Edit-History Vis: An Interactive Visual Exploration and Analysis on Wikipedia Edit History

Yuhan Guo[1,2] * Qin Han[2] † Yuke Lou[2] ‡ Yiming Wang[2] § Can Liu[1] ¶ Xiaoru Yuan[1,3] ‖

1) Key Laboratory of Machine Perception (Ministry of Education), and School of Intelligence Science and Technology, Peking University, China
2) School of Electronic Engineering and Computer Science, Peking University, China
3) National Engineering Laboratory for Big Data Analysis and Application, Peking University, China

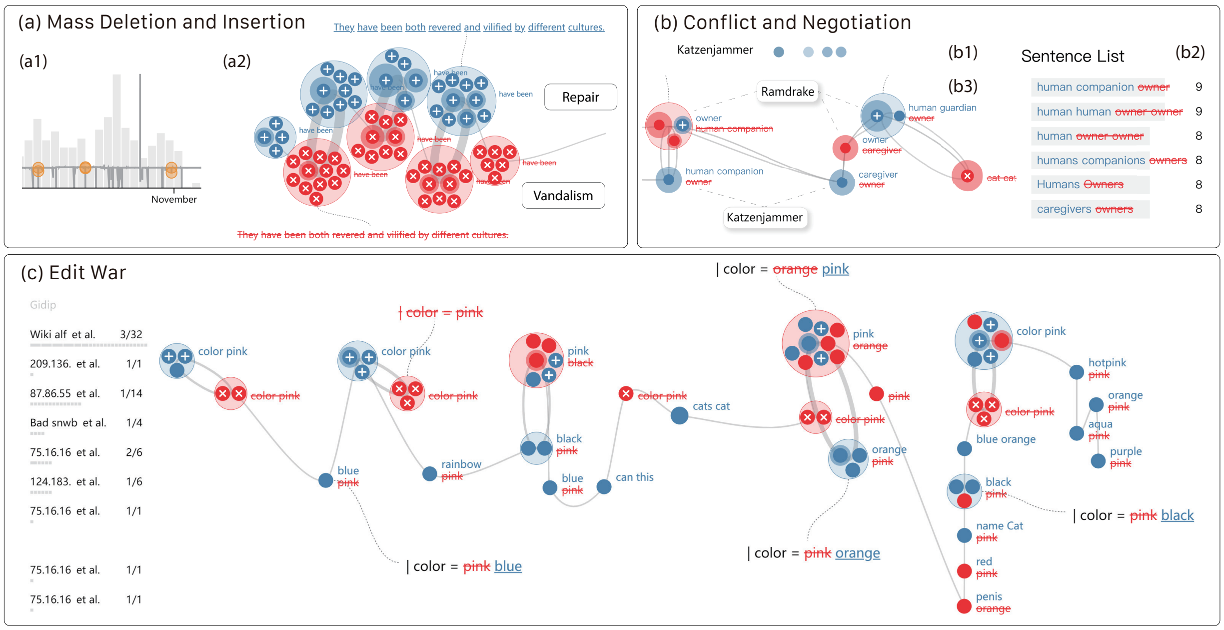

Figure 1: Exploration on Wikipedia article "Cat" using Edit-History Vis. (a) We observed sharp decreases and increases in the page size (a1). Vandals removed the content repeatedly, and other editors repaired the vandalism (a2). (b) Katzenjammer and Ramdrake conflicted on the relationship between human and cat, namely owner or caregiver (b3). Katzenjammer changed "owners" into "caregivers" and "human companions" (b2). Later, they reached a compromise, using "human guardian" (b3), and Katzenjammer stopped to change the "owner" (b1). (c) An edit war occurred over the template color of the infobox on the page.

Abstract

We propose Edit-History Vis, a visual analytics system designed to facilitate interactive exploration on Wikipedia edit history at a fine-grained level. The examination of detailed changes in Wikipedia articles is crucial for understanding how authors' perspectives vary and conflict during the collaborative editing process. However, it is challenging to reveal the details while preserving the heterogeneous attributes of revisions, namely the time, content, and editor. The Edit-History Vis system integrates editor and textual changes of revisions by utilizing a force-directed revision graph that groups revisions based on standpoints. Through this revision graph, users can identify and analyze editing events such as edit wars, vandalism, repair, and normal updates. The effectiveness of the system in analyzing the edit history is validated through a qualitative comparison with prior work and a quantitative rating from a user study.

Index Terms: Human-centered computing—Visualization—Visual Analytics

1 Introduction

Wikipedia, the largest online encyclopedia, is targeted to enable the free acquisition and spread of comprehensive knowledge. Built on a wiki-based editing system, Wikipedia allows open collaboration, which means anyone, either domain experts or the general public, can contribute to the pages by editing them directly. Therefore, a Wikipedia article may have amounts of revisions, which are documented in the Wikipedia revision history[1]. These recorded revisions are valuable for both researchers and the general public. For researchers, they reflect Wikipedia's efficiency and quality as an open knowledge platform, as well as the underlying mechanisms of online collaboration. For general users, it is interesting to explore how the articles form, and awareness of the collaborative process promotes critical thinking on the shown content. However, the revision data is complex because of heterogeneous attributes, namely the edit content, the edit time, and the editor, posing a challenge for comprehending the edit history.

*e-mail:yuhan.guo@pku.edu.cn
†e-mail:hanqin@pku.edu.cn
‡e-mail:louyuke@pku.edu.cn
§e-mail:wym12416@pku.edu.cn
¶e-mail:can.liu@pku.edu.cn
‖e-mail:xiaoru.yuan@pku.edu.cn (corresponding author)

[1] https://en.wikipedia.org/wiki/Help:Page_history

Many visual tools have been developed to facilitate understanding of the complicated data. A majority of these tools focused on one or two of the three attributes. For instance, from the editor's perspective, Flöck et al. [10] proposed WhoVIS, which used a time-variant graph to show the dynamics of conflicts between editors but did not show which content is controversial. Contropedia [6] studied the edit history from a content perspective, computing the controversy of the content and visualizing its trend over time, but did not support analyzing the interaction between editors. From our perspective, the content evolution and the editor interaction are both indispensable for understanding the dynamic and collaborative writing process of Wikipedia articles, including how the content forms, how editors cooperate and conflict, what standpoints exist, and how the standpoints evolve. However, there is no existing work that is able to integrate the three dimensions (the content, the time, and the editor) well while preserving the detailed information for analysis.

To fill this gap, in this work, we proposed Edit-History Vis, a visual analytics system that allows visual exploration and analysis of the detailed development of a Wikipedia article. Edit-History Vis is featured with a revision graph view, which is aimed to visualize the dynamics of standpoints in an editing event. The revision graph encodes each detailed edit as a node whose horizontal position is fixed according to the edit time and links the related edits to illustrate the evolution of the text content. We used a force-directed model to lay out the nodes, which integrates the editor and text changes to indicate standpoints, as well as aggregation and annotation methods so that users can grasp the editors' opinions and interactions by reading the graph. To support a comprehensive exploration of the edit history, we combined the revision graph, an article view, and an overview, allowing users to detect and analyze editing events (e.g., vandalism, conflict, and progress). We introduce our discoveries through analysis of three cases via the Edit-History Vis system. We evaluate our work by a qualitative comparison with previous work as well as a user study. The results show that Edit-History Vis is capable of assisting users in understanding the editing process. To summarize, the main contributions of this paper include:

- A novel visual design for the editing process of fine-grained Wikipedia article content which shows the dynamics of standpoints and interaction between editors.
- Edit-History Vis, a visual analytics system that enables users to interactively explore the edit history, as well as analyze the editing patterns and events comprehensively and detailedly.

2 Related Work

In this section, we review the existing visual tools for exploring the edit history of a Wikipedia article and methods for detecting Wikipedia editing events. We also review parallel-text visualizations.

