CEPT: Collaborative Editing Tool for Non-Native Authors

Yeshuang Zhu, Shichao Yue, Chun Yu*, Yuanchun Shi
Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
{zhu-ys13, ysc12}@mails.tsinghua.edu.cn, {chunyu, shiyc}@tsinghua.edu.cn

ABSTRACT
Due to language deficiencies, individual non-native speakers (NNS) face many difficulties while writing. In this paper, we propose to build a collaborative editing system that aims to facilitate the sharing of language knowledge among non-native co-authors, with the ultimate goal of improving writing quality. We describe *CEPT*, which allows individual co-authors to generate their own revisions as well as incorporating edits from others to achieve mutual inspiration. The main technical challenge is how to aggregate edits of multiple co-authors and present them in an easy-to-understand way. After iterative design, CEPT highlights three novel features: 1) cross-version sentence mapping for edit tracking, 2) summarization of edits from multiple co-authors, and 3) a collaborative editing interface that enables co-authors to examine, comment on, and borrow edits of others. A preliminary lab study showed that CEPT could significantly improve both the language quality and collaboration experience of NNS authors, due to its efficacy for sharing language knowledge.

Author Keywords
collaborative editing; non-native writing; knowledge sharing

ACM Classification Keywords
H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

INTRODUCTION
Language editing is important and necessary for second language (L2) writing [34]. Due to language deficiencies, non-native authors face various difficulties in L2 writing, such as generating appropriate wording and phrasing [31], interference from their first language [10], and so on. To deal with this problem, one can recruit a language expert to help edit. However, only a minority do so. The reasons include considerable expense, slow response, lack of domain expertise, and potential mistrust issues.

*Denotes the corresponding author.

CSCW '17, February 25-March 01, 2017, Portland, OR, USA

DOI: http://dx.doi.org/10.1145/2998181.2998306

In practice, collaborative editing is most often adopted by non-native authors, with the support of two kinds of collaborative tools [29]. Via an email system, non-native co-authors can share their edited documents with each other. In this manner, however, both version control and revision synthesis are manual, which are inefficient and error-prone. To overcome the shortcomings, online collaborative writing systems (such as Google Docs and MS Word on OneDrive) can synchronize edits among multiple co-authors. Studies have demonstrated the effectiveness of such tools for content composition and completion [29]. Unfortunately, these tools are not optimized for collaboration in the context of language editing. For example, Google Docs only allows collaborators to maintain a most recent version, and simply lists historical edits and comments beside the document. Such an interface is inconvenient for authors to synthesize revisions, and worse, reduces the willingness of each individual to contribute due to the social considerations of editing content directly [6].

The goal of our research is to design a collaborative editing system geared toward improving the language quality of L2 writing. The motivation is to enable multiple NNS co-authors to efficiently contribute and share their language knowledge to compensate for individual deficiencies. We ground the system on two theoretical bases. First, individual NNS authors may have different levels and corpora of English knowledge dependent on their unique learning trajectory, and thus can generate language expressions of varying qualities [38]. Second, although non-native authors cannot recall appropriate expressions sometimes, they have the ability to compare different expressions and recognize better ones [32].

With a pilot study and iterative design, we instantiate our idea as a fully functional online system called *CEPT*, which allows an NNS author to post a draft to the server and have multiple co-authors edit separate parallel versions. Individual co-authors can generate their own expressions, which will then be directly incorporated by others or inspire new ones. Our system highlights three novel features: 1) sentence mapping and edit tracking across revisions by different co-authors, 2) edit summarization that minimizes the author's effort to synthesize edits from co-authors, and 3) a collaborative interface that enables co-authors to examine, comment on, and borrow edits of others. We evaluated the performance of CEPT with a preliminary lab study. Results showed CEPT could effectively facilitate language knowledge sharing among NNS co-authors. It could significantly improve both language quality and collaboration experience.

RELATED WORK

Our research is most related to previous work on tools for collaborative writing and editing, as well as those on summarization and visualization of edits from multiple collaborators. In addition, we also review recent work on crowdsourcing/outsourcing approaches to improving writing quality.

Tools for Collaborative Writing and Editing

More and more authors are now writing together [20]. They rely on both online synchronous authoring tools and asynchronous reviewing tools for collaboration [29]. Various collaborative writing tools have been proposed in previous research [13, 22, 2, 1, 27, 28]. These tools enhance collaborative writing with communication and information sharing functions including annotation, messaging, conferencing, and notification. Different roles of collaborators (e.g., co-author, commenter, and reader) are supported so that views of a document can be tailored according to each collaborator's position. Furthermore, in order to achieve concurrent document editing, novel paradigms [30] and data structures [25] have also been investigated for version management and conflict resolution. The ideas and techniques behind some of these works have already been implemented or integrated into synchronous documents like Google Docs.

Our research focuses on a more specific problem: we aim to promote language quality of NNSs by supporting sharing of language knowledge in collaborative editing, with an emphasis on how to resolve language issues. Current tools are not optimized for this task. They are inefficient for common tasks in collaborative language editing such as proposing, comparing, and synthesizing candidate revisions. Moreover, in synchronous settings authors refrain from contributing to other co-authors' text due to social considerations [6]. In contrast, our system works asynchronously and allows individual co-authors to generate candidates on their own copies, supported by novel aggregation and visualization methods.

Edit Tracking and Summarization

In collaborative editing, appropriate ways to communicate about changes to the document are important for effective understanding and information sharing [6]. *Flexible diff* [26] allowed authors to control what kinds of changes should be reported and how they should be visualized. The purpose was to reduce spurious and semantically meaningless entries in comparing text revisions. Fong et al. [14] derived and categorized high-level edits from the history of Wikis in order to present edits in an intuitive way and to develop a flexible measure of edit significance. Gehrmann et al. [15] adopted Levenshtein edit distance and topic modeling techniques to track paragraphs, identify significant edits, and predict parts of the paper that likely required editing as a result of previous edits. Xue et al. [39] proposed to automatically derive and label higher level semantics of edits as feedback for non-native authors to improve their writing skills.

Summaries of collaborative editing history are helpful for analyzing the activity and contribution of each collaborator. Wang et al. [37, 36] deployed two systems named *DocuViz* and *AuthorViz* that visualized how a Google Docs document evolved over time. The systems highlighted the text contributed by each co-author in each intermediate version throughout the writing process. Their summary visualization helped to identify authors' collaboration patterns. Summarization techniques are also helpful for instructor-student interaction in the context of online learning and MOOCs. They can help teachers to effectively browse course feedback [12] or grade students' answers at scale [3, 7]. Summary views can also be tailored to specific tasks, e.g., grading students' programs in programming classes [16, 17].

In our research, we proposed sentence tracking and summarization methods to effectively aggregate and visualize language-oriented edits to facilitate collaborative editing of NNSs.

Crowdsourcing/Outsourcing Approaches

Crowdsourcing has been proved to be capable of accomplishing complex tasks by decomposing them into micro and context-independent sub-tasks, and has been adopted in writing [4, 21, 19, 35]. Bernstein et al. presented *Soylent* [4, 5], which utilized crowds to perform text-related tasks including paragraph shortening, proofreading, and macro editing. For quality control, they proposed a three-step process where three separate groups of workers were recruited to first find parts of the text that can be improved, then propose revisions, and finally vote on the best candidates. With regard to collaborative writing, new paradigms have been proposed to facilitate collaboration between requesters and crowd workers. For example, *Ensemble* [19] was a collaborative story-writing platform where leaders directed the high-level outline for a story and crowd collaborators contributed meaningful ideas to it. *WearWrite* [24] enabled authors to write documents from their smart watches by dictating tasks and leveraging a crowd to help translate their ideas into text.

