# Comparison of Tools for Digitally Tracking Changes in Text

**Columbia University Data Science Institute**

**Eve Washington[1], Bernat Ivancsics[1], Emily Sidnam-Mauch[2], Ayana Monroe[3], Kelly Caine[2], Susan McGregor[1]**
**Columbia University[1], Clemson University[2], The University of North Carolina at Chapel Hill[3]**

## Introduction

Over the past four decades, the need to trace and attribute changes to a digital text has been concentrated in two primary areas of digital technology use: word processing and software development. In this work, we survey the landscape of tools for digitally tracking changes in text and identify similarities and differences across tools.

## Background

widespread use of digital word processing by the late 1970s and early 1980s[2]

**1976** — the algorithm for the Unix diff command[1]

*1970*

**1987** — Introduction of "revision marks" in Microsoft Word[3]
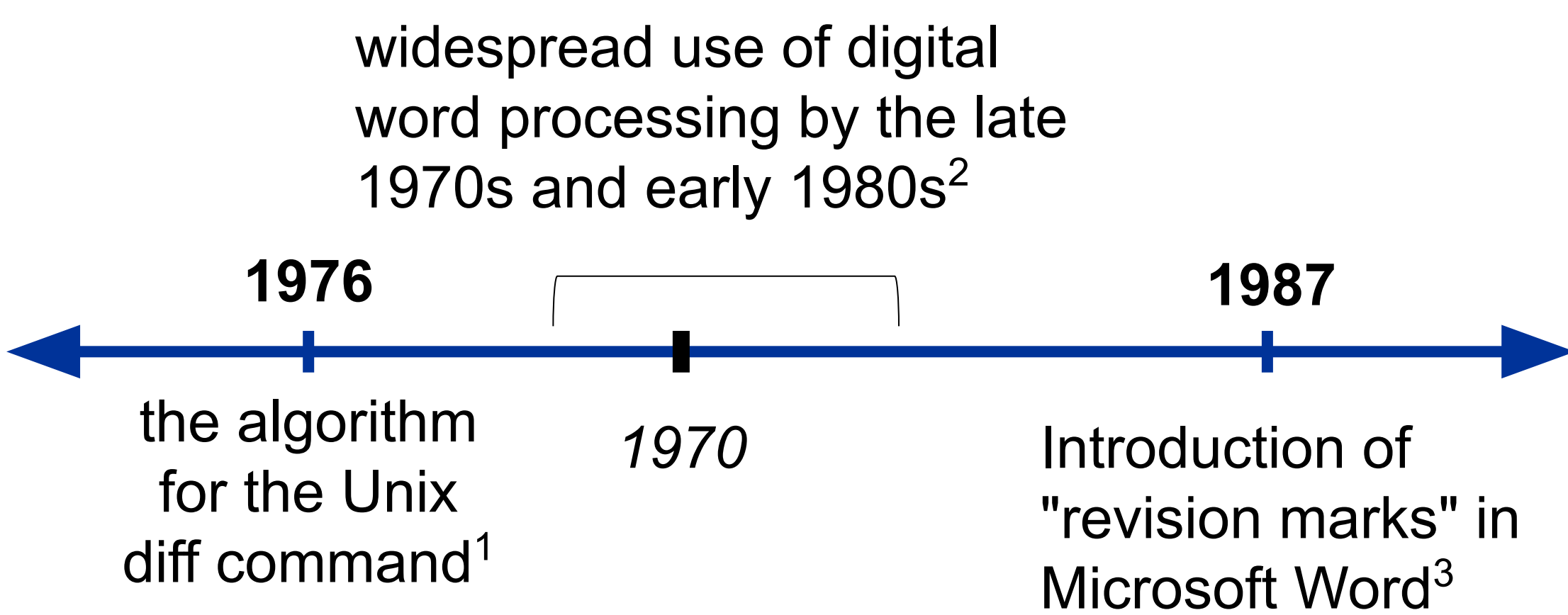
## Methodology

Our goal is to identify the user interface presentation mechanisms for the following elements in text tracking tools:

1. Authorship attribution;
2. Date/time of change;
3. Type of change (e.g. addition, deletion)
4. Substance of change (new text/old text)
5. Location of change within the document.

