IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 19, 2020, accepted March 5, 2020, date of publication March 23, 2020, date of current version April 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982639

Design and Evaluation of WriteBetter: A Corpus-Based Writing Assistant

ALESSIO BELLINO[1] AND DANIELA BASCUÑÁN[2]
[1]Facultad de Ingenieria y Ciencias, Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago 8370109, Chile
[2]WriteBetter, Santiago 8320107, Chile

Corresponding author: Alessio Bellino (alessio.bellino@mail.udp.cl)

ABSTRACT Corpus-based writing assistants are aimed to show how words are used in real context: they provide word use examples from which users can (1) draw inspiration for their writing and (2) understand how words are used together to improve their own writing. Although the idea of integrating a corpus-based writing assistant (concordancer) into word processors is not new, their integration is designed to be not as straightforward as writing in a word processor. In this paper, we present WriteBetter, a corpus-based writing assistant designed to be integrated into Microsoft Word, Google Docs, and Overleaf. This integration makes its use straightforward and easy as users can see corpus-based examples (1) in real-time while writing in the word processor or (2) just selecting a piece of text in their document. This facilitates user-corpus interaction as the required user's interaction is minimal. After contextualising the state of the art regarding the benefits of corpus-consultation, we discuss the design features of WriteBetter that make it novel in relation to other tools. Next, we present a user evaluation of the first version of WriteBetter, which was carried out on 11 undergraduate students of a Chilean university, who were asked to trial the software while writing in English. Based on this evaluation, we designed a new version of WriteBetter, which was further evaluated online on 36 users. WriteBetter is now available for everyone as SaaS.

INDEX TERMS Computer-assisted writing, concordancer, corpus-assisted writing, user interface design, user evaluation.

I. INTRODUCTION

Academic writing is a challenge for many people. That is why the market is full of writing assistants aimed at providing feedback to improve writing. Most of them (e.g. Grammarly, Ginger) are designed to improve spelling, grammar and semantic choices, among others. A minor class of writing assistants (e.g., WriteFull, Ludwig), in contrast, provide corpus-based examples according to the word searched. Based on these examples, users can (1) draw inspiration for their writing and (2) understand how words are used together to improve their own writing. Corpus-based tools are built on the concept of concordance: searching a word/chunk of text, sentences containing that word/chunk of text are displayed surrounded by their co-text. In contrast to the most traditional writing assistants, a corpus-based writing assistant requires users to make an additional effort since different examples must be explored to select the most relevant ones that fit what users are writing. Research has demonstrated that, despite the additional effort, this approach has a favourable effect: corpus consultation is particularly effective for learning scopes [1] and improving writing [2]–[4]. Moreover, the examples provided can be helpful to draw inspiration to complete ideas [5].

In this paper, we present WriteBetter, a corpus-based writing assistant designed to be seamlessly integrated into the word processor (Microsoft Word, Google Docs, and Overleaf until now). Such integration makes the use of WriteBetter straightforward and easy as users can see corpus-based examples without leaving the writing environment.

The design of WriteBetter aims to overcome some drawbacks of current concordancers. Particularly, research-oriented concordancers such as AntConc and COCA work in a separate window from the word processor without any integration (Fig. 1). Therefore, users must switch application to get suggestions.

Other commercial concordancers, designed to be more user-friendly such as WriteFull and Ludwig, are partially integrated into the word processor. After selecting a text, users can open the concordancer using a keyboard shortcut.

The associate editor coordinating the review of this manuscript and approving it for publication was Waleed Alsabhan.

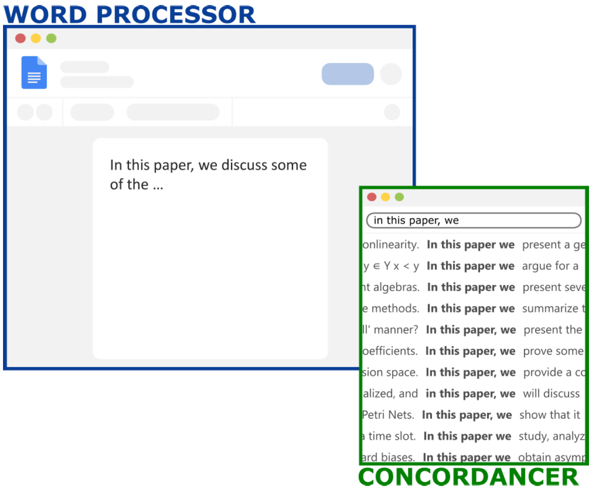

FIGURE 1. **Traditional concordancers are not integrated into the word processor. The word processor and concordancer are displayed in two separate windows. To get suggestions, users must change the window and write their queries manually in the concordancer.**

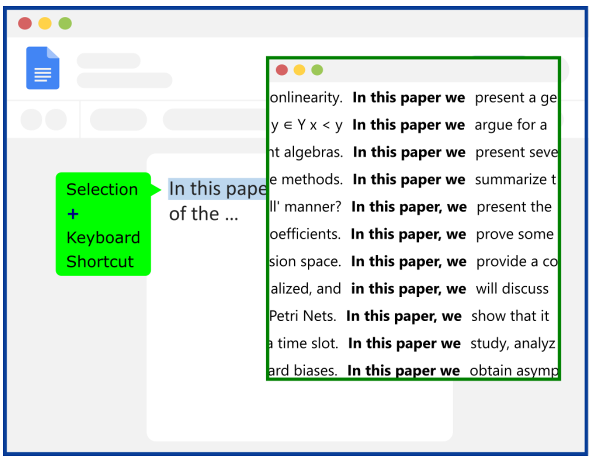

FIGURE 2. **In more user-friendly software such as WriteFull and Ludwig, the user must select a piece of text and press a keyboard shortcut to open the concordancer as a popup window: there is no full integration into the word processor.**

However, suggestions are displayed over the document in a popup window (Fig. 2).

WriteBetter is fully-integrated into the word processor and this facilitates user-corpus interaction in comparison to the concordancers mentioned above. Particularly, interaction can occur in two modalities: (1) selecting a piece of text (Fig. 3A) and (2) without explicit action of the user (Fig. 3B). In the former case, users can select a chunk of text and get suggestions - without any need of keyboard shortcut as required by WriteFull and Ludwig. In the latter case, suggestions are displayed in real-time while writing since they are dynamically linked to what users type into the word editor, unlike WriteFull and Ludwig.

Considering the novelty of our design, it becomes relevant to (1) determine to what extent a corpus-based writing assistant integrated into the word processor can improve the

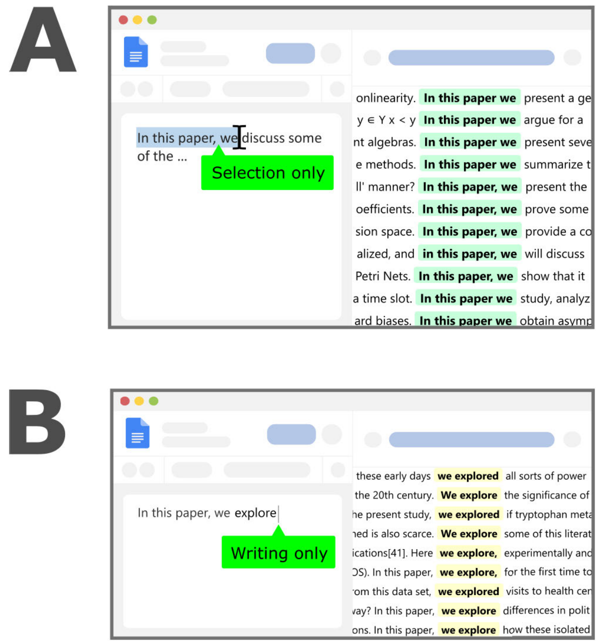

FIGURE 3. **In A, suggestions are displayed just after users select a piece of text, without the need of any keyboard shortcut. In B, suggestions are displayed in real time as the user writes in the word processor. It means that suggestions are dynamically linked to whatever string of characters the user types into the text editor, unlike WriteFull and Ludwig. In any situation, suggestions do not cover the document as they are fully-integrated on the right side of the word processor without any invasive popup window.**

writing performance of users and (2) identify recurrent users' behaviour when approaching the writing environment for the first time. To do so, we carried out two user studies, which are presented below.

Particularly, the paper is organized as follows. In the next section, we present the related work focusing on linguistic aspects and software tools. After, we discuss the design of the first version of WriteBetter. Then, we present the first user study carried out on 11 undergraduate students of a Chilean university - who were asked to trial the software while writing in English. Based on this evaluation, we designed a new version of WriteBetter on which we carried out the second user study to identify recurrent users' behaviour. This study involved 36 users, who tried WriteBetter online. WriteBetter, in fact, is currently available online as SaaS for everyone.

II. RELATED WORK

A. LINGUISTIC BACKGROUND

A corpus is a collection of texts upon which linguistic analysis can be conducted [6]. To explore a corpus, software tools – technically known as concordancers – are needed. Concordancers allow learners to be exposed to rich, vast and diverse linguistic contexts, which foster and consolidate language knowledge. In this regard, it is argued that "no dictionary or grammar is able to fully describe the language" [7], and that "there is probably no substitute for having first-hand access to a corpus and concordancer" [8].

Almost 30 years ago, Johns [9] coined the concept of Data-driven Learning (DDL), which refers to the learner sorting through large amounts of authentic language data to figure out patterns or rules of language use. The essential components in DDL are corpora and concordancers to query them, and its basic idea is to allow massive exposure to linguistic data that is still organised and focused [10]. In DDL, concordancers usually present the learner's query surrounded by their different co-texts by using the Key Words in Context (KWIC) format, which encourages learners to analyse authentic language to discover its use.

The literature reports several advantages related to corpus consultation and Data-driven learning (DDL). Although corpus consultation and learning are potentially different [11], when the former leads to effective learning it has the potential to foster learner's autonomy and life-long learning [12]–[18]. As Boulton states, DDL is a constructivist and inductive approach that allows users to "reach their own conclusions that are meaningful to them individually, and the cognitive processing should lead to longer retention than simply being taught" (p.2) [10].

The rich exposure to authentic texts involved in corpus consultation allows users to become aware of different language forms and patterns, which can improve their cognitive and meta-cognitive development and, consequently, language skills and communication abilities. Particularly, O'Sullivan [15] refers to the cognitive activities associated with corpus consultation: "predicting, observing, noticing, thinking, reasoning, analysing, interpreting, reflecting, exploring, making inferences (inductively or deductively), focusing, guessing, comparing, differentiating, theorising, hypothesising, and verifying". Despite the effort required to explore corpora, the active role of users as language researchers is what enhances their skills to discover unknown language patterns, which can potentially expand their language knowledge, and let them improve their writing.