However, the key focus of crowdsourcing/outsourcing approaches is how to break writing into micro-tasks and recruit public crowd workers to complete the sub-tasks. Though effective, crowdsourcing/outsourcing is of limited use in document editing tasks, as it relies on public environments where uncertainty is ubiquitous due to the limited control of the process, the incompetence of workers, the risk of vandalism, etc. In contrast, our research focuses on collaboration among co-authors who are usually acquainted with each other and share background knowledge about the document, e.g., in the context of collaborative academic paper writing.

A PILOT STUDY

To gain insight into interface design requirements for collaborative language editing, we conducted a pilot study with four Chinese graduate students (P1-P4). P1 and P3 both provided an abstract from their paper drafts, and all four participants edited them collaboratively to improve language quality. However, as noted before, current synchronous tools impede co-authors' contributions, and traditional editing software does not support effective information sharing. Therefore, we mocked up an asynchronous, multi-versioned collaboration environment by presenting four copies of the same text in four separate windows of Google Docs. Each participant could only edit in her/his own workspace, but was free to view and comment on the other three revisions. After editing, we interviewed participants about their strategies and experiences in such a collaboration task and environment.

Collaboration Strategies

We found that in this pilot study, participants interacted with edits by collaborators actively and intensively. For example, when a participant identified and marked an error, others would notice that and also try to fix it; and when a participant had proposed a reasonable edit, others would follow by copying the edit into their own text. There were also occasions where three of the participants had edited the same place in different ways, the fourth would choose the best candidate to incorporate into her/his text. Besides, edits by different participants often overlapped with each other, which would lead to conflicts if participants had used a synchronous document.

Effect of Collaborative Editing

Overall, participants felt that collaborative editing could improve the language quality. Specifically, they mentioned that edits by others enabled them to notice more errors and acquire more alternatives for correction. "*It helped me find out those errors that I would have possibly missed when editing all by myself*" (P1). Different edits on the same area of the text "*provided more choices*". "*When several candidate edits were proposed for the same problem, I could pick out the best candidate quickly and confidently*" (P1). For these participants who had limited English proficiency, the alternatives were also highly valuable: "*those edits inspired me a lot and improved my English writing skills*" (P4).

During the study, participants were pleased to reach agreements with others. "*I was particularly glad when one of my edits were found to be exactly the same as others', whether the co-author had ever referred to mine or not*" (P3). "*In such cases (when a consensus was reached), I would be more confident on my edit. I thought it should be the best correction.*" (P4). Moreover, participants were not satisfied with only referring to edits by co-authors, they also wanted to interact with the edits in a variety of ways, such as directly incorporating part of the edits into their own text. "*I wanted to follow those good corrections by others but found no way other than typing or copying them manually. These processes were inconvenient and made me frustrated.*" (P2).

Issues in Presenting Edits

Participants also commented on the problems of presenting multiple document versions at the same time in collaborative editing. All of them emphasized that the edit mode was unsuitable for presenting the editor's intent, and that the search process for figuring out what others have done to the current sentence was quite distracting and time-consuming. Figure 1 shows the editing result by P1, which is among the four parallel versions. Character-level text changes in edit mode "*can be extremely hard to understand.*" (P1). "*I was often frustrated by edit mode... The same thing happened when I received an edited draft from my adviser. It would take a great effort for me to figure out what those strike-throughs and underlines actually meant*" (P4). The difficulty of following edits was exacerbated by the presence of multiple revisions. "*It took me a lot of time to locate sentences in paragraphs edited by co-authors as those revisions were quite different from each other. I would give up tracking changes in each co-author's revision to the same area of the original sentence*" (P1). "*If I could more efficiently locate in others' sentences the area that I was focusing on, I would be more willing to view edits by others*" (P3). As a result, participants worried that collaborative editing with multiple co-authors would take more time and effort. "*I am not sure whether proofreading collaboratively in existing document interfaces will really improve efficiency or slow me down at last*" (P3).

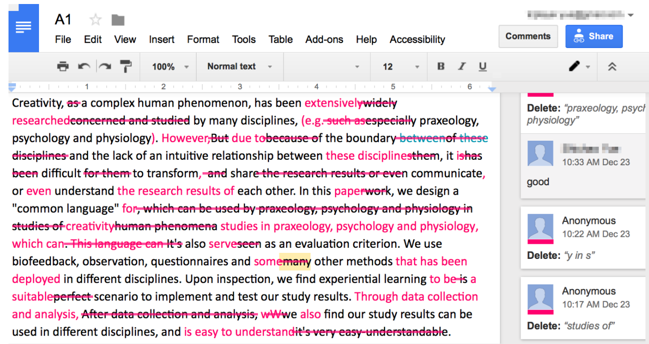

Figure 1. One of the four parallel texts in our pilot study. Various large areas of text insertion and deletion made it difficult for participants to capture the editor's intent. Besides, it was even harder to track changes across all the four revisions.

Lessons Learned

According to the results, we summarized three design insights of collaborative editing interface for NNS authors:

- An asynchronous, multi-versioned paradigm is promising for improving language quality and is acceptable to NNS co-authors. Co-authors generate diverse revisions, recognize better candidates, and mutually inspire each other.
- The efficiency of locating and grasping edits in different versions is key to authors' reception of collaborative editing. Hence, the visualization of edits should go beyond presenting the character-level histories as-is, by providing more semantically meaningful and cumulative results.
- For effective knowledge sharing, the interface should enable authors to directly interact with co-authors' edits, e.g., to incorporate, comment on, and vote for candidates.

INTERACTION DESIGN OF CEPT

Based on the guidelines as summarized above, we propose CEPT, a collaborative editing tool geared toward improving the quality and efficiency of L2 writing by facilitating language knowledge sharing in the editing process. CEPT features a novel interface that automatically aligns, aggregates and refines edits from parallel revisions, and enables co-authors to interact with the edits more efficiently (Figure 2). In this section, we describe the interface design of CEPT. We