2.1 Visualization of Wikipedia Edit History

Wikipedia documents all history revisions but does not provide an easy way to browse and understand them. Many research efforts have been devoted to shedding light on the complicated underlying process of editing a Wikipedia article, from the perspective of evolution patterns and editors' contributions.

Visualizing the evolution of Wikipedia articles. In the early days of online mass collaboration platforms, the quality and cooperation mechanisms are not known. To get an overview of the complex edit history, Viégas et al. [29] designed the *history flow* visualization, which represents each revision by a vertical "revision line" and links corresponding segments with shaded connections. The shape of the flow-like visualization indicates the patterns of conflict and cooperation between authors, such as vandalism and negotiation. Bach et al. [5] took the similarity between revisions into consideration and designed *time curves*, a general method for visualizing the evolution patterns of temporal data. Apart from the overview methods, there are visual tools for viewing the detailed evolution process of the article content. For example, Wikireplay [2], which was developed by the Wikipedia community, uses animation and transformation to show the edits happening in the article.

Visualizing the contribution of editors. The authorship of Wikipedia articles is important for evaluating the authenticity and deciding copyright, yet hard to judge due to large numbers of editors overriding each other's contributions. WikiDashboard [25] uses a user dashboard to visualize users' contributions over time, but it does not provide details of the edits. Adler et al. [1–3] developed WikiTrust to measure the authorship and assign trust to passages in Wikipedia articles, which is visualized by color shades. Flöck et al. improved the measuring algorithm with a tree model and proposed the WikiWho algorithm [9], which calculates the authorship of tokens in Wikipedia articles. Based on the algorithm, WhoCOLOR [11] was created, coloring each word according to its author.

These tools help users understand the overall revision history of Wikipedia articles, but do not aim to support the analysis of specific content changes and editing events. In Edit-History Vis, users have easy access to the details of the evolution of sentences and the interaction between editors.

2.2 Wikipedia Event Detection and Analysis

The occurrence of social events often causes outbreaking editor and user activity in multiple relevant Wikipedia articles, which can be detected by measuring edits [8,24,28] and page views [27] of related Wikipedia articles combined with information from other social media to detect events [18]. Sumi et al. [26] designed an algorithm for detecting edit wars in Wikipedia by carefully comparing the revisions. Chhabra et al. [7] studied the conflicts in Wikipedia articles extracted in the form of edit war sequences. Despite the increase in the accuracy of automatic event detection algorithms, these methods do not tell the story of the development and editors' interactions in the events.

Some studies develop visualization methods for analyzing conflicts. Kittur et al. [15] constructed the Revert Graph to show the conflicts between groups of users. Flöck et al. [10] also built over-time editor networks based on the WikiWho algorithm. Aiming to provide a visual tool for the real-time analysis of controversies, Contropedia [6] extracted controversial objects from both the revision history and the discussion pages and provided a layer view as well as a dashboard view for detecting and understanding controversies. An edit view listing the changed texts related to the controversial objects is also provided. However, the number of editors and edits regarding a controversial element is so huge that it is still hard to get a thorough comprehension of the conflict by reading the separated views and the long list. In this work, We designed a revision graph to present the relevant edits in a legible manner, and the "what", "who", and "when" tasks that Contropedia is concerned with can be solved directly in this view.

2.3 Visualization of Parallel Texts

Parallel text visualizations are commonly used in the digital humanities by close reading and distant reading methods to analyze different versions of documents or re-used texts [16, 23, 30]. A common approach for visualizing parallel text is juxtaposition, using side-by-side views [29] to show the alignment of the texts [4, 14]. The diff page in Wikipedia [3] provides a close view aligning paragraphs of two revisions, using color shades to encode changed texts. Besides a zoomable distant view, TransVis [4] provided a close view juxtaposing the aligned texts and coloring the corresponding words. The side-by-side layout is limited to the screen size, supporting showing about ten texts on average [30]. Variant graphs merge equal or similar tokens to reduce redundant information and are

[2] http://cosmiclattes.github.io/wikireplay/player.html
[3] https://en.wikipedia.org/wiki/Help:Diff

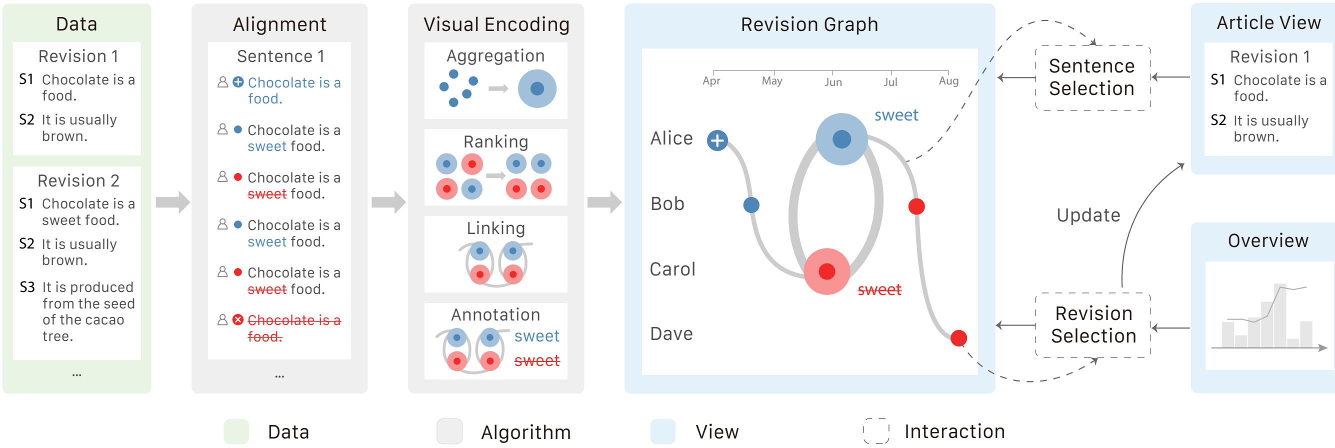

Figure 2: The pipeline of Edit-History Vis. First, we collect the revision data and align the consecutive revisions to get the changes in each sentence. We encode the evolution of the sentence on a revision graph by aggregating and ranking the revisions according to standpoints, as well as linking the relevant revisions and annotating the changed texts. The visual analytics system is developed for interactive exploration of the Wikipedia edit history, with revision graph as the major view, along with an article view and an overview.

capable of visualizing about twenty parallel sentences at the token level [13, 20]. Silvia et al. [22] applied the storyline metaphor to the variant graph with a force-directed layout algorithm to facilitate the comprehension of text variations.

However, the revisions of a Wikipedia article differ from the classical variant texts in that they are in a temporal order where each revision is the successor of the previous revision. In this work, we focus on the visualization of aligned sentences. The traditional parallel text visualization methods do not support showing the temporal development of texts with numerous edits and editors. Therefore, we proposed the revision graph, using aggregated nodes to represent revision clusters to save space.

3 Design Rationale

In this section, we describe the Wikipedia data and analyze the tasks. We also provide an overview on the Edit-History Vis system.

3.1 Data Description

The Wikipedia system documents all history revisions of the articles and provides a history page [4] for each article, allowing users to browse the previous edits. MediaWiki provides an API for getting the revisions of given articles in the wiki format [5].

Revisions' attributes. The *edit time*, the *edit content*, and the *editor* are the essential attributes of the revision data. The three heterogeneous dimensions result in the complexity of the editing history, and integrating them in a visualization view is the major challenge we face.

Sentence-level evolution. The edit history can be decomposed into sentence-level evolution. The revisions of each sentence are a subsequence of the article revisions and, similarly, consist of the *edit time*, the *edit content*, and the *editor* dimensions. The *edit content* can be classified into four types, namely inserting a new sentence, deleting an old sentence, moving a sentence to another position, and modifying some words of the old sentence.