B. SOFTWARE ARTEFACTS FOR CORPUS CONSULTATION

Many corpus-based software artefacts have been developed over the years. Some of them are proper concordancers whereas others just use corpora to provide suggestions, but direct user-corpus interaction is not possible. In Table 1, we summarised some relevant corpus-based tools distinguishing between concordancers and non-concordancers. Even though the idea of integrating a concordancer into word processors is quite old [19], some efforts to integrate concordancers into the word processor have been made only in the last years.

The first generation of concordancers has been usually developed as standalone applications like AntConc [20], or web-based applications like COCA [21]. Nevertheless, this lack of integration could be unfavourable for learners during the writing process. In fact, "writers who have to stop writing to look up a word can be distracted from getting their ideas down on paper" [5].

The second generation of concordancers – e.g., SketchEngine [22], WriteFull[1] and Ludwig[2] – tackled this issue by allowing some forms of integration. Sketch Engine can be integrated into SDL Trados, which is a tool for translators only, and therefore it is not suitable for normal writing activities. However, WriteFull and Ludwig are tools that allow some form of integration into the word processor.

Ludwig can be used in two modalities: online – without any form of integration – or by means of an application for Windows and MacOs. The latter allows users to search for a word/chunk of text from any word editor by selecting a piece of text and using a keyboard shortcut (Ctrl+Shift+L). WriteFull, in contrast, offers an add-in for Chrome – without any form of integration – or an application for Windows, Linux and MacOs that, as Ludwig, allows users to search for a word/chunk of text by selecting it and using a keyboard shortcut (Ctrl+Space). Therefore, both Ludwig and WriteFull are not fully integrated into the same UI of the word processor since they are displayed as a popup window and, more importantly, they do not provide real-time suggestions while writing.

In terms of design, we can differentiate WriteBetter from the aforementioned tools in two dimensions: (1) integration into the word processor and (2) real-time suggestions, as displayed in Table 1. To the best of our knowledge, in fact, WriteBetter is the first concordancer designed to be seamlessly integrated into the UI of word processors so that it can provide real-time suggestions while writing. Moreover, our form of integration allows WriteBetter to be aware of the entire co-text of the selected text and, accordingly, can display advanced suggestions such as advice to replace a word, or examples to complete a sentence. Section V displays WriteBetter design and features in detail.

In regard to corpora-based, but non-concordancer tools, researchers have also developed some integrated solutions such as ColloCaid [5], [23], WriteAway [24] and Write Assistant [25]. The first was envisioned to provide feedback on collocations in real time. WriteAway (former WriteAhead) is a web-based writing environment that provides suggestions based on grammar patterns and examples. WriteAway, however, is integrated into a web-based text editor and not into conventional editors more familiar to users (e.g. Microsoft Word, Google Docs). Finally, Write Assistant assists Danish speakers to write in English by means of a bilingual Danish-English dictionary integrated into Microsoft Word. We remark that, even though ColloCaid, WriteAway and Write Assistant are corpus-based tools, they do not allow direct corpus-user interaction as concordancers do.

C. TRAINING AND CORPUS CONSULTATION

Since corpus consultation may be cognitively demanding and requires some effort to be beneficial, it becomes relevant to discover the need of training in corpus consultation activi-

[1] https://writefullapp.com/
[2] https://ludwig.guru/

TABLE 1. Corpus-based tool comparison in terms of design. The tools covered are the ones discussed in the related works, sub-section B.

	Corpus-based tools	Integration into the word processor	Real-time suggestions
NON-CONCORDANCER *	• WriteAway	*NOT INTEGRATED* However, suggestions are displayed in the same place where the user writes, i.e., a web-based text editor.	*AVAILABLE*
NON-CONCORDANCER *	• ColloCaid • Write Assistant	*INTEGRATED*	*AVAILABLE*
CONCORDANCER	• Antconc • Coca • SketchEngine⁺	*NO INTEGRATION* Users need to change window to access the concordancer.	*NOT AVAILABLE* Users need to write their queries manually in the concordancer.
CONCORDANCER	• WriteFull • Ludwig	*PARTIALLY INTEGRATED* Users need to select a text and press a keyboard shortcut to show suggestions. A popup window - which may cover the document in which users are working on – is used to display suggestions.	*NOT AVAILABLE* Users must perform an explicit action (text selection + keyboard shortcut) to enable suggestions.
CONCORDANCER	• WriteBetter	*TOTALLY INTEGRATED* Users need to select a text - or just write – for the software to display suggestions. No keyboard shortcut is need. Moreover, the concordancer UI is displayed in the same window of the word processor. Particularly, suggestions are displayed on the right side of the screen without covering the document on which users are working.	*AVAILABLE* Suggestions are displayed without explicit action of the user. Just writing is enough. Suggestions are dynamically linked to whatever string of characters the user types into the text editor in real time. See video https://youtu.be/D3is_gMLjDc at 0:40 to see real-time suggestions in action.

* They are corpus-based tools but not concordancers since do not allow direct user-corpus interaction.
⁺ SketchEngine is integrated into SDL Trados, tool for translators and not a word processor.

ties. Unfortunately, it seems there is not a clear agreement among researchers. For example, studies on paper-based corpus materials show that corpus consultation is suitable even for students with a low English level without any training [12], [13]. Quinn [26], conversely, states that learners need to be trained to take advantage of corpus use. According to the author, such training ''involves practising corpus research and referencing skills as well as learning to make data-based generalizations''. Previous studies describe training sessions that vary from ''minimal training'' – referred as indirect use of corpora to familiarise users with the concordancer and raise their DDL awareness [27] – to longer time [3], [28], [29] until requiring some lessons [30], [31]. At any rate, it is not clear if training is needed due to the (low) accessibility/usability of software tools [32] or because corpus use requires learners to manage some knowledge [26].

Considering the concern previously posed, we aim to discover whether WriteBetter requires some training to be used fruitfully. From the usability perspective, we made our best effort to make the use of WriteBetter as simple as possible. First, WriteBetter has a minimal interface, which could help reduce cognitive load and the increase easiness of use [33]. Secondly, the integration into the word processor avoids disruptions in the writing activity and helps convey a sense of familiarity in the user. Generally speaking, users reject the sudden introduction of changes (baby duck syndrome) because of their unfamiliarity [34], [35]. Therefore, it is usually preferable to maintain a familiar design – and the integration of WriteBetter goes in this direction. Third, the installation of WriteBetter is really simple: it can be used online (without the need of installing anything in the computer), in Google Docs and Overleaf just installing a Chrome extension, or in Microsoft Word as online add-in.

To conclude, by improving as much as possible user experience, we aim to discover to what extent training is needed to use WriteBetter effectively.

D. SITUATED COMPUTING AND WriteBetter

The integration of a corpus-based writing assistant into the word processor that displays real-time suggestions deals with the aim to change the way in which users use and consume corpora: where a concordancer comes to the language learner instead of the language learner having to seek a concordancer. We discuss the question by introducing the concept of 'Situated computing' [36], which characterises socio-technical systems that are designed to leverage contexts and situations of use to facilitate interaction in everyday human practice [37]. WriteBetter meets the fundamentals of situated computing since its integration into the word processor allows the system to leverage the context of use – the fact that learners write a text – for providing real-time suggestions in context.

According to Gershman *et al.* [38], situated computing systems should be able to reduce the gap between users' intentions and the actions they must take to achieve them: such gaps are physical, of information, and of awareness. WriteBetter fills these gaps by providing information (suggestions in context) that are physically near the document written by the learner (see Fig. 4) increasing awareness on what they are writing through real-time examples. In this regard, it can be argued that standalone concordancers (e.g. AntConc) create a significant gap between the search of words (users' intentions) and the actions that users must take to search for them, as users are required to switch software and explicitly search for the word of interest in those software tools. Using a concordancer integrated into the word processor, instead, the gap noted above is significantly reduced since suggestions are provided just by writing a word, or selecting a chunk of words, i.e., users are not required to switch software and write their query in a text field to search for it explicitly. Although a step towards situatedness has been undertaken also by some of the developers of some of the software tools described above (e.g., WriteFull), WriteBetter takes situatedness to the next level. In fact, WriteBetter provides (1) real-time suggestion while writing, and (2) the simplest way to search for a word/chunk of text of interest, i.e., just by selecting – without requiring of any keyboard shortcut.

III. DESIGN OF THE FIRST VERSION OF WriteBetter

The first version of WriteBetter provided only the Context function (Dictionary and Translate functions were added later). Fig. 4 shows the first version of WriteBetter integrated into Microsoft Word. This aims at improving usability and attractiveness of the software, as users are less likely to be puzzled when approaching its interface since it is integrated into the word processor and convey a sense of familiarity in the user. Regarding interaction, just writing a word was enough to show suggestions in-context for that word (Fig. 4, A). The suggestions are dynamically linked to whatever strings of characters the user types into the text editor. This feature aims at encouraging users – including the overconfident ones [39] – to take a look at the corpus even when they do not have the intention to do so. This could lead them to explore new language patterns they do not expect to discover.

A more "advanced" interaction required users to select a word/chunk of words to see how they are used together (Fig. 4, B). Moreover, there was a smart search option for the software to display word inflections and derivations (Porter stemmer) (Fig. 4, C).

The corpus used in the first version of WriteBetter was extracted from the featured articles and good articles of Wikipedia (a selection of around 36,000 articles out of a total of around 5.5 million of articles – 93,763,639 total words, 1,111,920 unique words[3]). According to Wikipedia grading scheme,[4] they represent the best of Wikipedia in terms of writing quality. As a matter of fact, these kinds of articles are reviewed by several users of the Wikipedia community to check errors. We chose this source to create our corpus because it covers many fields of human knowledge and the quality of writing is quite reliable. The corpus was extracted from Wikipedia XML sources. We removed all but text using WikiExtractor.[5] The text was further processed deleting all those paragraphs that do not start with a capital letter and finish with a full stop. Then, paragraphs shorter than 60 characters were deleted. Finally, we also deleted all paragraphs containing two consecutive spaces. In fact, especially in scientific articles, there are symbols between words whose deletion generates a double space, which results in nonsense sentences.

Regarding technical specifications, WriteBetter is developed in.NET as an add-in for Word (2016 or subsequent versions). The text is indexed using SQLite with FTS4 (Full Text Search) Extension[6] (libraries for.NET[7]) in two tables: the first with Porter stemming whereas the latter without any stemming. The former table is used to make the Smart Search function work so that results for word inflections and derivations could be provided. The latter index (without any stemming) is used to find exact matches.