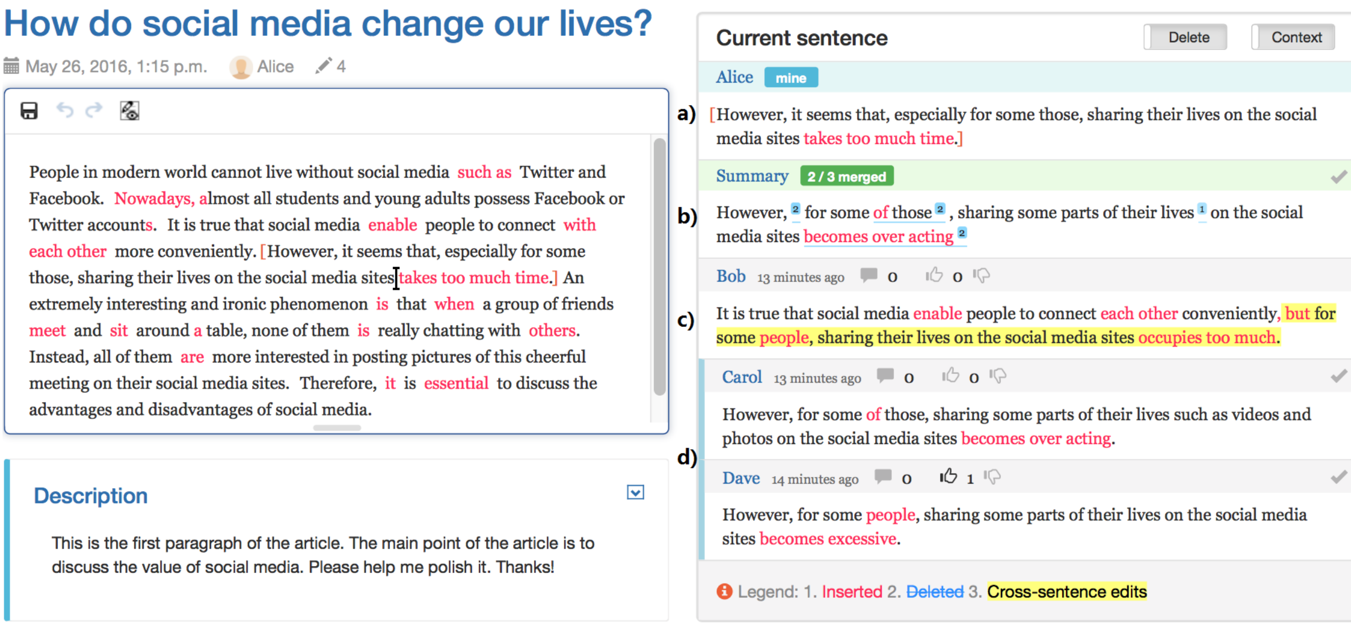

Figure 2. The interface design of CEPT. Left: the editor and the task posted by the author. Each co-author who has logged in will receive a copy of the text to correct. Right: a list of sentences aligned for reference. The sentence list consists of (from top to bottom) a) the currently focused sentence as in the editor; b) the automatically summarized sentence; c) the co-author's sentence with cross-sentence edits so that cannot be summarized into b); and d) other co-authors' revisions whose edits are already collected into b). All deleted contents are hidden by default to improve text readability.

will then discuss the implementation details in the Algorithms section.

CEPT: An Example of Usage

We use the following example to illustrate how CEPT works. Suppose that Alice, an L2 author, has finished a draft in which she has conveyed the main ideas almost clearly. But due to her insufficient English skill, the draft is full of syntax errors, poor and redundant word usages, improper sentence structures, and so on. So she asks three colleagues (Bob, Carol, and Dave) to help proofread and polish her draft in order to improve its language quality.

When Bob, the first co-author, looks at the draft, no one else has edited it yet, so he provides the first revision of the draft without any assistance. But the situation is different for the co-authors that follow, Carol and Dave. There already exists a revision by Bob, which can help the latter two as they make further edits. With the aligned sentences view of CEPT, they can quickly understand what others have done through a quick glance at the edit summary (Figure 2b). If a co-author wants to know exactly how others have modified the current sentence, they can refer to the candidate list where all corresponding revisions are displayed one by one. Moreover, if a co-author finds that an edit is good enough, they can simply click on a "borrow" button to incorporate that sentence into their own version. If they have already modified the original sentence, a dialog box for resolving conflicts will pop up and let the co-author decide which parts to incorporate.

When all the co-authors finish editing, Alice can use the same interface to browse revisions for each sentence and make final decisions on each revision with the help of the co-authors' votes and comments. Even though co-authors are often also NNSs, the final quality of the draft can be improved to a level higher than anyone of them could reach on their own, since their individual language knowledge has been efficiently aggregated and shared.

Asynchronous and Multi-Versioned Workflow

Each edit by co-authors contains valuable language knowledge and has the potential to improve language quality [38], but current synchronous documents like Google Docs cannot preserve diverse editing history. A previous study by Birnholtz et al. on the impact of social meanings of edits in synchronous environments revealed that co-authors often feel constrained when contributing edits to others' text [6]. The study also suggests presenting changes as possibilities for improvement rather than definitive changes, which is poorly supported by the reviewing and commenting features of existing tools.

We address the above issues by basing the design of CEPT on an asynchronous, multi-versioned workflow. As depicted in the above use case, an author posts a paragraph and each co-author revises separate copies. When a co-author submits their revision, it is shared among all co-authors. The author can also continue editing without the necessity of waiting until all co-authors have finished. Diverse revisions are all preserved and authors decide on the best candidate according to the intelligence of both the individual and the group.

However, multiple parallel versions pose an additional burden on authors to grasp and incorporate edits from collaborators. To address this challenge, we propose novel methods to aggregate, visualize, and enable authors' interaction with edits

a) Newton lost him~~s~~self in thought of the relation between objects.

b) Newton lost hims~~s~~self in thought of the relation between objects.

c) Newton lost him~~his~~self in thought of the relation between objects.

d) Newton lost himself~~hisself~~ in thought of the relation between objects.

Figure 3. Different appearances of four synonymous revisions. In real-time tracking, author's editing operations are presented as-is, which may result in various inconsistent appearances such as (a)-(d). The word-level diff-ing algorithm always produces results as (d) regardless of the underlying editing process, which is cleaner and more human-readable.

in parallel revisions. In the following sections, we will focus on these key features of CEPT.

Edit Aggregation and Visualization

As learned from the pilot study, the presentation of edits is key to co-authors' understanding and knowledge sharing efficiency. In order to minimize authors' cognitive load while reviewing edits by others, we propose several methods to aggregate and visualize text changes across parallel revisions. The designs also embody the principle of presenting changes as candidates [6].

Aligned sentences view

CEPT aligns corresponding sentences across different revisions, as shown in the right panel of Figure 2. The aligned sentences are listed in the following order: a) the author's current sentence being edited, b) the automatically summarized sentence, which we will describe later, c) the sentence(s) with cross-sentence edits that cannot be summarized into one sentence, d) other sentences edited by co-authors whose edits are already merged in the summary, and e) the number of unmodified sentences that are collapsed (if any). With these sentences aligned together, it is more convenient for authors to browse and compare parallel versions at a glance.

A particular issue arises when one-to-one correspondence of sentences is violated due to cross-sentence text changes, specifically, after merging and splitting of sentences. By inferring the sentence mapping relations between two revisions, CEPT can determine where in co-author's half or multiple sentences the currently focused sentence exactly corresponds. For example, if a co-author has combined two short sentences *A* and *B* into a longer one, CEPT will highlight the part that originated from *A* or *B* accordingly when other co-authors focus on *A* or *B* (Figure 2c). Similarly, if a co-author has split a sentence into two, the two new sentences will be both presented at the same time in the list of other co-authors. The visualization strategy also applies to merging and splitting more than two sentences. More details about our sentence aligning strategy will be described in the Algorithms section.

Moreover, in the aligned sentences view, the context of each sentence is hidden by default. If authors want to have a global view of the edited versions by other co-authors, or to examine the changes in sentence order, the context can be displayed by turning on the context toggle button.