Editors' standpoints and relationships. Editors may have different standpoints on the topic of the article, which are embedded in the texts they edit. For instance, the reversion and restoration edits indicate the conflict between editors. In our context, the relationship between two editors is defined by whether they have edited the same sentence, and the standpoint is derived from the editing behavior. Similar edits of sentences indicate similar standpoints.

[4]https://en.wikipedia.org/wiki/Help:Page_history
[5]https://www.mediawiki.org/wiki/API:Revisions

3.2 Task Analysis

Drawing from previous literature and our goal of analyzing standpoints, we identified the tasks. In addition to detecting conflicts, as addressed by prior tools, we aim to enable a comprehensive analysis (T3 and T4) based on the details of the sentence-level evolution (T2), which is not supported by prior work.

- **T1: Getting an overview of the editing history.** Due to the scale and complexity of the revision data, an overview of the edit history is necessary for enabling users to perceive the editing process in an intuitive and straightforward manner, as well as facilitating the observation of interesting subsets of data [21]. The distribution of edits over time outlines the editing process and indicates possible events. The overview of each author's edits allows an easy evaluation on the social activities and patterns around the article [25].
- **T2: Understanding the sentence-level editing process.** The editing process of the sentence helps understand the evolution of the Wikipedia article in a detailed and comprehensive manner, combining the *edit time*, the *edit content*, and the *editor* dimensions. Moreover, the sentence-level edit history is essential for understanding what content is disputed [6]. The understanding of the evolution of a sentence is the basis for analyzing controversial objects. Specifically, users need to know the time of the edits on the sentence, the major participants, and the main standpoints.
- **T3: Detecting and analyzing controversial content.** The reliability of Wikipedia content has received long debates. The detection of controversial content helps users increase their awareness of which piece of content might be unreliable [1]. How the controversies evolve mirrors the development of societal controversies. [6] In prior work, the evolution of controversial content is often represented by the temporal distribution of the inserts and deletes of specific elements [11]. We would like to support the analysis of comprehensive standpoints beyond the changes of single words, such as conflicting political, cultural, and religious stances, which are reflected in the controversies.
- **T4: Observing and analyzing editing events.** Observation of editing events, especially vandalism and edit wars, helps Wikipedia administrators to manage the platform and researchers to study the collaboration patterns of the Wikipedia community [29]. Beyond observation, we dig into the details of the events, namely, the cause and development of the events, along with the behavior of editors and the relationship between them.

3.3 System Overview

We proposed Edit-History Vis, a visual analytics system supporting interactive exploration on Wikipedia edit history. Figure 2 illustrates the pipeline of Edit-History Vis. We performed sentence-level alignment of the revisions to get the evolution process. A controversial sentence involves many edits and editors with various standpoints, and it is laborious to read through the changes. We represented the development of a sentence by aggregating and ranking the edits according to time and *standpoints*, linking relevant revisions, and annotating the major changes. Based on these visual encoding methods, we implemented the revision graph, which visualizes the edits regarding chosen words, sentences, or revisions on demand and supports an in-depth analysis of these edits (**T2**, **T3**, **T4**). We also provide an overview (**T1**, **T4**) and an article view (**T3**) for users to detect and select interesting sentences, revisions, and events.

4 Revision Graph

This section introduces the design of the revision graph, namely, data preprocessing and visual encoding. In the revision graph, the nodes represent the revisions, and the links represent the editing relationships between revisions. We set the sentence as the basic unit to generate the relationships and build the revision graph for the following reasons. (1) Sentences are the smallest semantically complete unit and editors probably correlate if they have edited the same sentence. (2) An editing event usually involves debates or updates on several sentences. Therefore, we first clarify how to construct the revision graph for the evolution process of a single sentence.

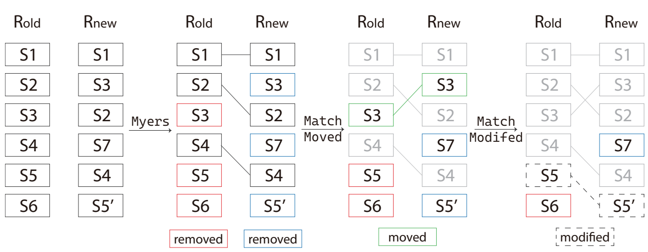

Figure 3: Sentence-level comparison algorithm.

4.1 Data Preprocessing

We compare every revision from the MediaWiki API [6] with its successor to align the sentences so that we can get the revisions of each sentence. For each edit of the sentence, we compare the old text and the new one to get the token-level changes. We conduct a rough detection of vandalism edits, as well as calculate sentence controversy and token controversy.

Sentence-level comparison. We split revisions (denoted as $\{r_0, r_1, r_2, \cdots\}$) by sentence and compare each revision r_i with the previous one r_{i-1} to get the changes in sentences. The edit of a sentence is classified into four categories: *inserted*, *deleted*, *moved*, and *modified*. As is illustrated in Figure 3, first, we run the Myers difference algorithm [17] to align the sentences. The Myers algorithm, which does not recognize the moved sentences and the modified sentences, returns the list of the unchanged sentences, the inserted sentences, and the removed sentences. Then, we match inserted and removed sentences to get the *moved* sentences (S3 in Figure 3). Finally, we traverse the remaining sentences in inserted and removed ones to get the *modified* sentences. Specifically, we use a sentence-transformer [19] to calculate the similarity between sentences. If the similarity between a deleted sentence and an inserted sentence exceeds the threshold (in the current implementation, we choose 0.9 based on manual checking of the computed results), we set the latter as the modification result of the former (S5, S5' in Figure 3). The rest of $S_{inserted}$ and $S_{removed}$ are the *inserted* and the *deleted* sentences.

[6]https://www.mediawiki.org/wiki/API:Revisions

Token-level comparison. For the *modified* sentences, we compare its old version with the new one to get the changes in tokens. There are three types of changes of tokens: *inserted*, *deleted*, and *moved*. Similar to the sentence-level comparison, we use the Myers difference algorithm [17] to get the unchanged tokens, the inserted tokens, and the removed tokens. Then, we compare the inserted tokens and the removed tokens to get the *moved* tokens.

Vandalism detection. We conduct a rough detection of vandalism edits to filter out the mass deletion and insertion, which would otherwise bring noise when measuring controversy. Let μ denotes the average page size, and σ denotes the standard deviation of the page size. For each revision r_i, let $size(r_i)$ denote its page size. If $|size(r_i) - \mu| > \lambda \cdot \sigma$, where λ is a threshold, we consider the revision as vandalism. We set $\lambda = 3$, which is the common setting for outlier detection.

Sentence controversy and token controversy. The sentence and token controversy are denoted as the number of changes in a sentence and token, respectively. As vandalism does not influence the controversy, we filtered out the vandalism edits when calculating sentence and token controversy.

4.2 Graph Construction and Visualization

In order to display the revisions in a readable and accessible manner, we proposed the revision graph, which encodes temporal, textual, and author information using position channels. Each revision of the sentence is represented with a point. We lay out the points through four visual encoding modules: aggregation, ranking, linking, and annotation. To reduce the overlap caused by the huge number of revisions in a controversial sentence, we aggregate the adjacent points. A force-directed algorithm is applied to optimize the vertical ranking of the revisions so that the revisions with the same *standpoints* are close in the vertical direction. We link the revisions with curves to augment the temporal order, along with the repeated modification, which indicates a possible edit war. The aggregated revisions are annotated with the keywords of the revision.

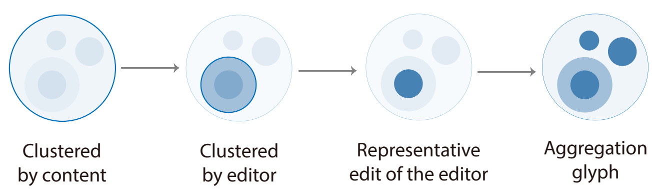

Figure 4: Glyph design for aggregated nodes. The nodes aggregated by the text are packed in a circle. Within the packing circle, nodes are grouped according to the editor.