IV. FIRST RESEARCH QUESTION/USER EVALUATION

Considering the design novelty of WriteBetter, we aim to measure its effect in writing production involving students without experience in corpus consultation. To do so, we aim to identify any learner-corpus interaction mediated by WriteBetter and evaluate how, and to what extent, such an interaction changed their writing. The research question is, therefore, the following:

How and to what extent can a word-processor-integrated corpus-based writing assistant improve writing considering an ordinary writing activity of users without experience in corpus consultation?

To answer this question, we carried out a user evaluation on 11 users on the first version of WriteBetter, which also provides us with significant insights to design the subsequent versions of WriteBetter.

A. RATIONALE OF USER EVALUATION AND DATA ANALYSIS

We aim to understand the effect of WriteBetter during writing production. We are not interested in the writing activity per se. Rather, we see writing activity as an opportunity to catch any learner-corpus interaction (use cases) mediated by WriteBetter. Use cases were extracted from the participants' performance. Analysing such cases allowed us to classify them to further determine the linguistic aspects that may arise from corpus consultation, e.g. spelling, collocations, etc.

Broadly speaking, our interest is to understand how WriteBetter can affect corpus-learner interaction (as introduced by Park [2]). Before Park's work, corpora use was mostly investigated using questionnaires, interviews and written reflections [29], [30], log analysis [1], [40] or using control groups to prove the effect of corpus consultation [13]. These ways to collect data might not be suitable to investigate how

[3]Extracted using https://github.com/Bridgeconn/unique-words-counter
[4]https://en.wikipedia.org/wiki/Template:Grading_scheme
[5]https://github.com/attardi/wikiextractor
[6]https://www.sqlite.org/fts3.html
[7]https://system.data.sqlite.org/index.html/doc/trunk/www/news.wiki

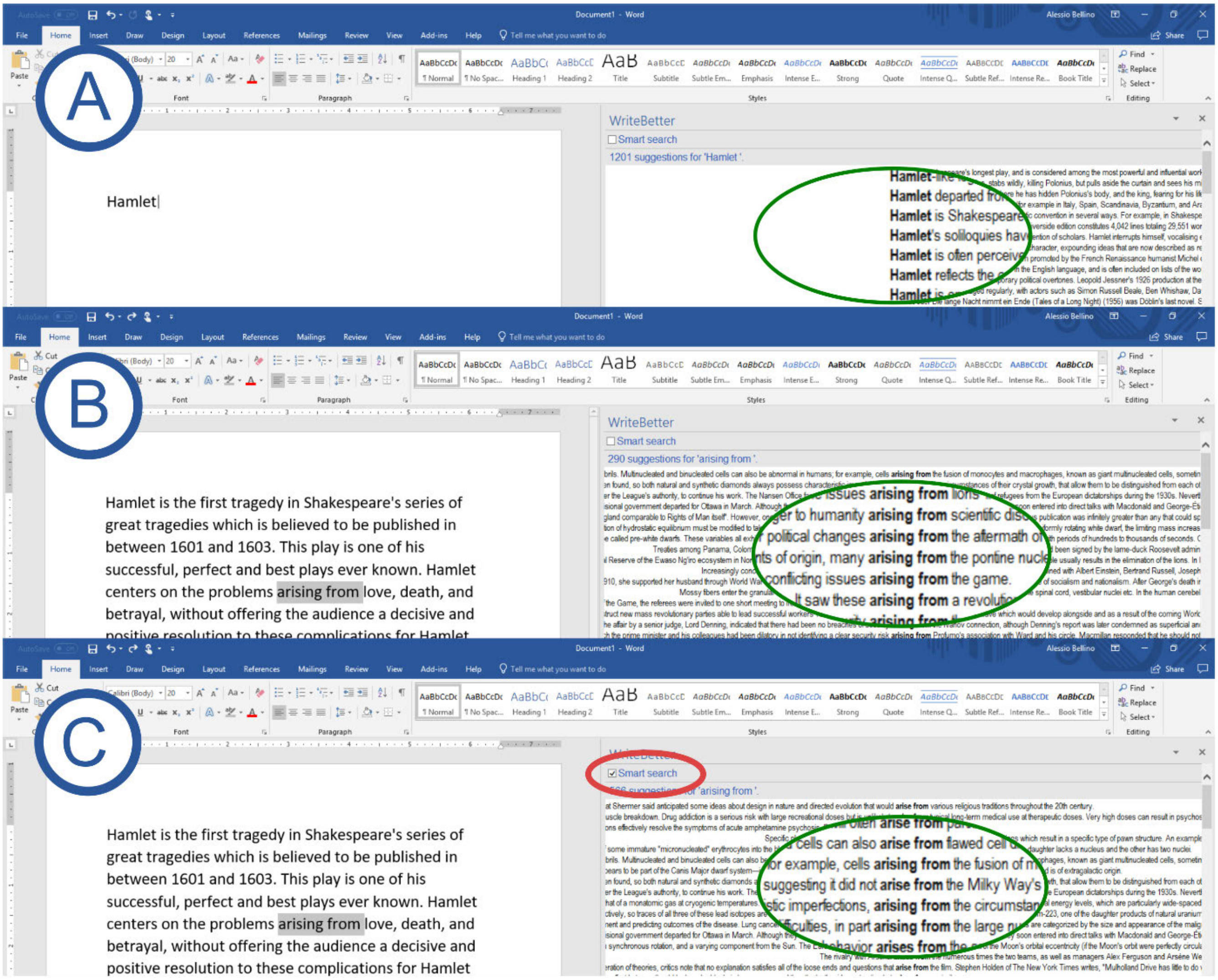

FIGURE 4. The interface of the software on the right. In A, suggestions without selection; in B, suggestions selecting some words; in C, suggestions selecting some word with the Smart Search.

learners interact with corpora since the objectivity regarding the reproduction of the procedures carried out by learners is questionable [2] (e.g. what they change in their writing after corpus consultation). In this regard, Park [2] was a pioneer introducing screen recording – in addition to log analysis – to analyse corpus use cases and provide more objective findings regarding how learners use corpora to change their writing. Later, screen recording and query analysis were also introduced by other researchers [3], [4]. Park [2] and Yoon [3], [4] have shown that the extraction of use cases can be used to check the influence of corpus-learner interaction on writing. Park analysed use cases with the idea that iterative corpus consultation can be considered as a dialogue in which learners "discuss, question, or modify their language". By understanding such a dialogue, Park could identify how learners can improve their writing. C. Yoon, instead, used a more practical approach treating any corpus consultation as a "problem-solving activity the participants engaged in to search for solutions to a range of difficulties encountered during text formulation". This problem-solving process let C. Yoon identify how writing changed and improved.

Therefore, taking inspiration from Park and C. Yoon, we extracted and analysed use cases by using screen recording and consequent query analysis to grasp the learners' interaction process. Moreover, we have complemented screen recording and query analysis with eye tracking. In this regard, we can argue that using screen recording and query analysis only leaves aside what learners look at. By using eye tracking, instead, we were able to reproduce what learners look at to understand more realistically if that reflects, to some extent, changes in their writing. In usability studies, eye tracking lets researchers understand "micro-level behavior such as the focus of attention during a task and distraction" [41], among others. These aspects "usually have little awareness to an individual, and thus are not reported [neither] in the think-aloud protocol". The focus of attention and distraction are important aspects of our study since we aim to understand what sentences learners observe and how this observation reflects changes in writing. Ocular observation is an innate behaviour that requires eye movements (saccades) and fixations toward the target of interest (i.e., the examples displayed by a concordancer), and this information can be provided only by the eye tracker.

We also point out that, in our case, the eye tracker was indispensable to discover how learners interacted with the software without selecting anything (see Fig. 4A) – considering that WriteBetter offers real-time suggestions without any active user interaction.

B. PARTICIPANTS, CONTEXT AND APPARATUS

Experiments were performed on 11 participants (8 M/3 F) aged 20-25 (M = 22.5, SD = 1.2), who were recruited through convenience sampling. They were undergraduate students of a four-year English teaching programme at a Chilean university, and whose native language is Spanish. Four of them were in the second year and seven were in the third year, and their English level varies from B1 to B2 according to the language progression of this English teaching programme.

Regarding corpus use, three participants declared an occasional use of corpora to write in English; five declared that they knew what a corpus is, but never used it for writing English; the remaining three declared not to know what a corpus is. At any rate, we point out that the English programme of that university has no courses on corpus linguistics, and this matter was not taught in any course of the programme. We excluded freshmen as their level of English could have prevented them from producing a significant piece of writing. The study was conducted at the library of the university, which is a natural setting for the participants. Each participant wrote their text individually in the computer provided by the researchers. We used a 15.4'' notebook equipped with the Eyetribe eye tracker. Eye movements and screen recording were collected using OGAMA 5.0 [42].

C. PROCEDURE

At the recruiting state, participants were asked to write a one-page essay using the software, which was simply described as a tool that could help them improve their writing. However, using it was a suggestion rather than a requirement: the participants were invited to use it freely when they felt the need in an ordinary writing process. No further details about the software were explained. After that, 11 students decided to volunteer. The day on which participants took part of the study, they signed a consent form and watched a video.[8] The video introduces the software, and participants could see some of the possible uses of WriteBetter.

After, we started screen and eye-tracking recording. Then, the study began: the participants were asked to write a short essay about any familiar topic or situation related to their lives, particularly:

Write one page about a story of your life. For example, tell a funny experience, what you liked to do when you were a child, a trip with your parents or friends, or another story you would like to write.

Although we requested to write one page, we set a limit of 40 minutes. Once the time was up, we asked the participants to conclude the idea they were writing even if the requested length was not achieved. At any rate, we said to the participants that they could leave if they felt the story/text was completed even before the requested time and length so that they could feel as free as possible reducing any pressure.

At the end of the study, the participants were invited to fill in a questionnaire where they provided their personal information (age, gender, previous experience on corpus use), and impressions through an open text field.

D. USE CASE EXTRACTION AND CLASSIFICATION

To extract use cases, we used a video composed of two superimposed layers. The first layer renders the eye traces based on eye-tracking data whereas the second shows the screen recorded. The latter includes mouse movements, text selection, and written sequences. This overlaid video was used to analyse in detail (frame by frame) the participants' interaction with the writing environment. While observing the video, an Excel file was filled in with several criteria to ensure an objective analysis of each use case (see supplementary materials). Then, according to such information, the two authors of this paper discussed and classified use cases in different categories based on two dimensions: interaction with the software and writing outcome (see Table 2).