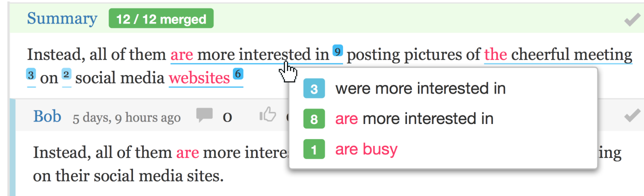

Figure 4. An example of edit summarization. Parallel sentence revisions are automatically summarized into one where edited areas are underlined and marked with the total number of edits. The drop-down menu lists the original text first (blue label), and then revisions edited by others as candidates (green label), each with a number of associated co-authors.

Refined edit presentation

CEPT follows the tradition of many commercial editors to support character-level change tracking. However, raw tracking results are often too overwhelming and disturbing to share among collaborators, as edit tracking at the character level can lead to enormous interrupted and semantically meaningless edits [26]. For example, Figure 3 shows four possible results of real-time edit tracking that essentially carry the same meaning. However, they do not take a consistent form as each co-author may produce a unique character-level edit sequence. As noted by participants in the pilot study, too many alternations of insertions (underlines) and deletions (strikethroughs) harm the readability and make the underlying ideas of revisions difficult to understand. The inconsistency also impedes the process of edit aggregation.

In order to reduce semantically meaningless edits, CEPT first takes advantage of a word-level diff-ing algorithm to compare two revisions and calculate word changes. Any insertion and deletion of characters related to a word is considered as a whole replacement of the word with a new one or empty one. As a result, the diff-ing result always takes the form as in Figure 3d, which is more robust and clearer than author's raw editing history. Second, neighboring word changes are merged into a longer replacement of phrase, which further reduces the disruption of the text flow. Third, to improve text readability, deleted words are hidden by default, unless the author clicks on a button to toggle their visibility.

Edit summary view

The summary view is designed as a final step to aggregate edits and minimize authors' effort to understand the thoughts of all other co-authors. As shown in Figure 4, parallel sentence revisions are automatically summarized into one sentence where edited areas are underlined and marked with the total number of edits. The number of edits is also reflected by the darkness of underlines and markers, indicating the "hotness" of edits in each area. By default, each edited area in the summarized sentence is filled with the candidate proposed by the most co-authors. On hover, more alternatives and the original text can be revealed in a drop-down menu, where synonymous edits given by different co-authors are presented as one candidate with a number of co-authors.

With the help of the summary view, co-authors can quickly spot potential areas to be improved and gain an overview of what all others have done. They can combine edits from dif-

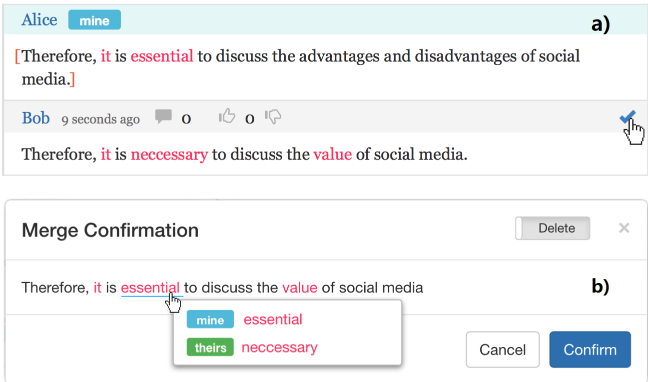

Figure 5. An example of edit borrowing: a) an author (Alice) "borrows" edits from one of her co-authors (Bob) to replace her own; b) the author decides how to resolve the conflicts, which happens only when both authors have changed the same words in different ways.

ferent co-authors by choosing a preferred candidate for each area to create better revisions, in coordination with the edit borrowing mechanism described below.

Interaction with Co-authors' Revisions

To further improve the efficiency of knowledge sharing, CEPT supports borrowing edits from others' revisions. Given a sentence to be revised, if the author is already satisfied with one of the existing revisions - either a particular co-author's version or the summarized one - they can update the whole sentence in their editor to match the preferred revision simply by clicking a "borrow" button (Figure 5a), instead of laborious and error-prone copy-pasting. CEPT also handles the situation when the sentence to be overwritten has already been modified by the author. If edited areas in both versions do not overlap, they are combined in the resulting sentence. Otherwise, the author needs to choose which solution to follow for each conflicted region (Figure 5b).

CEPT also supports commenting and voting on each revision. Comments can be used for raising questions, discussing solutions, and explaining authors' own edits. The number of votes determines the display priority of the revision in the sentence list. Borrowing edits from a co-author's revision will automatically trigger an up-vote for it.

ALGORITHMS

In this section, we describe the algorithms we have designed to support CEPT's interaction in detail.

Sentence Mapping

CEPT takes advantage of the HTML used in its rich text editor implementation to record and map sentences across revisions. Sentence delimiters, i.e., terminal punctuations, are wrapped in custom tags with unique and invariant IDs. Thus, sentences in different revisions with the same ID are considered to come from the same sentence in the original text. Even if sentences are reordered, the mapping can be preserved as long as the sentences are always moved together with their ending delimiters.

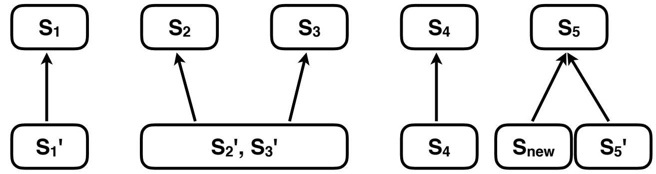

Figure 6. An illustration of the sentence mapping strategy. Sentences with only inner edits are mapped simply according to the IDs. Deleting the delimiter of S_2 leads to its merging with the next sentence to form (S'_2, S'_3), and each part of the new sentence is still aware of its ancestor. The inserted S_{new} points to the same ancestor as its next sentence S'_5.

In particular, if an author adds a new delimiter to the text, the current sentence is split into two. The new delimiter will be assigned with a special ID_{new} and at the same time with a pointer to the original sentence. Thus the two new sentences will both be linked to the ancestor sentence. If an existing delimiter is deleted, note that CEPT's editor is in edit tracking mode, so deleting does not actually remove the text but instead wraps the text with deletion tags. In such cases, the affected sentence will be joined with its next neighbor, and we can still refer to the old delimiter to tell which part of the new sentence is from which original sentence. The process of sentence mapping is illustrated in Figure 6.

Word-level Diff-ing to Extract Edits between Revisions

Based on the sentence mapping algorithm described above, we compare each revised sentence with its counterpart before editing to calculate inner edits, utilizing an adapted Myers' diff-ing algorithm [23]. In order to adapt the character-level algorithm to obtaining word-level differences, we first tokenize both sentences before and after editing into word and punctuation sequences (ignoring white spaces), and then apply the Myers' diff-ing algorithm to the two sequences to obtain minimized text differences at the word level. It is also important to clean up the differences in order to present them to authors in an easy-to-understand manner. Specifically, insertions are always placed before adjacent deletions, and text modifications involving multiple consecutive words are merged into one change.

Edit Summary and Merging

We propose a summary algorithm in order to aggregate all edits in multiple parallel sentence revisions into a single sentence. Given a revised sentence, we first find all corresponding sentences in parallel revisions, for each of which we compute the differences against their common original sentence. Then we combine these differences based on the following facts about the edits.