Aggregation. The revision points are usually unevenly distributed in time. Sometimes a large number of edits break out, resulting in severe overlap of points, especially when there is an edit war. Therefore, we aggregate the points that are close in time and have the same editing result. The aggregated revisions are represented by glyphs.

- *Aggregation algorithm.* We formalize the details of the aggregation method in Algorithm 1. The clustering radius d is determined by the window size. In the revision graph, we set the clustering radius as the width of the view divided by the max radius of the point. Initially, each revision is a cluster. If the distance between a revision outside a cluster and another revision in the cluster is less than the clustering radius, we merge the former revision into the cluster. We scan the revisions to get the aggregation.

- *Glyph design.* We use circles to represent the revisions. For a single revision, the size of the circle encodes the byte changes and the horizontal coordinates represent the time, which is consistent with the timeline coordinates in the overview (Figure 6 (b)). The type of edit is encoded with both the color and the central icon. The green ● represents the *move* operation. The red color represents the increase of bytes, and the blue color represents the decrease in bytes. The *inserted* is represented with an additional plus icon ⊕, and the *deleted* with a cross icon ⊗. The circles colored red ●, blue ● without the central icon are the *modified* type. As is illustrated in Figure 4, for the aggregated revisions, we group them by the editors. Each group is represented with an inner circle encoding the representative revision and a halo indicating the number of revisions. The groups are packed within a larger circle to represent the cluster. The horizontal coordinates of the clusters represent the average of the edit time of the revisions in it.

Algorithm 1 Aggregation: aggregating revisions with similar edit time and the same text content

Parameters: clustering radius d

Input: sentence revisions $R = \{r_1, r_2, \cdots\}$, $x(r_i)$ denotes the x coordinate of r_i, $t(r_i)$ denotes the text content of r_i

Output: aggregated nodes $N = \{n_1, n_2, \cdots\}$, $x(n_i)$ denotes the x coordinate of n_i, $t(n_i)$ denotes the text content of n_i

```
Initialize: nodes N = {n_1, n_2, ...}, where n_i = r_i, x(n_i) = x(r_i), t(n_i) = t(r_i)
while true do
    for each node n_i ∈ N do
        for each node n_j ∈ N, j ≠ i do
            if t(n_i) = t(n_j) and |x(n_i) − x(n_j)| < d then
                n_i ← n_i ∪ n_j, delete n_j
        update: x(n_i) ← average{x(r), r ∈ n_i}
    if nothing updated then
        return N
```

Ranking. We encode the *standpoints* into the vertical dimension. Generally, A standpoint is a specific manner through which a person thinks about something. In our context, *standpoint* is defined as follows: (**D1**) *standpoint* is presented as similar editing behavior; (**D2**) an editor usually has a consistent *standpoint*. To visualize the dynamics of *standpoints*, we use a force-directed algorithm to optimize the vertical coordinates so that the vertical position can indicate the *standpoint*. We set the forces according to the similarity of editing behavior and the consistency of editors' opinions. The Optimization goals (**G**) and constraints are (**C**) as follows:

- **G1.** The revisions with similar editing behavior should be close in the vertical direction (**D1**). We represent the editing behavior by the editing result and minimize the total distance d_{text} between revisions with the same revised text.
$$d_{text} = \sum_{t \in T} \sum_{\substack{r_i, r_j \in R_t \\ r_i \neq r_j}} |y(r_i) - y(r_j)|,$$
where T denotes the set of all versions of the sentence, R_t denotes the set of the sentence revisions whose editing result equals the sentence version t, and $y(r)$ denotes the vertical coordinate of the node that the revision r belongs to.
- **G2.** The revisions edited by the same editors should be close in the vertical direction (**D2**). We minimize the total distance d_{editor} between revisions with the same editors.
$$d_{editor} = \sum_{e \in E} \sum_{\substack{r_i, r_j \in R_e \\ r_i \neq r_j}} |y(r_i) - y(r_j)|,$$
where E denotes the set of editors that have revised the sentence, and R_e denotes the set of revisions by editor e.
- **G3.** The revisions with more common *standpoints* should be placed higher. We rank the sentence versions T according to the occurrence number of each version, and $rank(t)$ denotes the rank of version t. We minimize the total distance d_{rank} between the revision's vertical coordinate and the expected rank.
$$d_{rank} = \sum_{r \in R} |y(r) - rank(t(r)) \cdot h|,$$
where R denotes the revisions of the sentence, $t(r)$ denotes the revised text of revision r, and h denotes the row height.
- **G4.** The vertical span between each node and its successor should be small so that the curve will not be messy. We minimize the total vertical distance d_{span}.
$$d_{span} = \sum_{i=2}^{M} |y_n(n_i) - y_n(n_{i-1})|,$$
where n_i denotes the t-th node and $y_n(n)$ denotes the vertical coordinate.
- **C1.** The overlap of the nodes should be avoided.
$$\forall n_i, n_j \in N, |c_i - c_j| > rad_i + rad_j,$$
where N denotes the set of nodes, c_i denotes the center point of node n_i, and rad_i denotes the radius of node n_i.
- **C2.** The nodes should be within the boundary of the minimal area that contains all revisions without overlap.
$$\forall n \in N, y_{min} < y(n) < y_{max},$$
where y_{min} and y_{max} denote the lower and upper bound.

We set the force accordingly and use the d3 force-directed algorithm to get the optimized ranking of the revisions. Since we only optimize the vertical dimension, we fix the horizontal coordinates during the simulation and customize the update of vertical velocity and position according to the force.

- **F1.** *Text gravitation*: gravitation between every two revisions with the same editing result (**G1**).
- **F2.** *Editor gravitation*: gravitation between every two revisions by the same editor (**G2**).
- **F3.** *Gravity*: the weight of the versions (**G3**).
- **F4.** *Adjacent gravitation*: gravitation between each adjacent pair of points (**G4**).
- **F5.** *Collision force*: repulsion between points (**C1**).
- **F6.** *Boundary*: limitation on the range of the coordinates of the revisions (**C2**).

Linking. The basic idea of the linking method is to draw a curve that connects every revision of a sentence in the temporal order. If there are repeated modifications between two clusters, messy circle-shaped curves will appear, indicating conflicts between camps. To reduce clutter, we aggregate the curves between two conflicting clusters. First, we enumerate the clusters and record all pairs of clusters between which repeated modifications exist. For each pair, we draw a circular curve between the two clusters and the width of the curve represents the number of repeats. For example, in Figure 8, the largest circle indicates that a group of editors kept deleting "Indian", while another group kept inserting "Indian". Then, we connect the remaining nodes and the circles by the temporal order with a Bézier curve. The control points are calculated by the Catmull-Rom interpolation [12].

Annotation. We label the glyphs with the key tokens which are changed. We also annotate the representative editors.

- *Glyph labels.* For each edit on a sentence, we use the token-level comparison method to get the changed tokens. For each type of change, we select the two tokens with the highest word frequency (the tokens with no real meaning are excluded) and use color and text decoration to encode the type.
- *Editor labels.* We divide the vertical space into several bins whose height equals the maximum diameter of the revision glyphs. We select the editor with the most edits as the representative.