A use case starts (1) when users select a piece of text and look at the corpus or (2) when they do not select anything but stop writing and look at the corpus. The latter was possible thanks to the real-time suggestions displayed by WriteBetter. A use case ends when users continue writing after the last corpus consultation attempt related to the case.

Finally, we extracted further minor information regarding the search modality (selecting text/without selecting anything), Smart Search use, corpus issues and their consequences, and the influence of the grammar checker of Microsoft Word.

E. RESULTS

We present the type of interaction (good, neutral, bad) and the outcome (improvement, neutral, worsening). Then, the participants' individual performance is presented and discussed.

1) EMERGED USE CASES

We identified 99 relevant use cases. Table 3 shows the details on all cases classified, as noted above, according to interaction, outcomes, and linguistic aspects (with some examples for each of them). Note that the number of cases (99) does not correspond to the number of searches since, often, participants conducted several searches in a single case. Out of 99 cases, good uses of the software were 75, ambiguous uses were 14, whereas bad uses were 10.

[8] https://www.youtube.com/watch?v=D3is_gMLjDc

TABLE 2. Criteria to classify use cases.

Interaction	Good	Users interacted with the concordancer/corpus taking advantage of the linguistic data. This includes the users' verification of their previous knowledge, awareness of the use of a specific word/structure/tense/collocation/spelling, among others.
	Bad	Users interacted with the concordancer/corpus but without solving their linguistic doubts. This is the case of long searches and misinterpretation of the linguistic data.
	Ambiguous	The researchers were not able to understand the case clearly, e.g. uninterpretable users' behaviour, unclear eye tracker data, etc. Ambiguity is not related to disagreements between the coders.
Writing outcome	Improvement	The original word/structure/tense/collocation/spelling was incorrect and, after consulting the corpus, the user wrote it correctly. All improvements are good uses.
	Neutral	After consulting the corpus, the user neither improved nor worsened. It could be with utterance change (e.g. change of idea), or without utterance change. They could be *good uses* – confirmation of previous knowledge; *bad uses* – an error remained an error due to failed interaction with the corpus or misinterpretation of data; or *ambiguous uses*.
	Worsening	After consulting the corpus, the user's word/structure/tense/collocation/spelling worsened. They can be either bad or ambiguous.

Regarding interaction, we carried out binomial tests to understand whether good uses (75) were significantly higher than ambiguous and bad uses (24, aggregating these cases): clear positive tendencies toward good uses (proportion .24 vs .76, $p < 0.001$) were detected. Moreover, regarding outcomes, we carried out another binomial test to understand if improvements (31) were significantly higher than worsening (5): clear positive tendencies toward improvements (proportion .14 vs .86, $p < 0.001$) were also detected. Note that neutral cases (63) were simply ignored since they are not relevant for the test scopes.

Regarding good uses, 31 cases were classified as improvements whereas 44 cases were classified as neutral without change, i.e., participants just checked if something they wrote was right (confirming and reinforcing their existent knowledge) and kept it. Most improvements regarded linguistic aspects such as collocations (including prepositions), spelling, and vocabulary, whereas most good uses regarded collocations, chunks of words, and vocabulary.

Regarding ambiguous uses, nine cases were classified as neutral (four without changes and five with utterance change) and five as worsening. Neutral cases without changes regarded those cases in which we did not grasp the semantic sense of the expression the participants checked through the software, so we could not understand its influence. Neutral cases with utterance change, instead, often regarded changes of ideas. In one case, "satuday" became "saturday" thanks to Microsoft Word spelling checker, so there was no influence of the concordancer, and we classified the case as ambiguous and neutral. Worsening, instead, regarded lexical collocations (four cases) and vocabulary (one case). Particularly, three cases were classified as "soft" worsening because the participants made unnecessary changes even when conveying the same meaning of the original form (e.g., "impatient person", which was not found in the corpus, became "the capacity for waiting and being patient"). Two cases, instead, were classified as serious worsening since participants made changes that conveyed a different meaning (e.g., "burning sun", which was not found in the software, became "sun"). In four cases, worsening occurred because there were corpus issues – i.e., no results – while searching for specific expressions. In one case, we noticed a useful result among the output of the software, but the participant probably did not notice it, i.e., searching for "van", most of the results regarded people's names (e.g. "Van Gogh"), but there was a result that meant what the participant was expecting, i.e., vehicle.

Regarding bad uses, eight were neutral with no changes of which three regarded long selections, two regarded verb use, and three were related to preposition use. In the long selection cases, participants selected long chunks and, although the concordancer provided no results, they kept the chunks without changes. In the verb use cases, the concordancer provided suggestions, but participants did not understand that these did not fit the meaning they were trying to convey and, therefore, the original text was erroneously kept. In the preposition use cases, participants checked forms like verb + preposition that were found among the results of the concordancer, but they did not check the right use (e.g. using them wrongly such as "arrived to + place" instead of "+ verb"). Regarding bad uses with utterance change, we identified two cases. The first one was a chunk of words which was wrong before and after the utterance change (i.e., "since everyone else were" became "since everybody else were", instead of changing the verb form). The second one was related to vocabulary: the participant replaced the word "stiff" with "rigid" even if among the suggested results there were many cases in which the meaning of "stiff" was clearly inferable. Although this change was not necessary, it does not represent an actual worsening and we classified it as a simple utterance change (perhaps the participant thought that "rigid" could be more

TABLE 3. Examples of cases extracted.

Interaction	Outcomes	Linguistic aspects	Examples
Good uses (75)	Improvements (31)	4 grammatical collocations	story to → story to tell participating on → participating in
		5 lexical collocations of which 1 vocabulary	exemplified behaviour → excellent behaviour in that moment → at that moment on the distance → in the distance
		1 person agreement	who were → who was
		9 spelling	grand children → grandchildren desmotivated → demotivated tourguide → tour guide espectacular → spectacular
		1 superlative	most silly → silliest
		3 verb forms	did not knew → did not know, have had bad lucky → have been unlucky
		8 vocabulary of which 2 spelling	exited → excited tropheums → cups and medals paviment → street took off → took out
	Neutral with no changes (44)	9 chunks	as usual one either lived one when I was running
		10 grammatical collocations	flying above floating in very proud of started becoming
		11 lexical collocations	phisical education dry landscapes in high school in the night night life
		1 passive voice	to be operated
		1 person agreement	there was no one
		1 quantifier	some food
		4 verb forms	I had never been remembered
		6 vocabulary	outstanding quick condors slippery
Ambiguous uses (14)	Neutral with no changes (4)	2 chunks	on a specific
		2 grammatical collocations	laying on
	Neutral with utterance changes (5)	2 chunks	with my father in → in my father's
		2 lexical collocations	in the afternoon → my friends and I always played
		1 spelling	satuday → saturday
	Worsening (5)	4 lexical collocations	impatient person → capacity for waiting and being patient burning sun → sun
		1 vocabulary	van → working as a tour guide with my father driving his bus
Bad uses (10)	Neutral with no changes (8)	3 (long) chunks	once I got there, I realized
		3 grammatical collocations	arrive to asked for
		2 verb forms	had chosen
	Neutral with utterance changes (2)	1 chunk	since everyone else were → since everybody else were
		1 vocabulary	was stiff → was rigid

appropriate in that case, regardless of the influence of the software). The entire data set of use cases is available in supplementary materials.

Moreover, we classified corpus issues, which occurred (1) when a query reasonably common (e.g. burning sun, quick moves) provided no results and (2) when a query provided at

TABLE 4. Participant data.

	Corpus experience	Use time (min)	Text length (words)	Good Uses	Ambiguous	Bad Uses	Improvements	Neutral	Worsening
1	None	24	242	4	2	3	0	7	2
2	None	28	268	14	0	1	8	7	0
3	Some	38	451	10	0	0	4	6	0
4	None	39	283	13	5	0	7	8	3
5	Some	29	224	10	1	2	2	11	0
6	None	7	88	3	0	0	3	0	0
7	None	25	314	5	0	0	3	2	0
8	None	39	293	6	6	2	1	13	0
9	None	15	301	4	0	1	2	3	0
10	None	10	151	3	0	1	0	4	0
11	Some	15	254	3	0	0	1	2	0
	Total	*269*	*2869*	*75*	*14*	*10*	*31*	*63*	*5*
	Mean	*24.5*	*260.8*	*6.8*	*1.3*	*0.9*	*2.8*	*5.7*	*0.5*
	SD	*11*	*88.4*	*4*	*2.1*	*1*	*2.5*	*3.8*	*1*

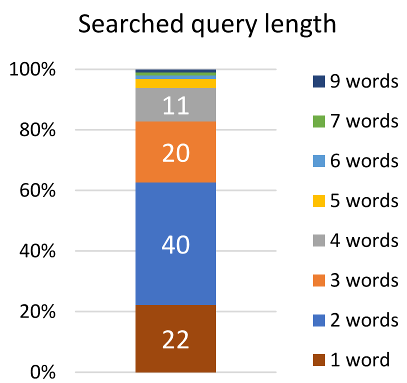

FIGURE 5. Distribution of the lengths of the searched queries with their frequencies.

least one wrong result (e.g. "did not knew"). Surprisingly, in all 10 cases, corpus issues did not cause errors – they only caused unnecessary utterance changes, which were classified as worsening, but they cannot be classified properly as errors.

We also considered the influence of Microsoft Word grammar checker. In all cases, except one, there was agreement between the concordancer results and Word suggestions. Just in one case Microsoft Word highlighted "and also" as an error, but the participant ignored Word suggestions since it was found many times in the corpus.

It was surprising that the Smart Search function was not as useful as expected. Just in one case, it provided one additional result. In the remaining cases, no search improvements were detected.

Finally, in 10 cases out of 99, the participants looked at the results without selecting any text. In three cases – classified as bad use – the participants selected unreasonably long chunks of texts. The length of the searched queries ranges from one word to nine words. More than 93% of searches regarded queries of less than five words (see Fig. 5).

2) PARTICIPANT DATA

Table 4 shows participant data. The average use of the software was 24.5 minutes per participant. In that time, the participants wrote an average of 260.8 words each. The use time varies substantially among participants (SD = 11 minutes).

The participants had an average of 2.8 improvements and 0.5 worsenings, whereas good uses were 6.8, bad uses 0.9 and ambiguous uses 1.3. According to the Wilcoxon signed-rank test, improvements were significantly higher than worsenings ($Z = -2.406$, $p = 0.016$). Moreover, we carried out the Friedman test discovering significant differences among good, ambiguous, and bad uses ($2(2) = 17.282$, $p < 0.001$). Post hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$. Such tests revealed significant differences both between good uses and bad uses ($Z = -2.941$, $p = 0.003$) and good uses and ambiguous uses ($Z = -2.810$, $p = 0.005$).