1. All edits are performed on a common original sentence.

2. Each edited area consists of a section of text to be deleted (may be empty) and a substitute text, e.g., *"Newton lost ~~hisself~~himself in"* equals to *"Newton lost himself~~hisself~~ in"*; i.e., all edits can be considered as substitutions.

3. Edited areas can be extended on both ends, e.g., *"Newton lost himself~~hisself~~ in"* equals to *"Newton lost himself~~lost~~*

~~*hisself*~~ *in*”; and hence they can be aligned to related areas in other parallel revisions.

4. Adjacent edits can be joined together, e.g., *“Newton devoted~~lost~~ himself~~hisself~~ to~~in~~”* equals to *“Newton devoted himself to~~lost hisself in~~”*.

The algorithm as described above can also be used for sentence merging, which extracts edits from one version and patches them into another co-author’s version. The sentence that the co-author would like to adopt can be either a sentence edited by another co-author or an automatically summarized one. The result of a sentence merging is exactly the summary of it and the co-author’s own sentence. Once conflicts occur, the co-author decides, for each conflicted part, which one to retain. In this way, sentence merging enables authors to combine good, local edits to create text with higher quality.

USER STUDY

We conducted a preliminary user study with 12 Chinese-native speakers who used English as a foreign language. The goal of the study was to examine the effectiveness of CEPT for improving editing quality in collaborative proofreading, compared with a traditional interface without any assistance of aggregation, visualization, or interaction of edits. The study also sought to investigate the editing quality of a group of L2 collaborators versus native speakers.

Participants were given two paragraphs written by non-native speakers (NNSs) and asked to proofread them, one with CEPT and the other with a baseline interface. The task was to help improve the overall language quality, e.g., correcting grammar errors, enhancing word choices, and refining sentence structures. In order to quantitatively compare edit quality, we controlled the flow of collaboration by preparing three revisions in advance and asking each participant to collaborate with the same three mocked co-authors. We measured participants’ time spent, improvement of editing quality, interaction patterns, and subjective feedback.

The Baseline Interface

The baseline interface was similar to CEPT but without the novel features of sentence mapping, edit aggregation and interaction. As shown in Figure 7, it used the same editor with text tracking as in CEPT. All edits were presented in diffing mode and authors could toggle the visibility of deleted text. However, the baseline interface differed from CEPT in how co-authors’ revisions were presented and whether authors could interact with these revisions: it simply stacked the parallel revisions given by collaborators besides the editor, so as to simulate how an author could currently handle multiple document versions returned by collaborators.

Tasks

For the editing tasks, we exploited the dataset of learner texts created by Bryant et al. [8]. In their study, they collected 50 essays written by NNS college students and had ten native raters (N1-N10) correct each essay and categorize writing errors in the schema of the NUCLE corpus [11]. The dataset provided us with a reasonable ground truth for evaluating editing quality quantitatively. From the dataset, we extracted

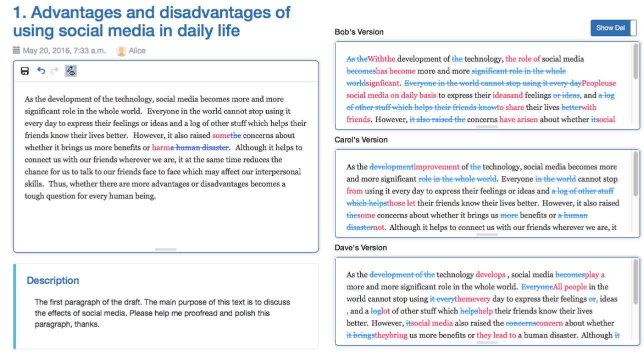

Figure 7. The baseline interface, which just shows the entire revisions of all co-authors without the aggregation or interaction features of CEPT.

two paragraphs on the topic of social media and life to create two proofreading tasks, each with about 120 words in five or seven sentences. We also extracted two shorter paragraphs consisting of two sentences as warm-up tasks.

Before the study, the tasks were first completed independently by three revisers (C1-C3) in order to create three mocked co-author’s revisions. The three revisers were also NNSs and with similar English proficiency and experience of collaborative editing as the study participants. Therefore, the quality of these prepared revisions was still far from that of native speakers, leaving space for further improvement in the subsequent study.

Participants

Twelve graduate students with an engineering background were recruited for our study (7 female). The subjects aged from 23 to 34 (*mean* = 25.8, *sd* = 2.8). They were all Chinese-native with self-evaluated intermediate English proficiency, and all had been learning and using English for years. Specifically, they all had experience in reading and writing English academic papers, and 11 had primarily authored at least one English paper (*mean* = 2.3, *sd* = 1.1).

In terms of familiarity with collaborative editing, most of them had experienced collaboration in research and academic writing prior to our study (9 out of 12). Of the 9 participants, all reported having used e-mail to share document copies among collaborators and revising in edit mode most of the time. Other tools for collaboration mentioned included instant messaging (3), Google Docs (1), and SVN (1). Moreover, all participants knew and felt comfortable with C1-C3 for they were colleagues that work in the same department.

Design and Procedure

The experiment was a mixed between- and within-subjects design. Each participant used both interfaces (*CEPT* vs. *baseline*) and completed both proofreading tasks, but was not exposed to all combinations. The interface assigned and the order of the tasks were fully counterbalanced.

Participants first followed a 15-minute warm-up session to get familiar with both interfaces. Then in the main session, participants acted as one of the co-authors besides C1-C3 to proofread the two paragraphs with the help of the three existing revisions. During the study, they were instructed to try their best to improve the language quality in terms of grammar, vocabulary use, sentence structure, etc. There was no time limit on each task but we suggested about 15 minutes for each paragraph. Participants were also allowed to refer to online dictionaries and other resources as they needed. Finally, they filled out a survey questionnaire asking about their preferences and feelings about the two interfaces with a 7-point Likert scale as well as free-form comments.

RESULTS

In this section, we present several findings on edit time and quality, user experience, and interaction patterns, illustrated by both qualitative and quantitative results.

Edit Time, Amount, and Quality

Edit time

On average, participant spent about 17 minutes (sd = 5.3) on revising each paragraph. No significant difference was found in edit time between the two interface conditions, indicating that participants devoted themselves to both tasks. Participants' interaction with edits by co-authors was interleaved with other activities such as initial reading, self-editing, and post-edit rereading, yet no quantitative conclusion can be drawn on whether CEPT could save authors' time and effort to locate and understand edits by co-authors. However, in subjective scores participants expressed a strong preference for CEPT over the baseline in several aspects involving efficiency, which we will discuss later.

Edit amount

We used the percentage of words deleted and added to measure participants' edit amount. On average, participants deleted 32% words from the old text (sd = 14.3%) and added 21% new words (sd = 9.6%). Again, no significant difference was found in edit amount between the two interfaces. For comparison, the revisions by N1-N10 altogether showed that 36% of all words were in need of improvement.

Edit quality

Edit quality was measured by an average number of remaining errors per sentence. This was done by a native speaker (A1) who annotated remaining errors for each revised sentence, as well as those formerly given by N1-N10 and by C1-C3, according to the NUCLE schema [11]. We counted the sum of error tags in a paragraph and calculated error rate as the average number of errors left in a sentence after editing.