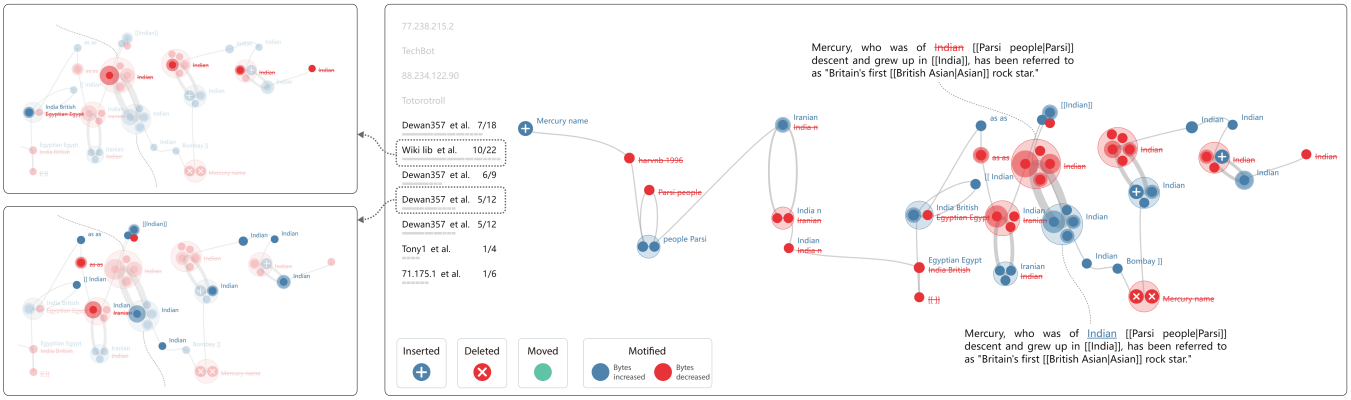

Figure 5: The visual design of the revision graph, taking an edit war in "Freddie Mercury" as an example. Revisions are represented by nodes, which are aggregated according to the text content and connected in temporal order. Main editors and changes are annotated. The revision graph shows that one camp of editors, represented by Dewan357, insisted on adding "Indian" before Mercury's descent "Parsi", and another camp, represented by Wiki libs, removed the insertion.

- *Revision annotations.* We select the n (n = 5) most common standpoints (standpoints with most revisions) to annotate. To lay out the annotations, we divide the screen into grids and enumerate the nearby grids until finding an area that does not cause overlap.

4.3 Revision Graph of Multiple Sentences

In previous sections, we have described the use of the revision graph to visualize the evolution of a single sentence. Building on sentence-level encoding, we can also define and encode higher-level relations. We introduce the revision-level relations and the editor-level relations. The relation between two revisions can be defined by the relations between sentences. Two revisions are related if they have edited the same sentence. For a selected revision, each changed sentence has its evolution process, which can be visualized by aggregating, ranking, and linking the revisions that edit it. The only difference with the single sentence view is that a revision might belong to several clusters (since the sentences generate the clusters independently). In this case, we merge these clusters to ensure that each revision belongs to no more than one cluster. For example, Figure 1 (b) shows the revision graph regarding a selected revision, in which several sentences are changed by replacing the word "owners" with words like "humans". The editor-level relations are built on the basis of the revision-level relations. Editors relate to each other if there are relations between their revisions. Through merging the revision-level graphs, we can get the editor-level revision graph. In this graph, users can understand who interacts most with the selected editor and what content they focus on.

5 Edit-History Vis System

Building on the revision graph, we designed and developed the Edit-History Vis system. Our system supports an enhanced reading experience by providing some simple visual hints on the article view (Figure 6 (a)). When observing controversial content in the article, users can select the sentence to view the details on the revision graph (Figure 6 (c)). We also provide an overview of the revision history (Figure 6(b)) and an information list panel (Figure 6 (d, e, f)) to support deeper analysis of the editing events.

- The **article view** shows the text content of the revision. The color shades of the sentences indicate the controversy. We marked the controversial sentences and changed sentences on the scroll bar (**T3**). Users can easily browse the article and select interesting sentences to view its revision history.
- The **overview** shows the temporal distribution of the number of edits (the bar chart) and the size of the article (the line chart) (**T1**). We marked the controversial revisions on the line chart (**T4**). Users can select a revision to view the edit and the revisions related to it.
- The information list panel includes the **token list view**, the **sentence list**, and the **selection information view**. The token list view shows frequent text tokens which are ranked by the frequency (**T3**). The sentence list view shows the changed sentences regarding the selected revision or token, allowing users to select sentences directly on this view. Due to space limitations, each sentence is represented by the most frequent tokens that are inserted or deleted. The selection information view displays the basic information regarding the current selection.
- The **revision graph** is the main view of the Edit-History Vis system, which visualizes the edits regarding the users' selection. If a sentence is selected, the revision graph visualizes the editing process of the sentence. Users can interact with the revision graph to view details of these edits or choose other interesting objects they detect on the revision graph. The revision graph supports selecting a sentence (by clicking the corresponding curves) and selecting a revision (by clicking the node).

6 Usage Scenarios

In this section, we introduce the exploration of three Wikipedia articles with Edit-History Vis, namely "Cat", "Cyclone Larry", and "Freddie Mercury". Our discoveries from these cases are summarized as follows. **Vandalism** occurs frequently in the edit history, including mass deletion and insertion (Section 6.1) and adding phony words (Section 6.2). **Conflicts** sometimes involve a few editors, and a quick compromise can be reached (Section 6.1). In other cases, the conflict evolves into a severe edit war, either over unimportant elements such as styles of templates (Section 6.1) or involves conflicting cultural perceptions (Section 6.3). **Authenticity** of Wikipedia content should be treated carefully. Incorrect information could survive on the page for a long time (Section 6.2). Wikipedia is "in **progress**", where editors actively refine the page content (Section 6.3).

6.1 Exploring "Cat"

The article "Cat" has attracted over six thousand editors and more than ten thousand revisions. The editing process is not peaceful.

Mass deletion and insertion. Sharp decreases and increases are observed in the overview (Figure 1 (a1)). Some editors removed almost all the content on the page. Clicking on the sentences, we observed repeated deletions and insertions (Figure 1 (a2)), representing the process of vandalism and repair. The numerous vandalism edits and the repair edits occupy a large space and distract users from viewing the normal revisions. Therefore, we provide a "hide

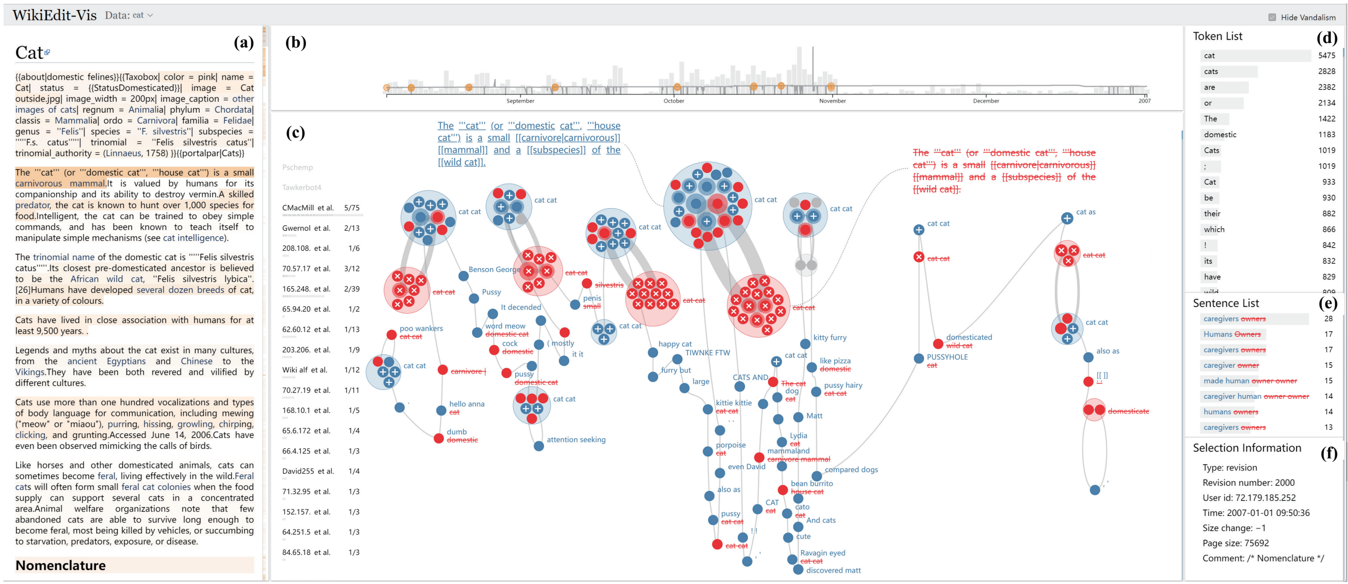

Figure 6: The system interface of Edit-History Vis. The article view (a)presents the text content of the article which is augmented by color shades representing controversy. The overview (b) uses a line chart to show the trend of the page size and a bar chart to show the distribution of the number of edits. The revision graph view (c), which is the main view of the system, shows the development of the editing events, including when the edits occur, what standpoints exist, and how authors conflict and cooperate. The token list view (d) and the sentence list view (c) show the tokens and sentences related to the selected event. The selection information view (f) provides basic information about the current selection.

vandalism" option for users to filter out the vandalism and repair edits. In the following cases, mass deletions and additions are filtered out by default.