We reviewed four instances of each type of software tool (VCSs and word processing programs).

- Software: Git, Github, Visual Studio Code, IntelliJ IDEA
- Word Processing: Wikipedia, Word, Google Drive, Libre Office

Line Level Changes tracked, character changes highlighted

Red is deleted text, green is added

+/- symbols

*Figure 1:* **Github Diff**



Character level edit tracking, strikethrough text reflects deletions

Color indicates User

*Figure 2:* **Google Docs suggestion mode**



Yellow is old text blue is new

Paragraph Level editing, highlights indicate specific character changes

*Figure 3:* **Wikipedia Diff**



*Figure 4:* **Word's 'all markup' mode**

Change can be noted in general area or with precise location marked

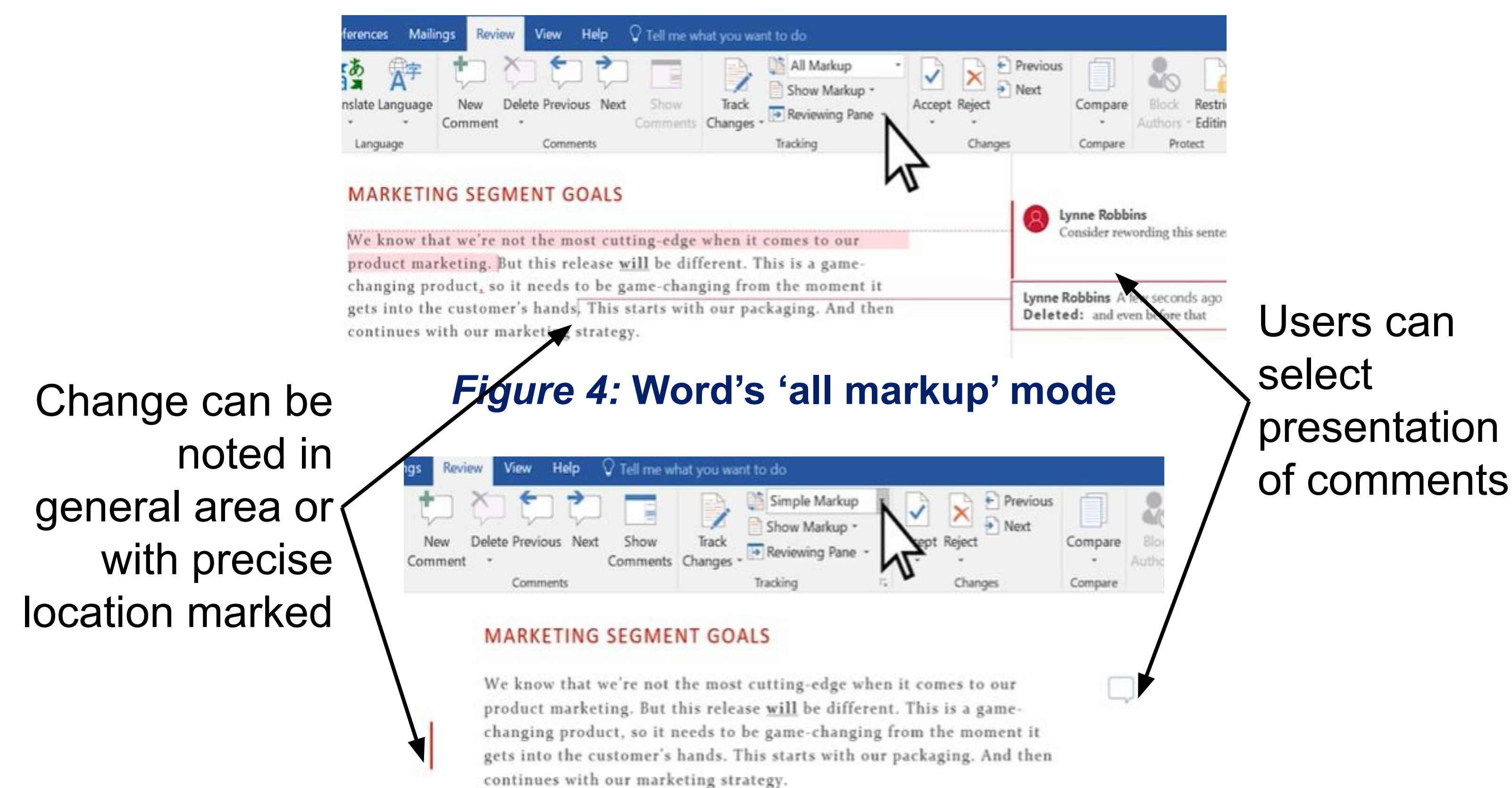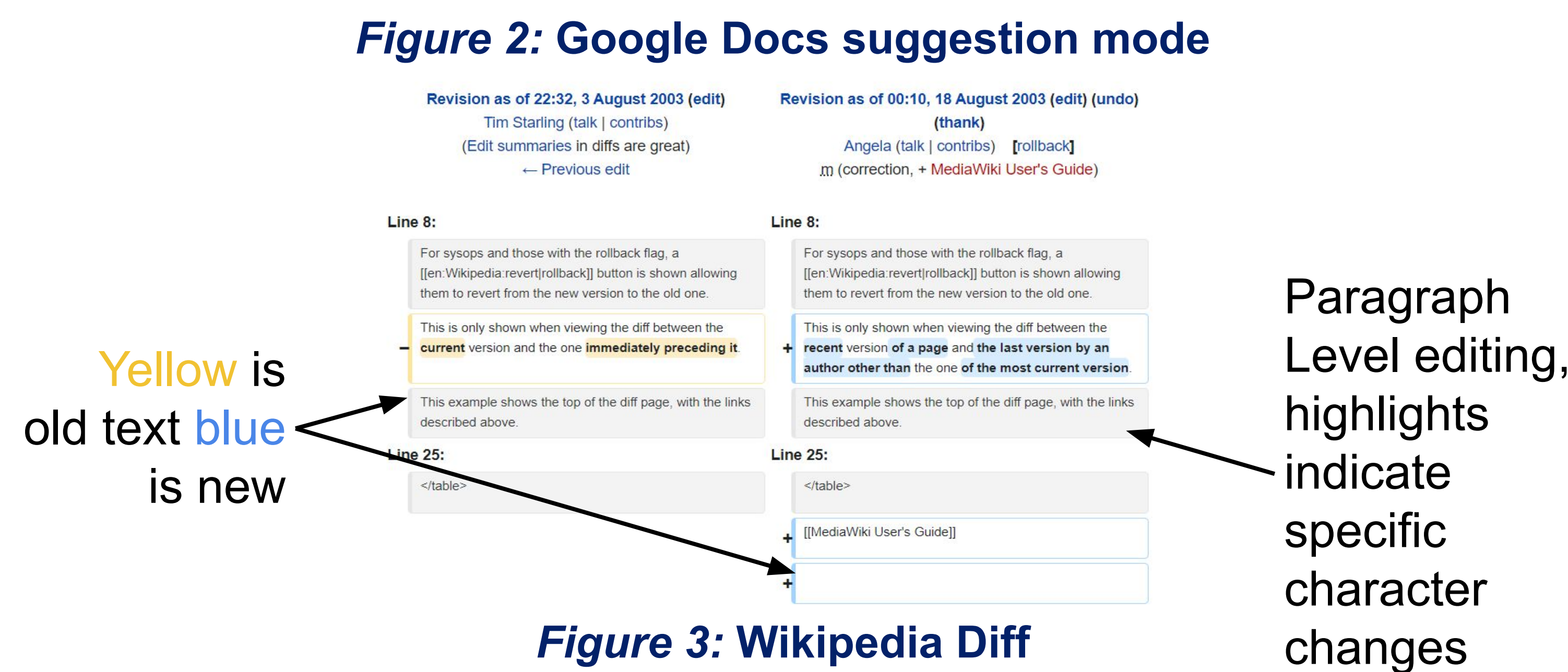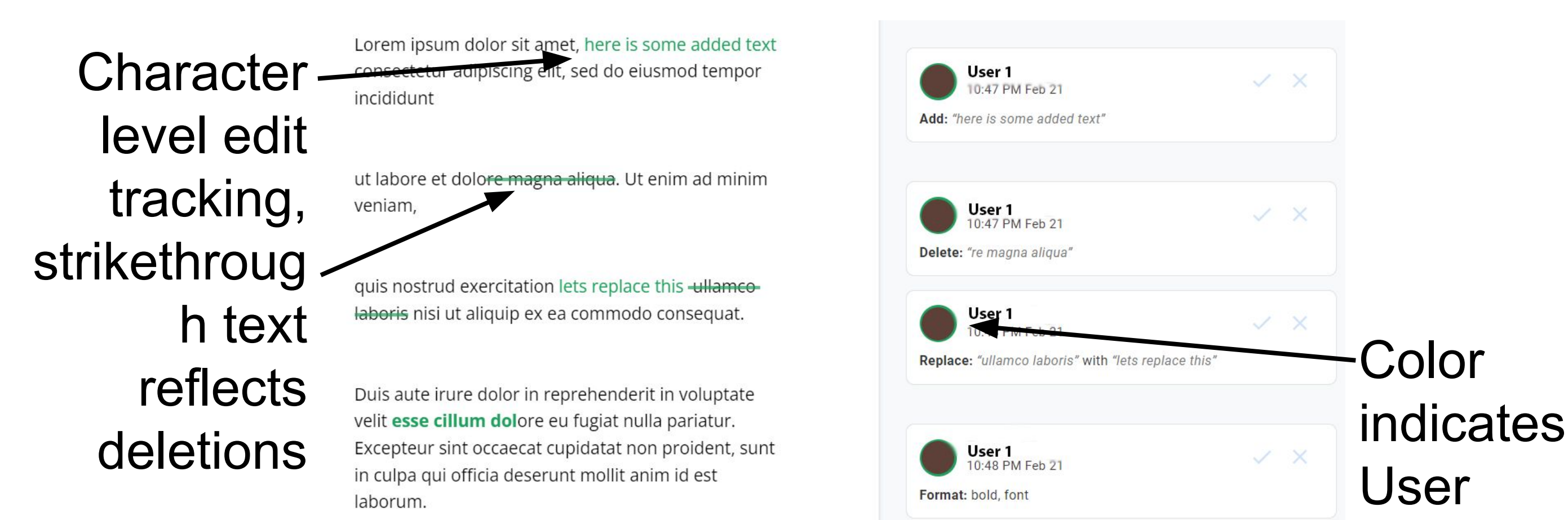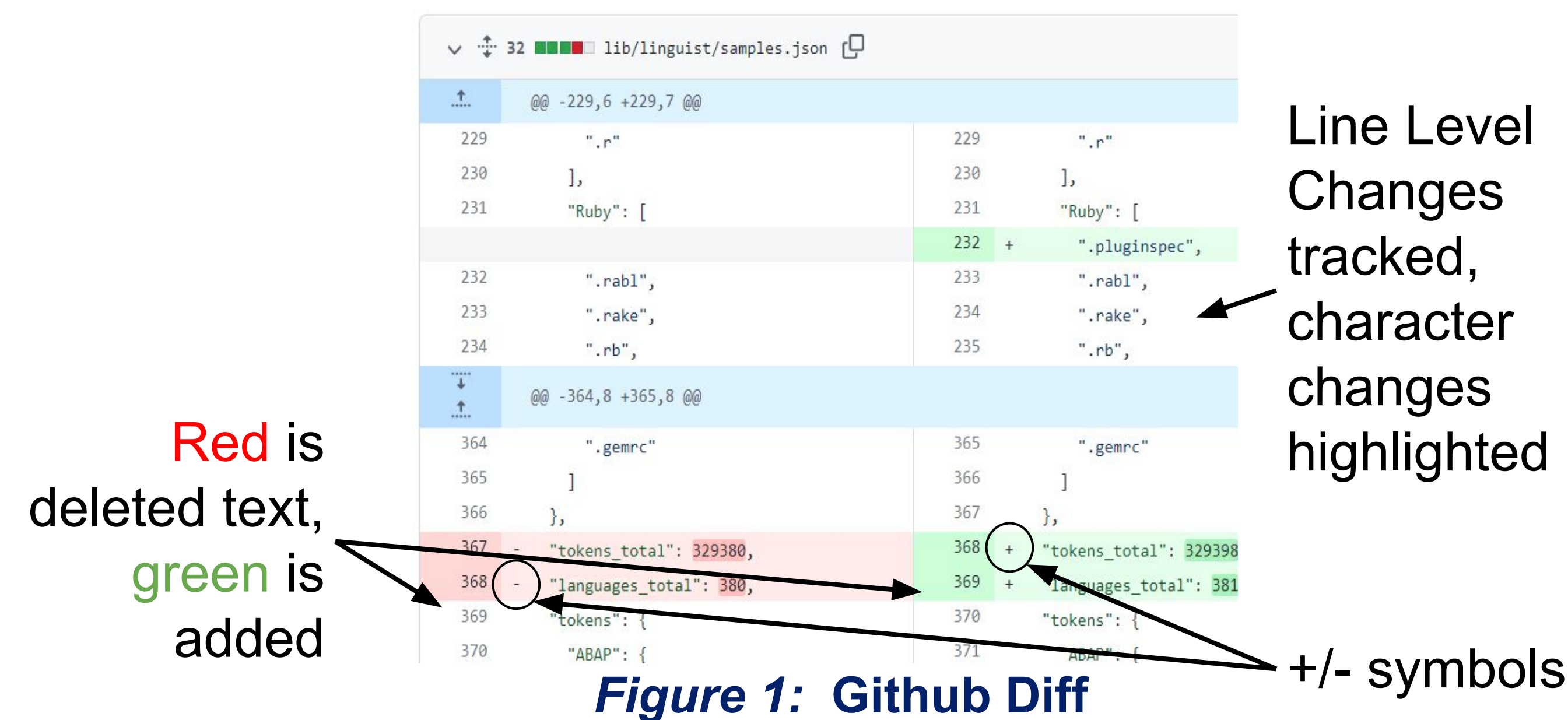Users can select presentation of comments