We first consider the effect of interface design on edit quality. On average, the error rates after editing with CEPT and the baseline interface were 0.76 (sd = 0.38) and 0.93 (sd = 0.38) respectively, i.e., CEPT lowered the average errors per sentence by 18% compared with the baseline interface. A paired t-test showed that there was a significant difference between error rates of the two interfaces ($t_{11} = -2.54$, $p < .05$).

Then we compare the edit quality of L2 co-authors with native speakers. The mean error rate of the two tasks before and after being corrected by N1-N10 was 2.4 and 0.08 respectively, which meant that there were almost no errors left. In comparison, the mean error rate of the study participants was 0.8 after the task. The average edit quality of L2 co-authors was still not as good as that of native speakers in terms of language accuracy. However, we also asked A1 to pick the best revision for each of the 12 sentences. Results showed that, in 9 out of 12 cases, participants' revisions were considered of higher quality than those of native speakers.

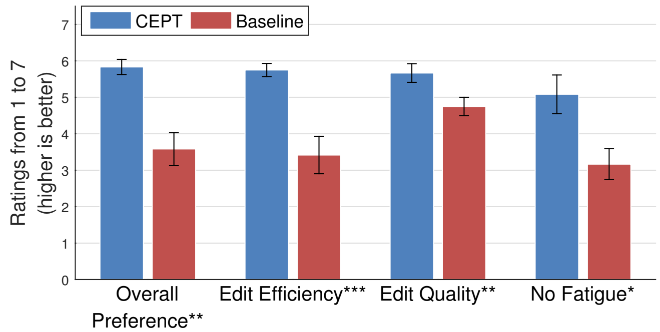

Figure 8. Subjective ratings for both interfaces, with standard error bars and results of Wilcoxon signed-rank tests. (*: p <.05, **: p <.01, ***: p <.001)

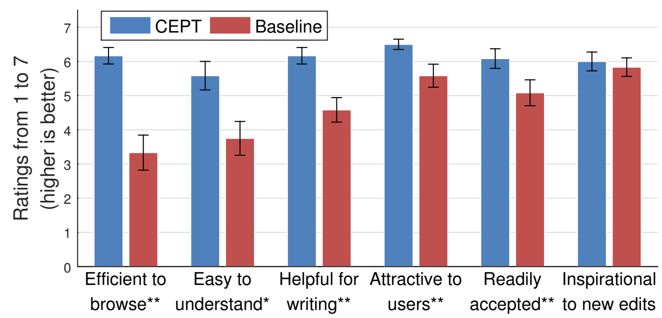

Figure 9. Subjective feelings about edits by others in both interfaces, with standard error bars and results of Wilcoxon signed-rank tests. (*: p <.05, **: p <.01, ***: p <.001)

The above results show that CEPT can improve language edit quality. Furthermore, collaborative editing by several NNSs may produce text that is close to, or even better than that of native speakers in term of language quality.

Subjective Feedback

Interface preference

As shown in Figure 8, participants preferred CEPT to the baseline interface overall and in several detailed aspects. We ran Wilcoxon signed-rank tests on the 7-point scale ratings. Results revealed that participants generally preferred CEPT to the baseline interface for collaborative editing. Specifically, they felt that CEPT was superior to the baseline interface in terms of both efficiency and quality of editing, and that collaborative editing using CEPT was less tiring. In free-form comments, participants expressed that they strongly liked using CEPT for its satisfying editing experience. The aligned sentences and the summary view were clear enough for them to understand what happened to the texts. In contrast, the

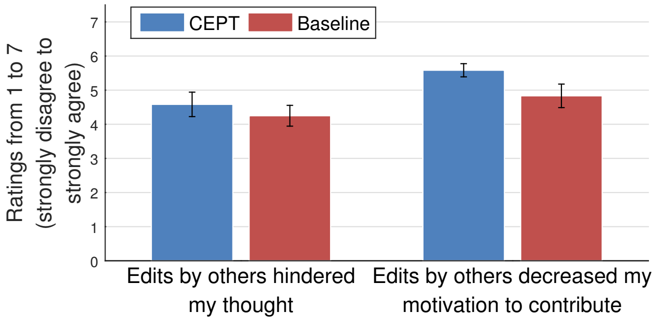

Figure 10. Subjective feedback on the negative impacts of other co-authors for both interfaces, with standard error bars. No significant difference was found by Wilcoxon signed-rank tests.

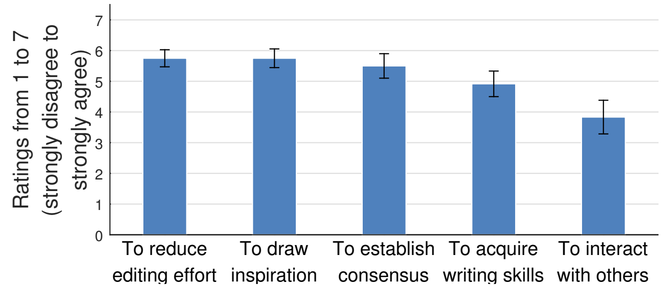

Figure 11. Reasons why an author was motivated to view edits by others, with standard error bars, ordered by participants' mean ratings.

baseline interface was thought to be poor and monotonous. Participants took great pains to locate and borrow ideas from co-authors' revision.

Effect of interface on collaboration

Participants' attitudes towards the helpfulness of co-authors also varied with interface condition (Figure 9). With CEPT, participants were significantly more willing to view and adopt edits made by other co-authors, possibly because they are more efficient to browse and easier to understand. Recall that, as revealed by our pilot study, easily following co-authors' edits was key to effective knowledge sharing in collaborative editing, and the interface design of CEPT meets this criterion. In addition, the power of revisions by others to inspire new edits was perceived at the same high level for both interfaces, indicating the advantage of collaborative editing for L2 authors. The result is in accordance with the lesson learned from the pilot study that asynchronous, multi-versioned collaborative editing is helpful for improving language quality and is accepted by L2 authors.

Potential negative impact of awareness

Moreover, to figure out whether presenting co-authors' edits will have a potentially negative impact on participants' concentration and initiative, we also asked participants about their perception of distraction and motivation in the presence of multiple edits by others in each task condition. The results are shown in Figure 10, and no significant difference was found. With awareness of edits by three co-authors, participants felt a moderate level of distraction, but a slightly higher level of decline in motivation. An improved interface design that eliminates this decline in motivation might further improve the language quality for L2 collaborators.

Usage Statistics of CEPT

We are also interested in participants' use of several key features provided by CEPT, including browsing co-authors' summarized and detailed edits sentence by sentence; borrowing others' edits in the author's own text; toggling the visibility of context and deleted text; and voting and commenting on each sentence version edited by others.

Intensively used functionality

Overall, each participant hovered 21 times (sd = 20.0) on conflicted texts in the summary view to browse alternatives and directly accepted others' sentences 4 times (sd = 3.7). We used hover events to estimate authors' frequency of interaction with edits given by co-authors, as there was no explicit log when an author was viewing co-authors' sentences. Note that CEPT was used in half of the tasks, i.e., 6 sentences. We therefore conclude that CEPT users view and adopt co-authors' revisions frequently. In particular, we also found that 70% of the acceptances were related to the automatically summarized sentence. The statistics above indicate that CEPT users prefer to read and refer to the summarized sentences. Usage statistics of individual participants varied due to different strategies they adopted in the study, which we will discuss later.