Compromise. In the overview of the revision graph, the editor Katzenjammer contributed a lot at the beginning but stopped to edit the page later (Figure 1 (b1)). Figure 1 (b2) shows the changed sentences in one of Katzenjammer's edits, and Figure 1 (b3) shows the revisions regarding these sentences.. In the selected revision, Katzenjammer changed "owner" to "caregiver" and "human" and commented that he thought these expressions were more accurate. However, the other editor Ramdrake thought the changes were POV (point of view) edits which violated Wikipedia's NPOV (neutral point of view) rule[7] and reverted Katzenjammer's revision. The "caregiver" was also reverted by Ramdrake. Interestingly, a few days later, Ramdrake replaced "owner" with "human guardian", a statement between "owner" and "human companion". A compromise was achieved between Katzenjammer and Ramdrake.

Edit war. Wikipedia provides templates for creating infoboxes. On the page "Cat", the sentence that sets the color of the template is extremely controversial. Figure 1 (c) shows the changes in the sentence from August to November 2008, which reveals an edit war over the template color. There are five main camps, namely pink, default, orange, blue, and black. "Pink" was supported by most editors, followed by "default" (removing the sentence). Some editors are vandals. For instance, 75.16.166.224 changed the color haphazardly to black, orange, blue, and red. To prevent conflicts on such unimportant elements, Wikipedia contributors have created pages suggesting editors not to engage in worthless edit wars (e.g., Wikipedia: Don't edit war over the color of templates[8]).

6.2 Exploring "Cyclone Larry"

"Cyclone Larry" introduces the meteorological features and impact of Larry, a tropical cyclone that formed in 2006.

Adding phony words. Some sentences introducing the basic information (e.g., name, year), which ought to be objective and

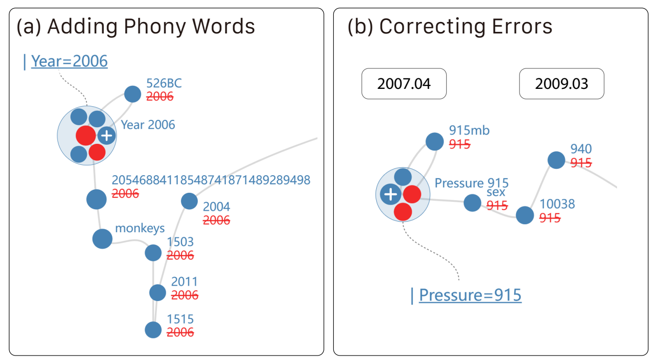

Figure 7: Exploration on "Cyclone Larry". (a) vandals changed the form year of Larry into other years and irrelevant words (a). (b) The pressure of Larry was corrected two years after it was created.

non-controversial, have undergone many vandalism edits. Vandals change the correct value into false values or insert funny tokens. Figure 7 (a) shows the vandalism and repair edits on the table row "year: 2006". Editors changed "2006" into other years, long strings of numbers, and irrelevant words such as monkeys. Wikipedia is vulnerable to vandalism since any casual users can edit the pages. Some editors are dedicated to monitoring the edits, and they make quick responses to vandalism edits. In "Cyclone Larry", the editor that contributes the most revisions is ClueBot NG, a robot developed by Wikipedia users to fight vandalism[9].

Correcting errors. There have been long debates on the authenticity of Wikipedia content. As for "Cyclone Larry", we are interested in whether the meteorological data recorded on this page is correct. Figure 7 (b) shows the changes of the table row "pressure: 940", which describes the lowest pressure of Larry. This element was created in April 2007, a year later than the form time of Cyclone Larry, with the value "915". Despite some vandalism edits, "pressure 915" survived for over two years and over three hundred revisions until an editor changed it into "940". On the one hand, misinfor-

[7]https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

[8]https://en.wikipedia.org/wiki/Wikipedia:Don%27t_edit_war_over_the_colour_of_templates

[9]https://en.wikipedia.org/wiki/Vandalism_on_Wikipedia#ClueBot_NG

mation in Wikipedia can be corrected by collaborative efforts. On the other hand, the correcting process might be long, especially in articles receiving less attention from editors (for reference, "Cyclone Larry" has over 1,500 revisions currently, and the correctness was made in the 952nd revision).

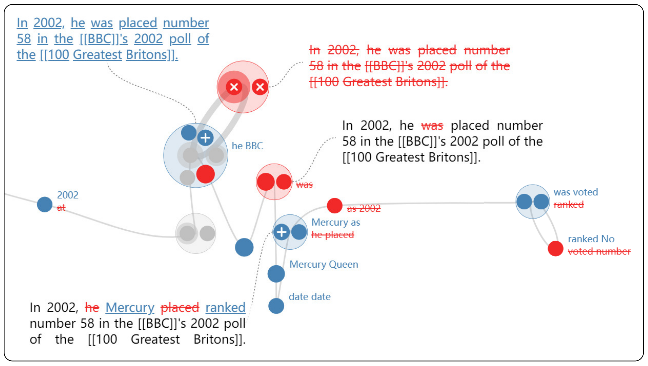

Figure 8: A progressive revision process in "Freddie Mercury". Editors discussed whether the necessity and refined the expression.

6.3 Exploring "Freddie Mercury"

Freddie Mercury was a British singer and songwriter whose parents were from the Parsi community of western India, and there have been debates on Mercury's ancestry.

Edit war. Figure 5 shows the editing process of the controversial sentence "Mercury, who was of Parsi descent and grew up in India, has been referred to as Britain's first Asian rock star." Editors debated on the ancestry of Freddie Mercury. We can see the two major camps of editors. One camp, represented by Dewan357, insisted on inserting "Indian" (or "[[Indian]]", where the square brackets denote an internal link to the Wikipedia page "Indian") before Mercury's descent "Parsi", and the other camp, represented by "Wiki libs", kept removing the insertions. This division of editors' opinions is due to the history of Parsis' migration from Persians to the Indian subcontinent, which reflects conflicting cultural perceptions.

Progressive revisions. Though we have introduced a lot of vandalism edits and edit wars, many edits in Wikipedia articles are progressive revisions of the content, e.g., refining the expression, appending internal and external links, and adding citations. Figure 8 shows the revision process of the sentence "In 2002, Mercury was voted number 58 in the BBC's poll of the 100 Greatest Britons." There are two main types of revisions regarding this sentence, namely deciding the necessity and refining the expression. (1) Deciding the necessity. Editors deliberate on what sentences should be put into the article. Keditz deleted the sentence four times, commenting that the"article lacked depth before, it was all general info." Other editors, however, thought the sentence is necessary. (2) Refining the expression. Initially, the sentence was expressed as "he was placed...", and then an editor change it into active voice "he placed". Later, this expression was modified as "Mercury ranked", and another editor consider "was voted" as more appropriate.

7 Evaluation

To evaluate Edit-History Vis, we conduct a comparison with prior work, as well as a user study.