Moreover, there are several notable individual differences in terms of uses. Surprisingly, P1 had more worsenings (2) than improvements (0), although the difference is not significant (binomial test, $p>0.05$). The five worsenings were distributed only between two participants (P1 and P4, with two and three worsenings respectively) whereas the others did not have any. Each participant, except for P1 and P10, had at least one improvement. Neutral outcomes are significantly higher than improvements in P5 and P8 (binomial test, $p<0.05$), so we could state that these participants used the concordancer mainly to verify the correctness of what they wrote. Unexpectedly, the sum of ambiguous and bad uses of P1 and P8 exceeds good uses, so we could state that these participants did not understand well how to use the concordancer. P2 had more improvements (8) than neutral outcomes (7); moreover, improvements of P2 were significantly higher than worsenings (binomial test, $p<0.05$); therefore, the concordancer was particularly useful for him/her. Good uses of P2 and P3 are significantly higher than ambiguous and bad uses in aggregated form (binomial test, $p<0.05$).

Regarding the free text field, all participants wrote mainly positive comments and few suggestions for improvement. P4, P5, P7, and P9 stated that the concordancer was useful to get suggestions on preposition use. P3 and P5 found the concordancer useful to check the spelling of words. P3 also would like to integrate a bilingual vocabulary (translator), since it would be useful to discover new words. P5 also stated that KWIC visualization displays a lot of text and does not facilitate reading. P1 stated that the concordancer was useful to find more words and improve vocabulary. P6 stated that the software was useful to correct small grammatical errors. P10 found the software useful to study English and that s/he would like to have it installed in his/her computer. P2 stated that the concordancer was useful, but it is needed to pay attention to the provided examples to choose carefully which of them are relevant. P11 stated that the software is useful both for teaching and improving writing. P8, P9 and P11 appreciated the real-time suggestions, which helped them correct errors rapidly (P8), saving time (P11), and making the user surer about what s/he is writing (P9). P7 and P9 appreciated the fact that it was not needed to access the internet allowing them to save time.

F. DISCUSSION

1) INFLUENCE ON WRITING

According to our evaluations, it could be stated that corpus-learner interaction through the integrated concordancer affected writing positively, both objectively – considering the improvement/worsening rate – and subjectively – considering users' impressions. Broadly speaking, the participants appreciated the simplicity of use, the ability to provide real-time suggestions without searching on the internet, and the fact that its use increases their confidence while writing, helping reducing errors. The error correction aspect is in line with other studies [43], [44].

Regarding the way in which participants solved their doubts, we could state that – at least in most of the improvements – participants' uncertainties were faced in interactive and iterative ways (as also pointed out by other studies, e.g. [2], [27], [45] performing and refining several queries until finding the answers the participants were looking for.

2) FOCUS ON THE CORPUS

The corpus quality was overall good, although the Wikipedia corpus cannot be considered properly adequate for writing stories related to everyday people's lives. Such kind of essay would probably require a more "narrative" corpus. In fact, common expressions frequently used in narrative writing (e.g. I flew by, quick moves etc.) were not found in the Wikipedia corpus. At any rate, despite corpus issues, participants were able to face them without making errors (but unnecessary utterance changes were often made). In the remaining cases, neither serious errors in the corpus caused errors in writing. For example, a participant tried to find "did not knew" finding just a (wrong) result in the corpus. Then, s/he tried to search for "did not know" finding many results in the corpus, realising that the latter was surely right.

3) INFLUENCE OF MICROSOFT WORD

Regarding Microsoft Word spelling and grammar suggestions, they were in accordance with the concordancer suggestions, except for a single, trivial case – i.e., "and also", which is highlighted by Microsoft Word suggesting to "consider using concise language" and inviting to replace it just for "and". We believe this agreement is quite important since, otherwise, learners would get confused by contrasting suggestions, and this would increase their writing doubts instead of reducing them.

4) SMART SEARCH

We must report that the Smart Search function was not as useful as expected. We realised that the name we gave to the function was probably misunderstood. In fact, observing how participants used it, they probably expected something really "smart". According to this consideration, it would be probably needed to find another name that better represents its function, e.g. "Include word variations". In fact, the function just searches for word variations to present more results.

5) LONG SEARCHES

Few bad uses of the concordancer were related to unreasonable long selections of 6-9 words. In such cases, the software simply provides a message saying, "No suggestions were found"; but often users did not try shorter selections to attempt finding other possible results. In such cases, meaningful suggestions could be provided, e.g., the software could show sentences related to the searched words. This solution is used in the new version of WriteBetter (see Fig. 10).

6) STRATEGIES EMERGED FROM LEARNERS

We identified several strategies adopted by users to solve their doubts. In some cases, participants searched for different alternatives, e.g. "participating on" first, and "participating in" later. According to the suggestions, then, the participants tried to understand which the right form was according to their context of use and, finally, chose one of them. In other cases, a strategy focused on observation emerged, e.g. searching "that moment", a participant observed which the probable prepositions were before the expression and, then, chose "at". This strategy was also used to find words after some expressions, e.g. "sitting on" followed by "a chair" or "story to" followed by "tell"; in such cases, "a chair" and "tell" were written after the participants observed those words among the corpus results. In addition, more sophisticated strategies also emerged when results were not available in the corpus. In several cases, when searching for long queries that provided no results (e.g. "which is what matter"), participants selected a reduced chunk of words in sequence (e.g. "which is what" first, and "is what matter" after) to see if they were present in the corpus singularly. Then, analysing the contexts in its entireness, they were able to understand the correctness of the expression. Another similar example regarded the substitution of a word for a more

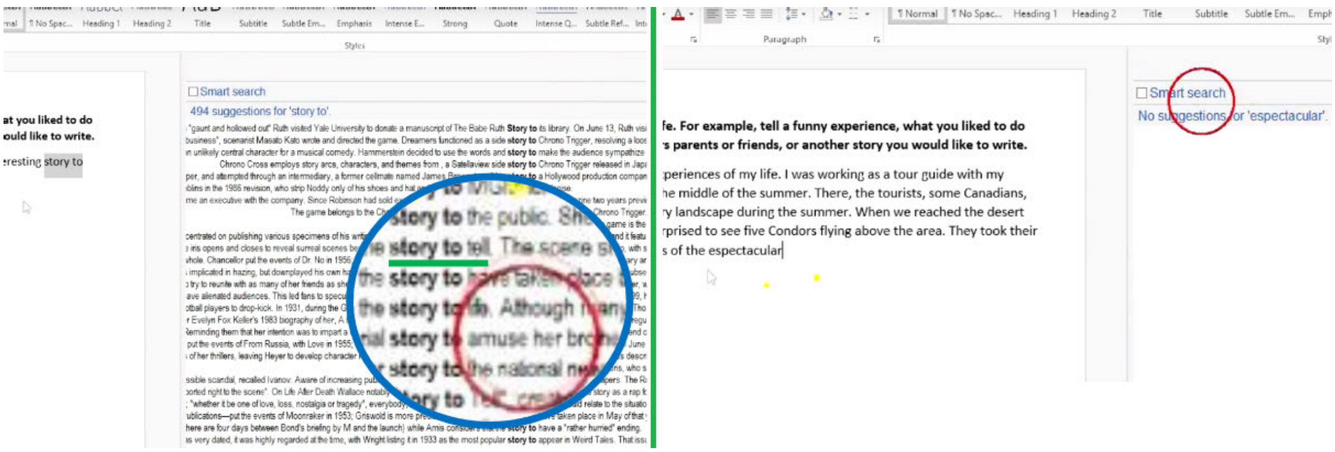

FIGURE 6. On the left, a participant selected "story to" and, observing the suggestions, realised that it was followed by "tell" and wrote it. On the right, the participant wrote "espectacular" and, realising that no suggestions were provided, removed the initial "e". Without the eye tracker, in these cases, we could not have identified clearly, and objectively, change reasons.

common word to increase the probability to find that expression in the corpus. For example, in the expression ''took out their camera'', the chunk ''their camera'' was replaced with ''something'' to see if any results would be shown. We must state that all these strategies emerged without any training in many participants, but not in all.

7) TO TRAIN OR NOT TO TRAIN: THAT IS THE QUESTION

Based on the section above, it could be said that training is partially needed, at least for the weakest learners. At any rate, training seems not to be needed to face technological issues related to the use of the software. According to previous studies, technological issues are one of the barriers for DDL [13], [46]. Introducing WriteBetter, we tried to remove such barriers by simplifying as much as possible learner-corpus interaction. We believe that the lack of agreement emerged in literature on the need for training (e.g. [13], [26], [32] should be questioned considering the influence of software tools. The latter is a means to access the corpus that cannot be excluded when investigating corpus-learner interaction, unless researchers investigate the effect of DDL using just paper-based materials. In the latter case – excluding the influence of computer – such materials were used, with good results, without any training [13]. Instead, when considering DDL computer effects, training seems to be needed [26]. Therefore, to reduce the need of training, it could be useful to minimise the computer negative effect. In this regard, our point of view emerges from our results: a well-designed concordancer can minimise computer negative effects, and most learners can achieve excellent (P2, P3, P4, P6) or good (P5, P7) results also without training – even if the weakest learners (P1, P8, P9, P10, P11) could benefit from a short training in which strategies on corpus use are briefly described.

8) EYE TRACKING: A METHODOLOGICAL ADVANCE

Analysing our videos, we realised that eye tracking was indispensable not only to identify when the participants used the software without selecting anything but also to understand if a change in writing was really caused by one of the results provided by the software. In previous studies, even where screen recording was used [2], [3], changes in writing could just be inferred with reasonable (or doubtful?) sureness. Neither can recall activities, often conducted to disambiguate doubtful cases, be effective to objectively understand whether changes in writing were caused by suggestions found in the corpus (due to biases that could occur, e.g. observer-expectancy effect or confusing memories). Using eye tracking, instead, in most of the cases, we could identify the reasons of writing changes with great sureness. Fig. 6 shows a pair of these cases that demonstrate the usefulness of the eye tracking.

9) LIMITATIONS

Regarding the participants selected for this study, there is a limitation related to their profile, which is quite narrow, i.e., undergraduate students aged between 20-25. Hence, it would be intriguing to discover whether the same interaction with the software emerges in other types of participants who use English for different purposes (e.g., workers, researchers). Clearly, if we had studied different types of participants - including workers and researchers for example - the total number of participants would have had to be much higher than 11 in order to be able to identify possible significant differences.

Regarding the eye tracking, it allowed us to understand in detail the learner-corpus interaction process. Nevertheless, the accuracy of eye tracking was not as good as expected in all the participants since the study was conducted in a natural setting where they could move freely. This condition affected the accuracy in a few cases.

