd*) from [9] and the number of edits from the text (*text*) and link (*link*) data models. We also compared with the less advanced techniques from [9] (the basic model, CR Average, CR Product, the age-aware basic model) but for the sake of presentation we report only the results of age-aware CR Average (*ageAvg*) and age-aware CR Product which showed the highest effectiveness.

For the English Wikipedia dump (more than 7 TB) the computation of the above statistics took around 1 month on one 4GHz CPU machine with 4GB memory running Ubuntu 12.04. The main time-consuming part is to compute differences between revisions and the iterative computation of the statistics from [9] (*ageAvg* and *ageProd*). Note that in this experiment we don't compute controversies and just use link and text edits (*text* and *link*) therefore in contrast to [9] our approach doesn't introduce any overhead with respect to the parsing and revision difference computation.

The results for the varying number of retrieved pages, $k = 100$ to $k = 10M$ on exponential scale, are shown in Figure 5. The measured accuracy metrics indicate that both the `text` and `link` edits are able to detect controversial pages with a similar or higher precision, recall and NDCG. In turn, it means that substitutions are highly correlated with the known controversial page level metrics (the revision number, the number of unique authors and the approaches from [9]).

D. Experiment Summary

The main experimental result is that the proposed techniques (the usage of substitutions, edit contexts, edit clustering, data models, and so on) provide a way to efficiently retrieve individual controversies with high accuracy.

In the recall experiment, CDA shows a higher recall than `baseline` almost with any set of parameters. The difference is higher when the author filter and `text` model are used. In the full recall experiment, CDA is able to retrieve a large portion of WPC which is more than 20% (53 more known controversies are found) improvement over `baseline`.

In the accuracy experiments, CDA (both `text` and `link`) outperforms `baseline`. More specifically, we have 0.19 vs 0.75 noise/signal ratio improvement, 0.91 vs 0.34 precision gain, 0.74 vs 0.3 Rand index increase and, finally, CDA finds more distinct controversies (80 vs 17).

Regarding the CDA parameter tuning, one of the main takeaways is that the `text` model is able to retrieve more WPC than the `link` one. However, the `text` model is more noisy and at a large scale it shows a much lower precision. This behavior can be explained by the fact that `text` has a large number of edits (it is more easy to update a word than a link) but at the same time it is more likely that `text` edits are more noisy or vandalised. In our experiments, we clearly observe that eliminating edits which are not repeated by at least two different authors (i.e. using the $k_{thrshld} = 2$)

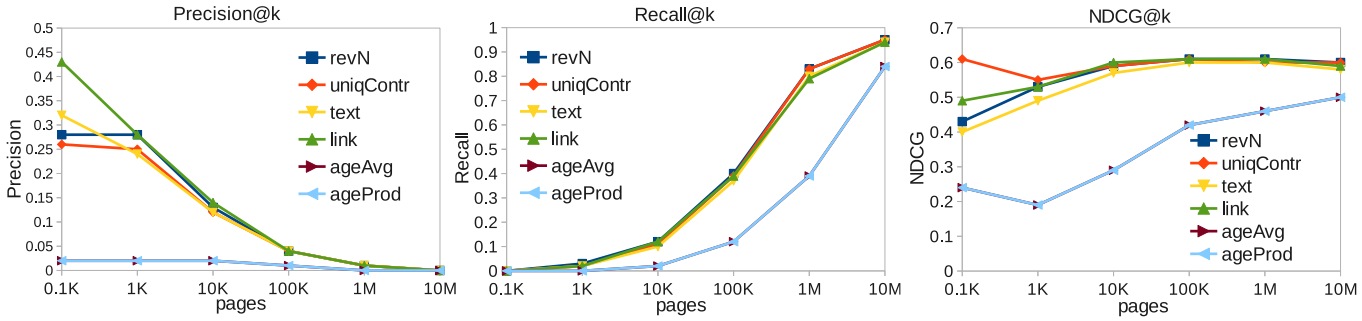

Fig. 5. Precision (a), Recall (b) and NDCG (c) of the approaches based on # of revisions (revN), # of unique authors (uniqContr), ageAvg and ageProd and the statistics based on the `text` and `link` edits on the entire English Wikipedia dump of Dec 2013

significantly improves the quality of results by reducing the amount of noisy edits.

Regarding the properties of controversies, we found that `cardinality`, `duration`, `plurality` are good indicators of content controversiality which experimentally confirms the semantic definition of a controversy as "a prolonged dispute by a number of different people on the same subject".