7.1 Comparison with Prior Work

Taking the tasks of both previous work and Edit-History Vis into consideration, we evaluated the functions of the tools from three perspectives, namely editor, content, and event. Our goal is to demonstrate the functions of our system, as well as the advantages of our design in analyzing editing events. Table 1 shows the summarized functions of the selected tools. Among these tools, History Flow [29] and WikiDashboard [25] aimed to reveal the overall collaboration patterns, providing effective indicators for further exploration. Revert Graph [15] and WhoVIS [10] supported analyzing the conflict between editors. WhoCOLOR [11] and Contropedia [6] combined multiple views to support a more comprehensive exploration of the editor interaction and content controversy. However, in these tools, the editor and the content are often separated, making it inconvenient to understand what the editors were debating. Edit-History Vis features a coupling of the editor perspective and the content perspective to visualize the variation of editors' standpoints. To demonstrate this, we explain the comparison regarding granularity, detection of editing events, and analysis of standpoints. The detailed explanation for Table 1 is attached in the supplemental material.

Supporting analyzing the sentence-level editing process. Different from prior work, Edit-History Vis supports visualizing the evolution of a sentence. WhoVIS, WhoCOLOR, and Contropedia conducted sentence-level and word-level comparisons to generate higher-level information (e.g., the conflict between editors). This extraction introduces a loss of information regarding the dynamics of the text content and editor interaction. For instance, the standpoint of an editor might change (Figure 1 (b)), which is probably overlooked in prior tools. By contrast, our system preserved and visualized the evolution process.

Supporting detection of various events. Except for WikiDashboard, which is not aimed at detecting the specific event, and History Flow, which can reveal the overall patterns of editors' conflict and cooperation, the remainder of the compared work focuses on the conflicts. Our system supports detecting and analyzing various editing events compared with other works.

Supporting detailed analysis of standpoints. While prior work studies the editors' opinions structurally by dividing editors into different camps, our system enables analyzing the standpoints semantically and comprehensively. Revert Graph and WhoVIS used an editor-editor graph to visualize the conflict relationships between editors, indicating opposing standpoints. However, the dynamics of various standpoints are neglected. It is also difficult to see the details of how standpoints differ among editors due to the separation of editors and content. In comparison, we aggregate and lay out the nodes according to standpoints on the revision graph, and keywords of the standpoints are annotated beside the nodes, showing the details of the changes in editors' opinions.

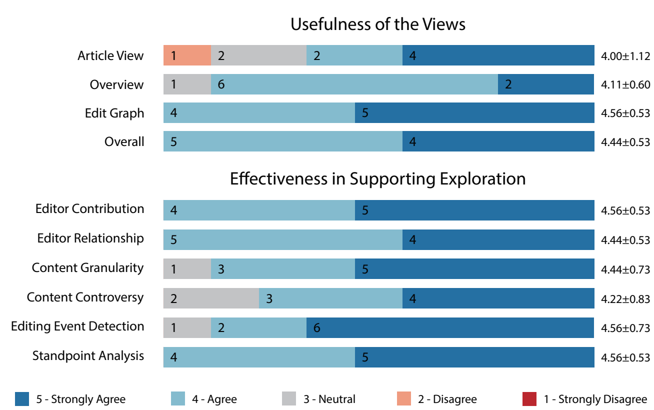

Figure 9: Ratings for the usefulness of the views, as well as the effectiveness in assisting users' exploration of the Wikipedia edit history from the editor, the content, and the event perspectives.

7.2 User Interview

We conducted a user study to evaluate the usefulness of the views in our system, as well as the effectiveness of Edit-History Vis in supporting users to understand the edit history of a Wikipedia article.

Table 1: Comparison with prior work.

	Editor Perspective		Content Perspective		Event Perspective	
	Editor Contribution	Editor Relationship	Granularity	Content Controversy	Editing Event Detection	Standpoint Analysis
History Flow [29]	✓	⍻	sentence	⍻	vandalism & conflict & progress	×
WikiDashboard [25]	✓	×	revision	×	⍻	×
Revert Graph [15]	⍻	✓	revision	×	conflict	⍻
WhoVIS [10]	⍻	✓	word	⍻	conflict	⍻
WhoCOLOR [11]	✓	⍻	word	✓	conflict	⍻
Contropedia [6]	⍻	✓	sentence & word	✓	conflict	⍻
Edit-History Vis	✓	✓	**sentence & word**	✓	**vandalism & conflict & progress**	✓

Participants. We recruited nine participants including three females for our user study. Three participants were undergraduate students, and the remaining six were postgraduate students. We asked them to specify their knowledge of Wikipedia. Seven participants visited Wikipedia articles regularly, and two visited occasionally. All participants have not participated in editing Wikipedia articles, and are not familiar with the collaboration mechanism of Wikipedia.

Procedure. First, we asked participants to fill in their basic information and give them a short tutorial about Edit-History Vis. Participants were guided to explore the case of "Cyclone Larry" to get familiar with our system. Then, we asked the participants to explore three cases in ascending order of complexity. For each case, we assigned several tasks from the editor, content, and event perspective. The first case is to understand the editing process of a given sentence in the article "Zhou Qi", the second is to explore the editing event regarding a given revision in the article "Cat" (Figure 1 (b)), and the third is to identify and analyze controversial content. The tasks were like "select the major editors", "judge the relationship between two major editors", "judge the type of the event", and "analyze the major standpoints". Then, the participants were allowed to explore the cases they were interested in freely to observe and analyze events. The study ended with a questionnaire on the subjective rating of Edit-History Vis functions and a short interview. In the interview, we asked participants' opinions on the effectiveness of Edit-History Vis and changes in their understanding of the edit history of Wikipedia before and after using the system. The study session lasted about 30 minutes. The full tasks and accuracies are attached in the supplemental material.

Participant feedback. Overall, Edit-History Vis received an enthusiastic reaction from the participants. The average accuracy of the tasks is 91.11%. As shown in Figure 9, participants approved the functionality and effectiveness of Edit-History Vis.

- *The revision graph is helpful for understanding the editing event.* All participants got the right answer for the first task, including identifying the main editors, judging the relationships between the main editors, and describing the development of the event. P5 emphasized that the revision graph shows the dynamics of the interaction between editors clearly, which is helpful for comprehending the event.
- *It would be better if there are some annotations of events.* Though the participants discovered various editing events during the free exploration, they reported that when the controversy is severe, it is hard to distinguish between subevents from the complex graph. In the third case, they were asked to detect the two most controversial sentences and select all types of events they observed for each sentence. All participants successfully detected the sentences, but the accuracies of specifying events were 63.89% and 69.44%. Most participants detected one type of event, ignoring the others.
- *Edit-History Vis improves the knowledge of Wikipedia edit mechanism.* All participants reported that they got more aware of the editing process of Wikipedia articles after the exploration. P3, P4, and P5 were surprised by the huge amount of vandalism edits occurring in the editing process.

8 Discussion

We discuss the scalability of the revision graph and the system, the limitations and future work of Edit-History Vis, and how our system can be combined with prior tools.

8.1 Scalability of Edit-History Vis

The performance of the revision graph is affected by the number of revisions shown on the graph, with a limit of around 300 revisions. For example, there are 220 revisions in Figure 6. This limit is sufficient as most sentences have no more than several dozens of revisions. The limit of multiple sentence selection is 10, which is also sufficient when selecting a single revision. The whole system supports analyzing the edit history with up to 3000 revisions, which is able to cover the whole editing history of most Wikipedia articles (e.g., "Cyclone Larry" has around 1600 edits over the past sixteen years). However, for popular articles like "Cat" which has over 10000 revisions, the exploration is limited to a specific time span. In the future, we can refine the aggregation algorithm and enable more flexible filtering and zooming on the time to support more revisions and a longer time span.

8.2 Generalizing the Revision Graph

The revision graph, which is designed for visualizing the detailed editing history of a Wikipedia article, can also be applied to other collaborative editing processes, which feature the *time*, the *text change*, and the *editor* attributes. The calculation of standpoints can be tuned by modifying the editor gravitation and the text gravitation according to the scenario.