Finally, further evaluations may be useful to compare integrated concordancer to non-integrated ones like AntConc or COCA. Excluding user-friendliness – provided by integration and the minimal interaction required by WriteBetter – we do not expect significant differences in terms of effectiveness in error correction. At any rate, it is likely to expect that the integration of a concordancer into the word processor encourages

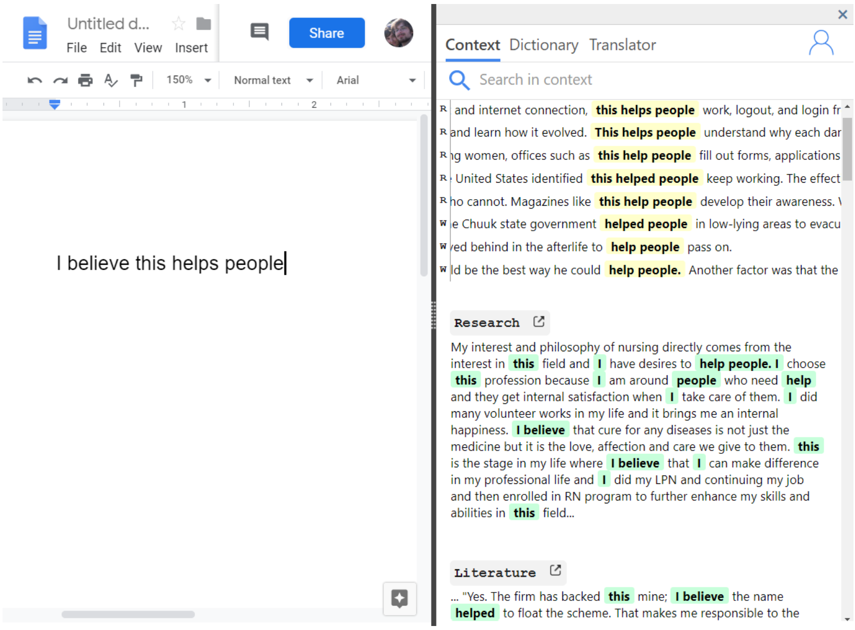

FIGURE 7. WriteBetter integrated into Google Docs showing real-time suggestions according to the three last words (on the right-top), and the entire context of the written text (on the right-bottom).

learners to use it more frequently than a non-integrated tool. In fact, the use of an integrated concordancer is objectively simpler and faster than the use of a non-integrated one since learners just need to see at the real-time suggestions while writing, or select something in the word processor instead of changing software and write their query.

V. DESIGN OF AN IMPROVED VERSION OF WriteBetter

The results of the study presented above encouraged us to continue the development of WriteBetter. Based on the experience gained in that study and more than one year of users' feedback, we developed a new version of WriteBetter. It offers a main function named Context, and two accessory functions named Dictionary and Translator. As the first version of WriteBetter, the new version is simple as writing in the word processor and its interface is minimal.

A. INTERACTION STYLE AND KINDS OF SUGGESTIONS

As in the first version, no active interaction of the user is required to display real-time suggestions – at least for basic use of the Context function. Just writing a word is enough to show real-time suggestions, but in contrast to the previous version – which used only the last word to predict suggestions – the new version now predicts possible ways to complete a sentence according to the last three words typed (Fig. 7, words highlighted in yellow). Moreover, it shows suggestions related to the entire context of the sentence (Fig. 7, words highlighted in green).

As in the first version, a more "advanced" interaction requires users to select a word to see how it is used (Fig. 8). In this case, and in contrast with the previous version, four kinds of suggestions are displayed: (1) suggestions that include the exact word searched, (2) word variations according to query lemmatization,[9] (3) possible word replacements and (4) possible ways to complete sentences.

Moreover, when users select more than one word, WriteBetter shows the exact search, but also possible ways to combine such words, and possible ways to complete sentences (Fig. 9).

Moreover, when selecting a sentence (more than 4 words, i.e., a long search as described in subsection IV-F.5), WriteBetter displays similar sentences from which users can draw inspiration while writing (Fig. 10).

Finally, WriteBetter provides a dictionary based on WordNet, which displays also the most frequent related expressions based on the corpus (Fig. 11), and a translator-in-context (Fig. 12) from English to nine languages and vice versa (mostly like Linguee or Reverso).

It is worth noting that the translator does not provide automated translations, but aims at displaying bilingual examples leveraging parallel corpora. Therefore, also the translator focuses on corpus consultation. Also, the dictionary and the translator show suggestions according to the selected word/chunk of text. At any rate, note that a search field is available in any function so that users can alternatively write their query instead of selecting a text.

B. TECHNICAL FEATURES

Cloud computing services are used to host WriteBetter, which works as SaaS. Corpora (both monolingual and bilingual) and dictionary definitions are stored and queried

[9]We used Lemmagen, http://lemmatise.ijs.si/ - https://github.com/vhyza/elasticsearch-analysis-lemmagen

[10]https://www.nltk.org/howto/wordnet.html

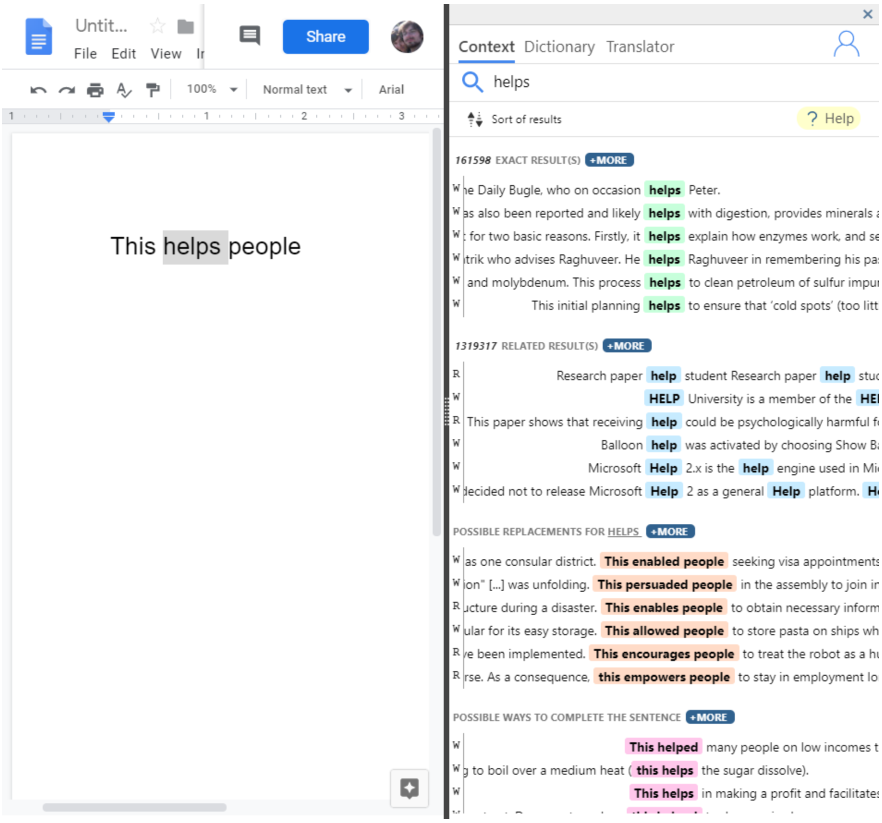

FIGURE 8. WriteBetter suggestions when selecting a word.

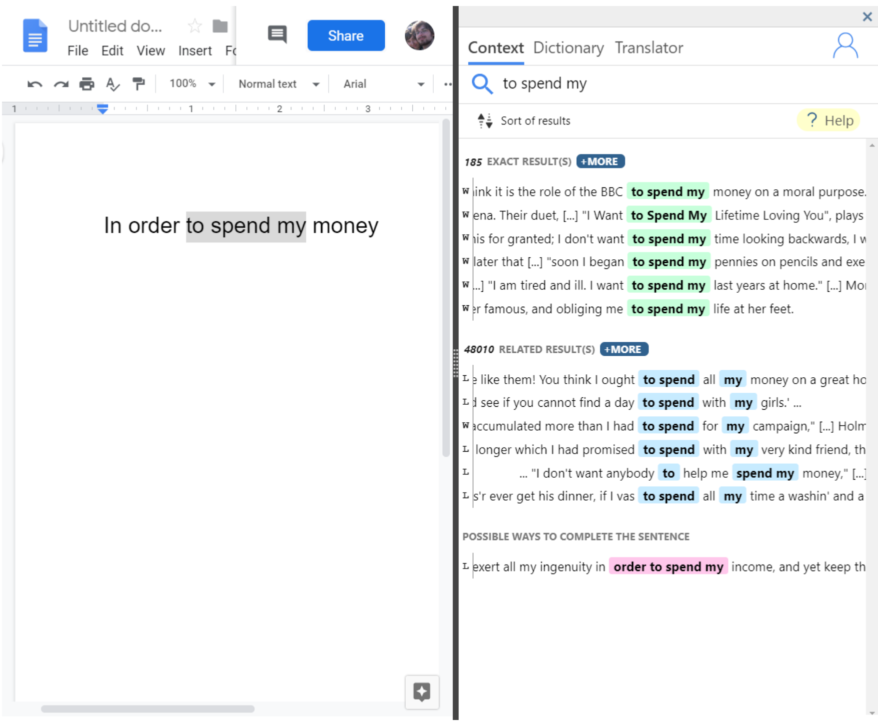

FIGURE 9. WriteBetter results when selecting more than one word.

on Elasticsearch. The backend is developed in PHP whereas the frontend in HTML5/JavaScript. The latter makes WriteBetter available as Word JavaScript Add-In and as an extension for Google Chrome. The extension for Google Chrome enables WriteBetter to work on Google Docs and Overleaf. The corpus included in WriteBetter reach almost

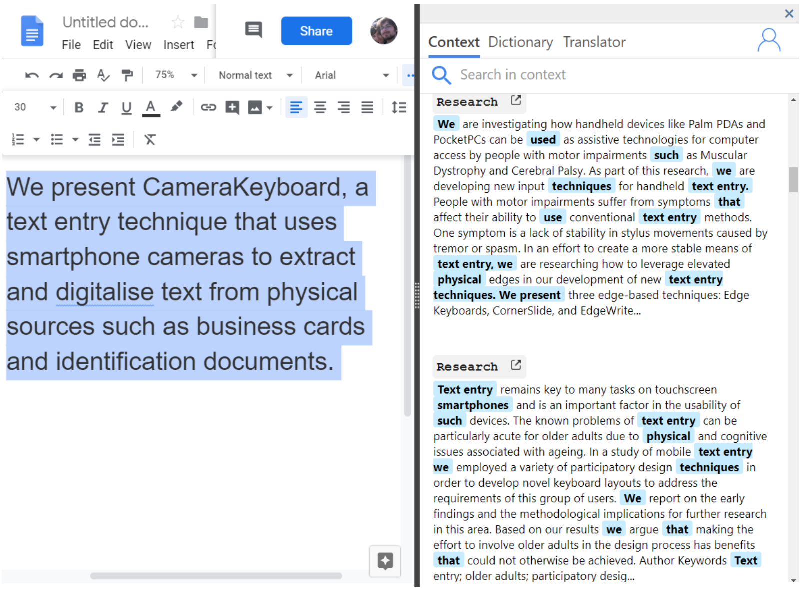

FIGURE 10. Selecting a sentence, similar suggestions are displayed.