Less-used functionality

Other features were used less by participants. Firstly, five out of the 12 participants switched to show or hide deleted texts (min = once, max = 4 times). Participants felt satisfied with the default design of hiding deleted text, as the deleted content shown in strike-through placed extra burdens of recognizing the final text. Similarly, only three of them toggled the visibility of context sentences (min = once, max = 4 times). We infer that authors can get used to the aligned sentence list view without extra surrounding context. This is possibly due to the fact that the task we focused on in our research was language editing and polishing, where many problems on grammar; wording and phrasing; and sentence structure can be identified and corrected within the context of a sentence. Finally, commenting and voting were seldom used, which was expected in our study as there was no strong need and motivation for participants to discuss with or explain ideas to the mock co-authors. Besides, participants stated that they paid more attention to the summarized sentence and hence were exposed less to the functionality of commenting and voting on individual's revisions, which was another possible reason that this functionality was used less.

DISCUSSION

Strategy of Collaborative Editing

We learned from participants' free-form feedback that different editing strategies were adopted when using CEPT and the baseline interface.

The experience of the baseline interface is more like an individual editing process for co-authors. Eleven out of the 12 participants expressed that when using the baseline interface, they read through the text first and tried to edit by themselves. On encounters of potentially incorrect areas, they pondered and proposed tentative corrections by themselves. Then they searched for the same areas in co-authors' revisions to check if consensus was reached. When participants found corrections that were different from their own ones, they tried to figure out the best candidate through a longer period of thinking or referring to external resources, and manually replaced their own text with the best one.

In CEPT, co-authors behaved differently from the baseline interface in that almost all participants (11 out of 12) quickly browsed co-authors' revisions first. Particularly, half of them mentioned that they browsed the summary view with the highest priority. One participant felt that the summary view was so appealing and especially powerful at locating errors and gathering correction suggestions that she could not help referring to it. Overall, CEPT users interact and ponder more on edits of collaborators, and they are more willing to accept them for the feasibility of the one-click borrowing.

In addition, we learned from the subjective feedback that users of the baseline interface tend to view others' revisions less in order to lower their cognitive load and focus on their own text. Several participants admitted that they only viewed parts of revisions and ignored others. In contrast, the sentence summary view of CEPT encouraged co-authors to consider all edits provided by others. Co-authors' different strategies and mental state when using the two interfaces may explain why there was no evident time saving when using CEPT.

Effect of Collaborative Editing

Despite that participants adopted different strategies in different interfaces, they rated the level of inspiration received from co-authors' edits as equally high (Figure 9). To closely examine the effect of collaborative editing compared with editing alone, Figure 11 shows subjective ratings on several reasons that motivated participants to view edits by other co-authors in collaborative editing. The top-rated reasons were to reduce effort in editing by themselves and to get inspiration from the edits by others. Furthermore, in response to free-form questions, participants expressed great engagement with edits by peers whether they considered their roles in collaborative editing to be authors or co-authors. They identified several benefits brought by collaborative polishing. Peer edits could 1) help them better understand the original text; 2) effectively facilitate the process of identifying errors; 3) "open their mind" and inspire new ideas about how to improve the text; and 4) help to learn writing by providing multiple alternatives of a similar meaning.

Compared with the baseline interface, CEPT provided additional means for authors to effectively collect language knowledge from co-authors, i.e., summarizing edits in different revisions and borrowing satisfying edits from co-authors. Participants mentioned that CEPT enabled them to work on the current "best" revision: they first recognized the best version in their opinion, borrowed the edits into their own text, and made further improvements. Thus, it suggests that sharing language knowledge in CEPT is more direct and effective, helping authors to make the best use out of co-authors' collective intelligence. Edit quality can be expected to be further improved if the above process could iterate over a longer period of time (instead of just one iteration in our lab study).

Effect for Learning

Apart from the benefits for edit quality, participants also found that they were able to gain writing knowledge from the inspirational candidates given by co-authors which they themselves might never come up with. This is in line with previous research on L2 education where collaborative writing has been adopted to facilitate teachers' feedback to students as well as students' peer-learning to promote writing skills (e.g., [9] and [40]). Although promising, testing CEPT's effect for language learning requires a dedicated study to examine co-authors' short- and long-term memory of the knowledge gained through collaborative editing, which is beyond the scope of this paper. It is a future direction to consider and evaluate CEPT as a collaborative tool for language learning. For example, co-authors or instructors can help revise language and use comments to explain their edits for teaching purposes.

The Value of Edits

Edits received from collaborators are of great value for improving text quality and for language learning. In our study, we have observed a number of examples in which authors incorporated writing knowledge from each other. Though L2 authors find it difficult to efficiently produce error-free and fluent text, they can still recognize better versions of text from multiple different revisions.

Unfortunately, in current editing practices, edits are largely abandoned once the writing task is done, whether they were proposed by collaborators or made on the author's own. In our study, participants also described the dilemma where it was hard for them to remember the corrections previously received when writing the next article, no matter how helpful these corrections used to be. Thus, we highlight the need for effective means to record, reuse, and share these valuable edits in the long run. Edits made by human revisers could be stored and used to enhance machines' capability of text polishing in the future, as suggested in [18].

Moreover, behind each correction is an author's complex mental process. Why does an author make such a correction? What information resources have they referred to in making the correction? And at what level is the author confident in the correction? Making collaborators aware of these processes behind the revisions will further help them derive insights, learn more from others' edits, and reach consensuses more effectively, which has also been studied in previous research [6, 33].

Limitations and Future Work

Our research has several limitations, which will be our focus in future work.

First, CEPT extracted edits based on aligned sentences between two revisions and a word-level diff-ing algorithm. Subject to these approaches, CEPT failed to identify and visualize movements of text. Furthermore, CEPT did not support deriving higher-level edits such as sentence structure changes and sentence reordering in a paragraph.

Second, the sentence summary view of CEPT still has room for improvement. We treated each edited area to be independent, which may lead to inconsistent combinations of corrections. For example, the summary of "*one of the greatest inventors*" and "*the greatest inventor*" might result in candidates like "*one of the greatest inventor*", which is grammatically wrong. The order of candidates for a conflicted area can also be optimized if we take more information into consideration, such as the author's English proficiency, co-authors' voting, etc.

Finally, our user study was a preliminary one. In the lab study, the collaborators were controlled to be fixed and mocked, causing the flow of collaboration to be one directional. While this enabled us to quantitatively compare edit quality of the two interfaces across participants, real-world co-authors' interaction and collaboration patterns had to be left for future work. In the future, we will carry out a field study to observe CEPT's effectiveness on practical and bi-directional collaborative editing by groups of NNS authors. Improvement of our quality measure is also needed in that we mainly focused on local errors and only invited one native rater to evaluate the revisions. Whether CEPT can improve NNSs' language at higher levels such as discourse and coherence remains to be investigated.