For example, we can tune the force model to visualize the revision history of papers. In Wikipedia edit history, we consider that the *standpoint* of an editor is consistent on the whole, so there is gravitation force between edits by the same editor. However, when writing a paper, the authors often override their own words. By decreasing the *editor gravitation*, we can emphasize the deliberation process. Moreover, we can integrate the semantic distance of the texts into the *text gravitation*, changing the current binary model (0 for unequal and 1 for equal) into a more sophisticated one where different text pairs are assigned with different weights.

8.3 Supporting Explicit Event Exploration

The Edit-History Vis system supports observing and analyzing events by exploring controversial objects but does not provide an explicit representation of these events. As is revealed in the user study, when the controversy is severe and there are several events regarding the controversial object, revisions regarding different events might mix up, which makes users ignore some of them. Taking a step further, we can calculate the events, mark them on the timeline, and users can click an interesting event to explore it in the revision graph. To implement this, we only need to change the input data of the revision graph view, which can be viewed as the basic selection data (revisions related to chosen tokens, sentences, or revisions) within a time range or filtered by other conditions. Accordingly, more intelligent interactions can be added, such as pinning an observed event and looking up the events in which an editor is involved.

8.4 Integrating Prior Tools

We positioned our main contribution as providing a visualization method for browsing and analyzing the editing process of fine-grained Wikipedia elements, which fills the gap in existing visual tools. By combining individual visual tools, more effective visual analytics systems can be achieved [11]. Based on the Edit-History Vis system, we discuss how prior tools can be integrated.

- *Controversy and event detection.* We can apply the prior controversy measuring algorithms [6] to provide hints on controversial elements and events.
- *Overview.* We focused on visualizing detailed information in this work, and the system will bring more insight if equipped with a more effective overview [5, 25].
- *Editor relationships.* In our system, editors are arranged in one dimension. Adding a supplemental editor graph view [10] can help to reveal the editors' relationships more straightforwardly.

9 Conclusion

In this work, we proposed Edit-History Vis, a visual analytics system that enables users to explore and analyze the Wikipedia edit history. We designed the revision graph to visualize the detailed development of controversial elements and editing events. In the revision graph, we embedded the standpoints of revisions in the vertical dimension by integrating text and author information with a force-directed model. We improved the readability of the revision graph by aggregating overlapping revisions, linking related revisions, and annotating the text changes. Three cases and the user study verified the effectiveness of our system.

Our system is an attempt to reveal the complicated editing process of Wikipedia articles in an easy-to-understand way. We discussed the scalability of the system to increase users' awareness of the underlying mechanisms of mass collaboration documents, which helps analyze authors' standpoints and judge the quality and reliability of the content. In the future, we can generalize the revision graph and integrate prior methods to make the system a useful visual tool.

Acknowledgments

This work is supported by NSFC No. 62272012 and No. 61872013.

References

[1] B. Adler, L. De Alfaro, and I. Pye. Detecting wikipedia vandalism using wikitrust. In *Notebook papers of CLEF*, pp. 22–23, 2010.

[2] B. T. Adler, K. Chatterjee, L. De Alfaro, M. Faella, I. Pye, and V. Raman. Assigning trust to wikipedia content. In *Proceedings of the International Symposium on Wikis*, pp. 1–12, 2008.

[3] B. T. Adler, L. de Alfaro, I. Pye, and V. Raman. Measuring author contributions to the wikipedia. In *Proceedings of the International Symposium on Wikis*, pp. 1–10, 2008.

[4] M. S. Alharbi, T. Cheesman, and R. S. Laramee. Transvis: Integrated distant and close reading of othello translations. *IEEE Trans. Vis. Comp. Graph.*, 28(2):1397–1414, 2020.

[5] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Trans. Vis. Comp. Graph.*, 22(1):559–568, 2015.

[6] E. Borra, E. Weltevrede, P. Ciuccarelli, A. Kaltenbrunner, D. Laniado, G. Magni, M. Mauri, R. Rogers, and T. Venturini. Contropedia - the analysis and visualization of controversies in wikipedia articles. In *Proceedings of The International Symposium on Open Collaboration*, p. 1, 2014.

[7] A. Chhabra, R. Kaur, and S. Iyengar. Dynamics of edit war sequences in wikipedia. In *Proceedings of the International Symposium on Open Collaboration*, pp. 1–10, 2020.

[8] M. Fisichella and A. Ceroni. Event detection in wikipedia edit history improved by documents web based automatic assessment. *Big Data and Cognitive Computing*, 5(3):34, 2021.

[9] F. Flöck and M. Acosta. WikiWho: Precise and efficient attribution of authorship of revisioned content. In *Proceedings of the International Conference on World Wide Web*, pp. 843–854, 2014.

[10] F. Flöck and M. Acosta. WhoVIS: Visualizing editor interactions and dynamics in collaborative writing over time. In *Proceedings of the International Conference on World Wide Web*, pp. 191–194, 2015.

[11] F. Flöck, D. Laniado, F. Stadthaus, and M. Acosta. Towards better visual tools for exploring wikipedia article development—the use case of "gamergate controversy". In *Proceedings of International AAAI Conference on Web and Social Media*, pp. 48–55, 2015.

[12] A. Hardy and W.-H. Steeb. *Mathematical tools in computer graphics with C# implementations*. World Scientific Publishing Company, 2008.

[13] S. Jänicke, A. Geßner, G. Franzini, M. Terras, S. Mahony, and G. Scheuermann. TRAViz: A visualization for variant graphs. *Digital Scholarship in the Humanities*, 30(suppl_1):i83–i99, 2015.

[14] S. Jänicke and D. J. Wrisley. Interactive visual alignment of medieval text versions. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 127–138, 2017.

[15] 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*, pp. 453–462, 2007.

[16] F. Moretti. *Graphs, maps, trees: abstract models for a literary history*. Verso, 2005.

[17] E. W. Myers. Ano (ND) difference algorithm and its variations. *Algorithmica*, 1(1):251–266, 1986.

[18] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. Bieber no more: First story detection using twitter and wikipedia. In *Sigir Workshop on Time-aware Information Access*, pp. 16–76, 2012.

[19] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

[20] D. Schmidt and R. Colomb. A data structure for representing multi-version texts online. *International Journal of Human-Computer Studies*, 67(6):497–514, 2009.

[21] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[22] S. Silvia, R. Etemadpour, J. Abbas, S. Huskey, and C. Weaver. Visualizing variation in classical text with force directed storylines. In *Workshop on Visualization for the Digital Humanities*, 2016.

[23] J. Stefan, G. Franzini, M. F. Cheema, and S. Gerik. On close and distant reading in digital humanities: A survey and future challenges. In *Eurographics Conference on Visualization*, 2015.

[24] T. Steiner, S. Van Hooland, and E. Summers. Mj no more: Using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection. In *Proceedings of the International Conference on World Wide Web*, pp. 791–794, 2013.

[25] B. Suh, E. H. Chi, A. Kittur, and B. A. Pendleton. Lifting the veil: Improving accountability and social transparency in wikipedia with wikidashboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1037–1040, 2008.

[26] R. Sumi, T. Yasseri, A. Rung, A. Kornai, and J. Kertész. Edit wars in wikipedia. In *Proceedings of IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing*, pp. 724–727, 2011.

[27] R. Tinati, T. Tiropanis, and L. Carr. An approach for using wikipedia to measure the flow of trends across countries. In *Proceedings of the International Conference on World Wide Web*, pp. 1373–1378, 2013.

[28] T. Tran, A. Ceroni, G. Mihai, K. Djafari Naini, and M. Fisichella. Wikipevent: Leveraging wikipedia edit history for event detection. In *Proceedings of Web Information Systems Engineering*, pp. 90–108, 2014.

[29] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 575–582, 2004.

[30] T. Yousef and S. Janicke. A survey of text alignment visualization. *IEEE Trans. Vis. Comp. Graph.*, 27(2):1149–1159, 2020.