*Figure 5:* **Word's 'simple markup' mode**

## Discussion

All tools make use of text color and formatting to visually indicate text additions, deletions, and replacements.

While both types of tool can track precise text changes, the unit of emphasis differs, with code editors emphasizing changes at the line level (ex. Fig. 1) while word processing programs typically show character-level changes (ex. Fig. 2). Some of that distinction may be due to different cultures of use.

Usability Research[4] indicates that users prefer that similar tools use similar interface conventions, so the shared visual representations of text changes within tool types is not surprising.

Wikipedia (Fig. 3) represents a type of crossover, following code editor conventions at the paragraph level. Wikipedia remains the only substantive example of an effort to display change tracking on a large corpus of published, human-readable text, suggesting that there is opportunity for significant experimentation in this area.

Customization options, where available in five out of eight of the tools explored. Fig. 4 and 5 show one option for customization in Word. They offer a glimpse into the breadth options for visually representing changes in digital text, many of which are underexplored.

## References

[1] Hunt, J. W., McIlroy, M. D., & Bell Telephone Laboratories. (1976). An algorithm for differential file comparison. Bell Laboratories.

[2] Kirschenbaum, M. G. (2016). Track Changes: A Literary History of Word Processing. Harvard University Press.

[3] Inc, I. M. G. (1987). InfoWorld. InfoWorld Media Group, Inc.

[4] Experience, W. L. in R.-B. U. (n.d.). Jakob's Law of Internet User Experience (2 min. Video)