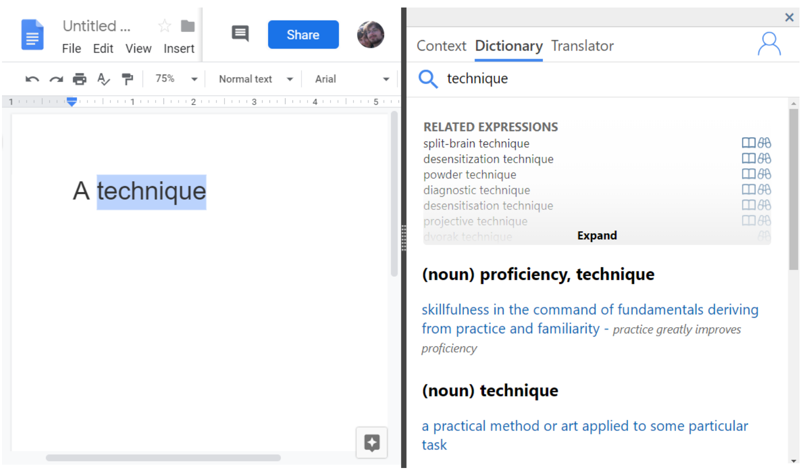

FIGURE 11. The dictionary shows definitions of the words searched focusing on synonyms (WordNet Synsets)[10].

60GB of open-licence texts including Wikipedia, fiction, and scientific articles and bilingual translations got from European Union documents, and the Open Subtitles project.

VI. SECOND RESEARCH QUESTION/USER EVALUATION

The design novelty of WriteBetter raises the question of how users behave when interacting with it for the first time. So, the present user study aims to answer the following research question:

How do users behave when approaching the new version of WriteBetter for the first time?

To respond to the research question, we collected data on 36 users while using WriteBetter for the first time.

A. METHOD, CONTEXT AND PARTICIPANTS

We collected data on users' behaviours between 10th and 15th of December 2019. To record users' behaviour we used Inspectlet,[11] a screen session recording app. We recorded interaction data on the online editor of WriteBetter, which could be accessed from the home page (see Fig. 13).

Users came on the WriteBetter website organically and through Google Ads where we have allocated a budget of around 8$ USD per day. In this way, we could capture 34 sessions per day – the highest number of sessions allowed by the free version of Inspectlet.

We considered only the sessions in which users (1) were able to login on WriteBetter successfully, and (2) interacted with WriteBetter for at least 30 seconds. Out of 170 sessions, the ones that met these conditions were 36 (since most users did not complete the registration, or the user left out after few seconds of trying WriteBetter).

According to users' IP Addresses, they came from 14 different countries (see Fig. 14 for details). Although we did not

[11] https://www.inspectlect.com

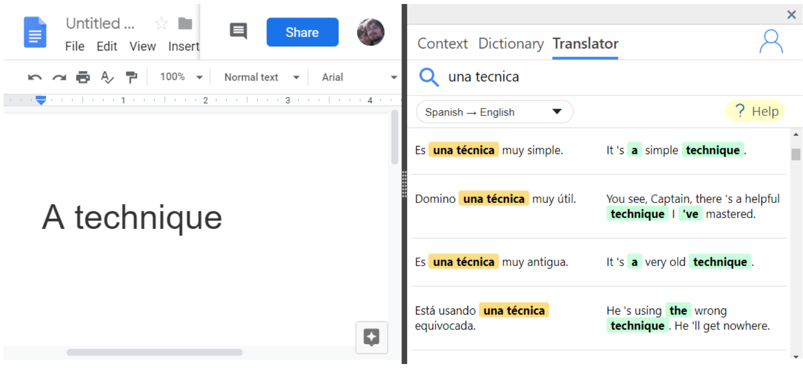

FIGURE 12. Context-based translations: searching a word, bilingual examples are displayed like Linguee or Reverso.

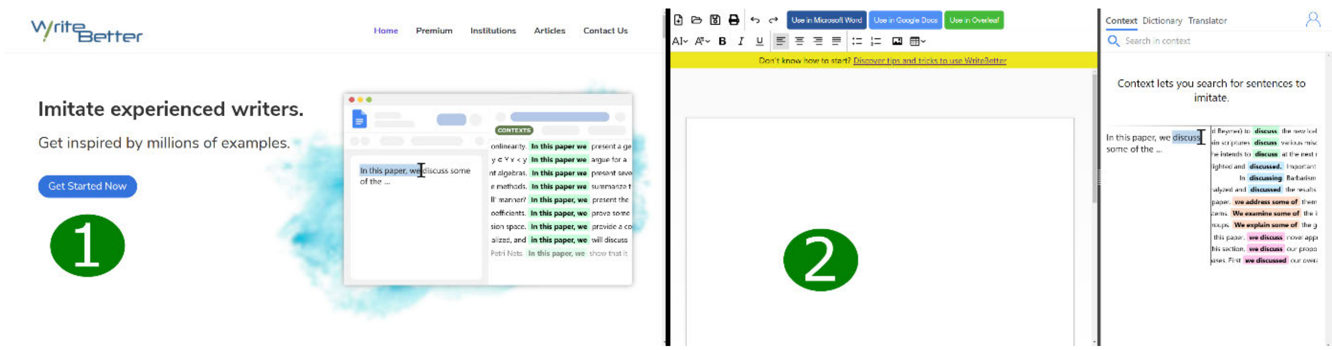

FIGURE 13. The home page of WriteBetter.io on the left. Clicking on "Get Started Now", users access the online version of WriteBetter (on the right).

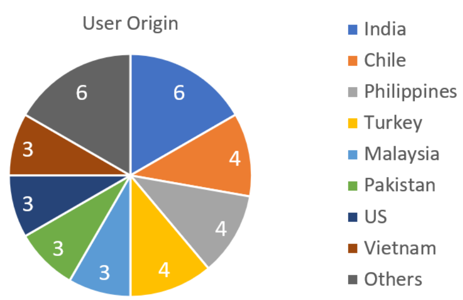

FIGURE 14. Origin of users. Numbers in the pie chart refer to the amount of users from different countries.

ask ages, data from Google Analytics between 10th and 15th of December suggests that 34% of users is aged 18-24, 43% is aged 25-34, whereas the remaining are aged more than 34.

B. USERS' BEHAVIOUR CLASSIFICATION

While observing the sessions, we classified the interactions with the online editor of WriteBetter. The classification regards the use (or not) of the function of WriteBetter (context, dictionary, and translator). Then, we classified other kinds of interactions such as the selection of a long, or short text, the use of the live suggestions, the use of the search field of WriteBetter, the use of scroll to see more suggestions, and the copy/paste of text in the online editor.

C. RESULTS

To begin with, all users (36) used the context function of WriteBetter in a way or another (real-time suggestion after pasting a text, selection, typing in the search field, etc). Only 6 users (out of 36) used the dictionary, whereas 4 used the translator. The context function was used in several ways: 23 users out of 36 pasted a text into the online editor of WriteBetter and click several parts of the text to get real-time suggestions. 17 out of 36 selected a long chunk of text, whereas 12 selected a short chunk of text (less than 5 words). Only five users used WriteBetter by typing in its search field. Surprisingly, three out of these five users did not understand that it was required to press enter to confirm the query to search, so they cannot perform the search as they intended to do – putting a button that explicitly serves to start the search could be a future improvement. Finally, 28 out of 36 users scroll down the examples.

D. DISCUSSION

Unlike traditional writing assistants, WriteBetter provides suggestions while writing (i.e., in the text composition stage), and not after writing. Despite that, it was surprising for us to discover that most users (23 out of 36) pasted a text written beforehand into the online editor to look for related suggestions. In our opinion, this reveals that there is a misalignment between what WriteBetter is, and what users think WriteBetter is. In fact, most users looked for suggestions according to a text written beforehand, namely, without using WriteBetter suggestions to benefit the writing process from the composition perspective. In some way, WriteBetter was mainly used as any other writing assistant, i.e., to check a text after its writing. Probably, this is because users do not expect a writing assistant like WriteBetter, and expect a tool that gives suggestions to improve a given text rather than a writing assistant that helps in the composition process of a text. Moreover, most selections regarded long chunks of text (17 out of 36) rather than short chunks of texts (12 out of 36). This indicates that users are more inclined to search for macro linguistic suggestions (i.e., searching entire paragraphs to discover similar ones, e.g., Fig. 10) rather than micro linguistic suggestions (i.e., searching of a word or combination of 2-3 words to discover how they are used in real contexts, e.g., Fig. 9). Therefore, users expected to find results related to entire sentences or paragraphs rather than related to the use of specific words or combination of words. Moreover, real-time suggestions, in most of the cases, were used to find related sentences (i.e., Fig. 7 right-bottom) rather than the use of the combination of the last 3 words near the cursor (e.g., Fig. 7 right-top). This remarks the interest of users towards macro linguistic suggestions than micro linguistic suggestions. Moreover, the dictionary and translator were not used as expected (respectively, 6 and 4 times out of 36). Regarding the translator, its poor use may be related to the lack of the languages of users (only European languages are available in WriteBetter whereas most users come from non-EU countries). Finally, we point out that most users scroll down the examples showing some degree of curiosity while reading them – many users took a long time to read.

VII. CONCLUSION

The results of the first evaluation let us think that WriteBetter may be a useful tool when it comes to writing. However, the second evaluation – where we have considered how ordinary users approached it for the first time – lets us think that WriteBetter needs some use guidelines. It should be made clear that WriteBetter is mainly designed to provide suggestions to improve the writing process from the composition point of view (idea generation, word choice, idea completion, etc), and not to improve a text given beforehand, even if the latter is also possible. This requires effective communication when presenting WriteBetter, especially considering usually users do not read web pages, but just scan them [47]. Finally, after careful consideration, we decided to remove the translator from the latest version of WriteBetter since its use was quite limited and does not justify extra costs of cloud computing services. However, we kept the dictionary considering that it could have a more general utility. However, the results of the dictionary are only shown when they are actually available (i.e. when searching for a word or expression included in it), as shown in Fig. 15. In this way, we could also simplify the UI as function tabs (i.e., Context, Dictionary, and Translate) are not needed anymore.