CONCLUSION

In this paper, we propose to improve language quality of non-native writing through collaborative editing by a group of NNS co-authors. We present CEPT, a novel collaborative editing system that supports language knowledge sharing among NNS co-authors by aggregating edits of all co-authors and presenting them in an easy-to-understand way. CEPT has three highlighted novel features: cross-version sentence mapping, edit summarization, and an interface enabling co-authors to examine, comment on, and borrow edits of others. With an initial user study, we showed that CEPT could reduce 18% of errors compared with a baseline interface. Moreover, co-authors preferred CEPT for its ease of use and the benefits it brought for sharing and gathering language knowledge from co-authors. We believe that CEPT is the first step to fulfilling the collective intelligence of NNS authors for improved writing quality.

ACKNOWLEDGMENTS

This paper was supported by Natural Science Foundation of China under Grant No. 61272230 and Microsoft Research Asia Collaborative Research Grant No. FY15-RES-THEME-018. We thank our participants for their time and work. We also thank Xiaojuan Ma and Xianghua Ding and all the reviewers for their constructive suggestions.

REFERENCES

1. Ronald M Baecker, Geof Glass, Alex Mitchell, and Ilona Posner. 1994. SASSE: The collaborative editor. In *Conference companion on Human factors in computing systems*. ACM, 459–462.
2. Ronald M Baecker, Dimitrios Nastos, Ilona R Posner, and Kelly L Mawby. 1993. The user-centered iterative design of collaborative writing software. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 399–405.
3. Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics* 1 (2013), 391–402.
4. Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 313–322.
5. Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2015. Soylent: a word processor with a crowd inside. *Commun. ACM* 58, 8 (2015), 85–94.
6. Jeremy Birnholtz and Steven Ibara. 2012. Tracking changes in collaborative writing: edits, visibility and group maintenance. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 809–818.
7. Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. 2014. Divide and correct: using clusters to grade short answers at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 89–98.
8. Christopher Bryant and Hwee Tou Ng. 2015. How Far are We from Fully Automatic High Quality Grammatical Error Correction?. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 697–707.
9. Rafael A Calvo, Stephen T O'Rourke, Janet Jones, Kalina Yacef, and Peter Reimann. 2011. Collaborative writing support tools on the cloud. *IEEE Transactions on Learning Technologies* 4, 1 (2011), 88–97.
10. Daniel Dahlmeier and Hwee Tou Ng. 2011. Correcting semantic collocation errors with L1-induced paraphrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 107–117.

11. Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. 22–31.

12. Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. 2015. CourseMIRROR: Enhancing large classroom instructor-student interactions via mobile interfaces and natural language processing. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1473–1478.

13. Robert S Fish, Robert E Kraut, and Mary DP Leland. 1988. Quilt: a collaborative tool for cooperative writing. In *ACM SIGOIS Bulletin*, Vol. 9. ACM, 30–37.

14. Peter Kin-Fong Fong and Robert P Biuk-Aghai. 2010. What did they do? Deriving high-level edit histories in wikis. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*. ACM, 2.

15. Sebastian Gehrmann, Lauren Urke, Ofra Amir, and Barbara J Grosz. 2015. Deploying AI Methods to Support Collaborative Writing: a Preliminary Investigation. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 917–922.

16. Elena L Glassman, Lyla Fischer, Jeremy Scott, and Robert C Miller. 2015a. Foobaz: Variable Name Feedback for Student Code at Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 609–617.

17. Elena L Glassman, Jeremy Scott, Rishabh Singh, Philip J Guo, and Robert C Miller. 2015b. OverCode: Visualizing variation in student solutions to programming problems at scale. *ACM Transactions on Computer-Human Interaction (TOCHI)* 22, 2 (2015), 7.

18. Nick Greer, Jaime Teevan, and Shamsi Iqbal. 2016. *An Introduction to Technological Support for Writing*. Technical Report MSR-TR-2016-1.

19. Joy Kim, Justin Cheng, and Michael S Bernstein. 2014. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 745–755.

20. David Lamas, Vladimir Tomberg, and Mart Laanpere. 2012. A conceptual model for collaborative scientific writing. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. ACM, 151–154.

21. Jin-woo Lee, Joohyun Kim, Uichin Lee, and Jae-Gil Lee. 2015. SocialKeyboard: Proofreading Everyday Writings in Mobile Phones. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1591–1596.

22. Mary DP Leland, Robert S Fish, and Robert E Kraut. 1988. Collaborative document production using Quilt. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. ACM, 206–215.

23. Eugene W Myers. 1986. AnO (ND) difference algorithm and its variations. *Algorithmica* 1, 1-4 (1986), 251–266.

24. Michael Nebeling, Alexandra To, Anhong Guo, Adrian A. de Freitas, Jaime Teevan, Steven P. Dow, and Jeffrey P. Bigham. 2016. WearWrite: Crowd-Assisted Writing from Smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 3834–3846.

25. Brice Nédelec, Pascal Molli, Achour Mostefaoui, and Emmanuel Desmontils. 2013. LSEQ: an adaptive structure for sequences in distributed collaborative editing. In *Proceedings of the 2013 ACM symposium on Document engineering*. ACM, 37–46.

26. Christine M Neuwirth, Ravinder Chandhok, David S Kaufer, Paul Erion, James Morris, and Dale Miller. 1992. Flexible diff-ing in a collaborative writing system. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM, 147–154.

27. Christine M Neuwirth, David S Kaufer, Ravinder Chandhok, and James H Morris. 1990. Issues in the design of computer support for co-authoring and commenting. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*. ACM, 183–195.

28. Christine M Neuwirth, David S Kaufer, Ravinder Chandhok, and James H Morris. 1994. Computer support for distributed collaborative writing: defining parameters of interaction. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 145–152.

29. Sylvie Noël and Jean-Marc Robert. 2004. Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work (CSCW)* 13, 1 (2004), 63–89.

30. François Pacull, Alain Sandoz, and André Schiper. 1994. Duplex: A distributed collaborative editing environment in large scale. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 165–173.

31. Taehyun Park, Edward Lank, Pascal Poupart, and Michael Terry. 2008. Is the sky pure today? AwkChecker: an assistive tool for detecting and correcting collocation errors. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. ACM, 121–130.

32. Jean Piaget. 1968. *On the development of memory and identity*. Vol. 2. Clark University Press.

33. Vilaythong Southavilay, Kalina Yacef, Peter Reimann, and Rafael A Calvo. 2013. Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM, 38–47.

34. William Strunk. 2007. *The elements of style*. Penguin.

35. Jaime Teevan, Shamsi T. Iqbal, and Curtis von Veh. 2016. Supporting Collaborative Writing with Microtasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 2657–2668.

36. Dakuo Wang. 2016. Exploring and Supporting Today's Collaborative Writing. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 255–259.

37. Dakuo Wang, Judith S Olson, Jingwen Zhang, Trung Nguyen, and Gary M Olson. 2015. DocuViz: Visualizing Collaborative Writing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1865–1874.

38. Paula M Winke. 2007. The psychology of the language learner: Individual differences in second language acquisition. *Studies in Second Language Acquisition* 29, 01 (2007), 143–144.

39. Huichao Xue and Rebecca Hwa. 2014. Improved Correction Detection in Revised ESL Sentences.. In *ACL (2)*. 599–604.

40. Soobin Yim, Mark Warschauer, Binbin Zheng, and Joshua F Lawrence. 2014. Cloud-Based Collaborative Writing and the Common Core Standards. *Journal of Adolescent & Adult Literacy* 58, 3 (2014), 243–254.