Finally, we show that both `text` and `link` edits themselves can serve as statistics to rank controversial pages and they show the same or higher accuracy with respect to the well-known statistics of the number of revisions, the number of unique authors as well as the state-of-the-art approaches [9].

VI. Conclusion

We have studied the problem of controversy detection in collaboratively-edited content, and more specifically in Wikipedia. In contrast to previous works in controversy detection in Wikipedia that studied the problem at the page level, we have developed an algorithm that considers the individual edits and can accurately identify not only the exact controversial content within a page, but also what the controversy is about and where it is located. Furthermore, apart from analyzing the text as is traditionally done in similar approaches, we have additionally developed a novel model that is based on links and we have shown that it generates more semantically meaningful controversies than the text-based model. Our extensive experimental evaluation showed that the proposed techniques can retrieve individual controversies with high precision and recall and outperform the existing approaches. Our future work includes extending the fine grained controversy detection to machine learning techniques (e.g. k-nearest neighbors clustering), sentiment analysis and natural language processing.

References

[1] J. Giles, "Internet encyclopaedias go head to head," *Nature*, vol. 438, no. 7070, pp. 900–901, Dec. 2005.

[2] S.-C. Chin, W. N. Street, P. Srinivasan, and D. Eichmann, "Detecting wikipedia vandalism with active learning and statistical language models," in *WICOW*, 2010, pp. 3–10.

[3] W. Gunningham and B. Leuf, *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.

[4] R. Sumi, T. Yasseri, A. Rung, A. Kornai, and J. Kertesz, "Edit wars in wikipedia," in *SocialCom*, 2011.

[5] S. Chaudhuri, "Editors Won't Let It Be When It Comes to 'the' or 'The'," *The Wall Street Journal*, p. A1, Oct 2012.

[6] M. Potthast, B. Stein, and R. Gerling, "Automatic vandalism detection in wikipedia," in *Advances in Information Retrieval*. Springer, 2008, pp. 663–668.

[7] K. Smets, B. Goethals, and B. Verdonk, "Automatic vandalism detection in wikipedia: Towards a machine learning approach," in *AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, 2008, pp. 43–48.

[8] A. Kittur, B. Suh, B. A. Pendleton, and E. H. Chi, "He says, she says: conflict and coordination in wikipedia," in *CHI*, 2007, pp. 453–462.

[9] B.-Q. Vuong, E.-P. Lim, A. Sun, M.-T. Le, and H. W. Lauw, "On ranking controversies in wikipedia: models and evaluation," in *WSDM*, 2008.

[10] S. Dori-Hacohen and J. Allan, "Detecting controversy on the web," in *CIKM*, 2013, pp. 1845–1848.

[11] G. Druck, G. Miklau, and A. McCallum, "Learning to Predict the Quality of Contributions to Wikipedia," *AAAI*, vol. 8, pp. 983–1001, 2008.

[12] B. T. Adler and L. de Alfaro, "A content-driven reputation system for the wikipedia," in *WWW*, 2007, pp. 261–270.

[13] B. T. Adler, K. Chatterjee, L. de Alfaro, M. Faella, I. Pye, and V. Raman, "Assigning trust to wikipedia content," in *WikiSym*, 2008, pp. 26:1–26:12.

[14] T. Yasseri, R. Sumi, A. Rung, A. Kornai, and J. Kertéz, "Dynamics of conflicts in wikipedia," *PLoS ONE*, 2012.

[15] H. Sepehri Rad and D. Barbosa, "Towards identifying arguments in wikipedia pages," in *WWW*, 2011, pp. 117–118.

[16] F. B. Viégas, M. Wattenberg, and K. Dave, "Studying cooperation and conflict between authors with history flow visualizations," in *CHI*, 2004, pp. 575–582.

[17] U. Brandes and J. Lerner, "Visual analysis of controversy in user-generated encyclopedias," *Information Visualization*, vol. 7, no. 1, pp. 34–48, 2008.

[18] T. Yasseri, A. Spoerri, M. Graham, and J. Kertesz, "The most controversial topics in wikipedia: A multilingual and geographical analysis," in *Global Wikipedia: International and Cross-Cultural Issues in Online Collaboration*, 2014.

[19] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 241–272, 1901.

[20] E. W. Myers, "An o(nd) difference algorithm and its variations," *Algorithmica*, vol. 1, pp. 251–266, 1986.

[21] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, Jun. 1998.

[22] D. Hawking, N. Craswell, P. Bailey, and K. Griffihs, "Measuring search engine quality," *Inf. Retr.*, vol. 4, no. 1, pp. 33–59, Apr. 2001.

[23] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. pp. 846–850, 1971.

[24] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *SIGIR*, 2000, pp. 41–48.