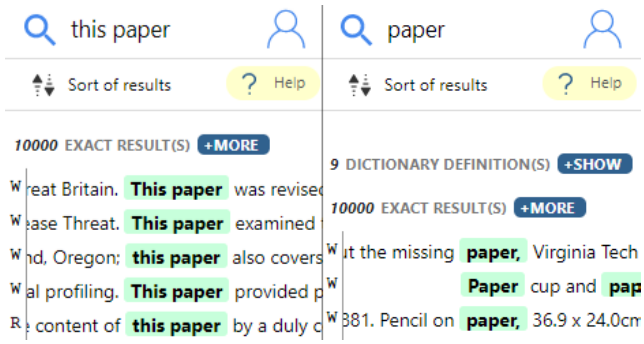

FIGURE 15. **Current version of WriteBetter without translate function. Dictionary results are displayed only when meaningful results are available. For example, searching "this paper" (on the left), no dictionary definition is available since the query is not appropriate for a dictionary. Instead, searching only "paper" (on the right), nine dictionary definitions are proposed and users can open them by clicking on +Show.**

From the concordancer point of view, integration is the feature that makes WriteBetter different from other concordancers. Despite that – according to the first evaluation – effectiveness of WriteBetter in error correction seems to be in line with the one of non-integrated ones (e.g. [43], [44]), we remark that the integration radically changes the paradigm with which learners access, perceive, and consume corpus, as it allows them to interact with corpus focusing on what to do with it, rather than where to find it and how to get it (concept borrowed from the definition of "Situated Software" [48], a variation of the "Situated Computer" concept mentioned above in the paper).

REFERENCES

[1] P. Crosthwaite, L. L. C. Wong, and J. Cheung, "Characterising postgraduate students' corpus query and usage patterns for disciplinary data-driven learning," *ReCALL*, vol. 31, no. 3, pp. 255–275, Sep. 2019.
[2] K. Park, "Learner–Corpus interaction: A locus of microgenesis in corpus-assisted L2 writing," *Appl. Linguistics*, vol. 33, no. 4, pp. 361–385, Sep. 2012.
[3] C. Yoon, "Concordancers and dictionaries as problem-solving tools for Esl Academic writing," *Lang. Learn. Technol.*, vol. 20, no. 1, pp. 209–229, 2016.
[4] C. Yoon, "Individual differences in online reference resource consultation: Case studies of korean ESL graduate writers," *J. 2nd Lang. Writing*, vol. 32, pp. 67–80, Jun. 2016, doi: 10.1016/j.jslw.2016.04.002.
[5] A. Frankenberg-Garcia, R. Lew, J. C. Roberts, G. P. Rees, and N. Sharma, "Developing a writing assistant to help EAP writers with collocations in real time," *ReCALL*, vol. 31, no. 1, pp. 23–39, Jan. 2019.
[6] C. F. Meyer, *English Corpus Linguistics: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[7] J. Flowerdew, "Corpora in Language Teaching," in *The Handbook Language Teaching*, M. Long and C. Doughty, Eds. Hoboken, NJ, USA: Wiley, 2009, pp. 327–350.
[8] N. Schmitt, *Vocabulary in Language Teaching*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[9] T. Johns, "Should you be persuaded: Two examples of data-driven learning," in *ELR Journal 4: Classroom Concordancing*, T. Johns and P. King, Eds. Birmingham, U.K.: The Univ. of Birmingham, 1991, pp. 1–16.
[10] A. Boulton, "Data-driven learning and language pedagogy," in *Language, Education and Technology*, S. Thorne and S. May, Eds. Cham, Switzerland: Springer, 2017, pp. 1–12, doi: 10.1007/978-3-319-02328-1_15-1.
[11] A. Boulton and C. Landure, "Using corpora in language teaching, learning and use," *Recherche Et Pratiques Pédagogiques En Langues De Spécialité. Cahiers de l'Apliut*, vol. 35, no. 2, pp. 1–13, Jun. 2016.

[12] A. Boulton, ''Testing the limits of data-driven learning: Language proficiency and training,'' *ReCALL*, vol. 21, no. 1, pp. 37–54, Jan. 2009.
[13] A. Boulton, ''Data-driven learning: Taking the computer out of the equation,'' *Lang. Learn.*, vol. 60, no. 3, pp. 534–572, 2010.
[14] S. Conrad, ''Will corpus linguistics revolutionize grammar teaching in the 21st century?'' *TESOL Quart.*, vol. 34, no. 3, p. 548, 2000.
[15] Ì. O'Sullivan, ''Enhancing a process-oriented approach to literacy and language learning: The role of corpus consultation literacy,'' *ReCALL*, vol. 19, no. 3, pp. 269–286, Sep. 2007. [Online]. Available: http://www.journals.cambridge.org/abstract_S095834400700033X
[16] U. Römer, ''Corpus research applications in second language teaching,'' *Annu. Rev. Appl. Linguistics*, vol. 31, pp. 205–225, Mar. 2011.
[17] C. Yoon, ''Concordancing in l2 writing class: An overview of research and issues,'' *J. English for Academic Purposes*, vol. 10, no. 3, pp. 130–139, Sep. 2011.
[18] H. Yoon, ''More than a linguistic reference: The influence of corpus technology on L2 academic writing,'' *Lang. Learn. Technol.*, vol. 12, no. 2, pp. 31–48, 2008.
[19] M. Levy, ''Concordances and their integration into a word-processing environment for language learners,'' *System*, vol. 18, no. 2, pp. 177–188, Jan. 1990.
[20] L. Anthony, ''AntConc: Design and development of a freeware corpus analysis toolkit for the technical writing classroom,'' in *IEEE Int. Prof. Commun. Conf.*, 2005, pp. 729–737.
[21] M. Davies. (2008). *The Corpus of Contemporary American English (COCA): 560 Million Words, 1990-Present*. [Online]. Available: https://corpus.byu.edu/coca/
[22] A. Kilgarriff, V. Baisa, J. Bušta, M. Jakubíček, V. Kovář, J. Michelfeit, P. Rychlý, and V. Suchomel, ''The sketch engine: Ten years on,'' *Lexicography*, vol. 1, no. 1, pp. 7–36, Jul. 2014.
[23] A. Frankenberg-Garcia, ''Combining user needs, lexicographic data and digital writing environments,'' *Lang. Teaching*, vol. 53, no. 1, pp. 29–43, Jan. 2020.
[24] T.-H. Yen, J.-C. Wu, J. Chang, J. Boisson, and J. Chang, ''WriteAhead: Mining grammar patterns in corpora for assisted writing,'' in *Proc. ACL-IJCNLP Syst. Demonstrations*, Jul. 2015, pp. 139–144.
[25] S. Tarp, K. Fisker, and P. Sepstrup, ''L2 writing assistants and context-aware dictionaries: New challenges to lexicography,'' *Lexikos*, vol. 27, no. 1, pp. 494–521, 2017.
[26] C. Quinn, ''Training l2 writers to reference corpora as a self-correction tool,'' *ELT J.*, vol. 69, no. 2, pp. 165–177, Apr. 2015.
[27] H. Yoon and J. W. Jo, ''Direct and indirect access to corpora: An exploratory case study comparing students' error correction and learning strategy use in L2 writing,'' *Lang. Learn. Technol.*, vol. 18, no. 1, pp. 96–117, 2014.
[28] M. Charles, ''Getting the corpus habit: EAP students' long-term use of personal corpora,'' *English for Specific Purposes*, vol. 35, pp. 30–40, Jul. 2014, doi: 10.1016/j.esp.2013.11.004.
[29] M. A. Conroy, ''Internet tools for language learning: University students taking control of their writing,'' *Australas. J. Educ. Technol.*, vol. 26, no. 6, pp. 861–882, 2010.
[30] M. Charles, ''Proper vocabulary and juicy collocations': EAP students evaluate do-it-yourself corpus-building,'' *English for Specific Purposes*, vol. 31, no. 2, pp. 93–102, Apr. 2012, doi: 10.1016/j.esp.2011.12.003.
[31] S.-L. Lai and H.-J.-H. Chen, ''Dictionaries vs concordancers: Actual practice of the two different tools in EFL writing,'' *Comput. Assist. Lang. Learn.*, vol. 28, no. 4, pp. 341–363, Jul. 2015.
[32] A. Boulton, ''Data-driven Learning: Reasonable Fears and Rational Reassurance,'' *Indian J. Appl. Linguistics*, vol. 35, no. 1, pp. 81–106, 2009.
[33] C. Dalton, ''Interaction design in the built environment: Designing for the 'universal user,''' in *Universal Design 2016: Learning From the Past, Designing for the Future*, H. Petrie, J. Darzentas, T. Walsh, D. Swallow, L. Sandoval, A. Lewis, and C. Power, Eds. Amsterdam, The Netherlands: IOS Press, 2016, pp. 314–323, doi: 10.3233/978-1-61499-684-2-314.
[34] J. Nielsen. (Sep. 20, 2009). *Fresh vs. Familiar: How Aggressively to Redesign*. Nielsen Norman Group. Accessed: Apr. 10, 2020. [Online]. Available: https://www.nngroup.com/articles/fresh-vs-familiar-aggressive-redesign/
[35] P. Seebach. (Mar. 2, 2005). *Baby Duck Syndrome: Imprinting on Your First System Makes Change a Very Hard Thing*. IBM DeveloperWorks. Accessed: Apr. 10, 2020. [Online]. Available: https://web.archive.org/web/20120419150252id_/http://www.ibm.com/developerworks/web/library/wa-cranky50/index.html
[36] R. Hull, P. Neaves, and J. Bedford-Roberts, ''Towards situated computing,'' in *Dig. Papers, 1st Int. Symp. Wearable Comput.*, Oct. 1997, pp. 146–153. [Online]. Available: http://ieeexplore.ieee.org/document/629931/
[37] G. De Michelis, ''Situated computing,'' in *Designing Socially Embedded Technologies in the Real-World*, V. Wulf, K. Schmidt, and D. Randall, Eds. London, U.K.: Springer, 2015, pp. 65–77.
[38] A. Gershman, J. McCarthy, and A. Fano, ''Situated computing: Bridging the gap between intention and action,'' in *Dig. Papers. 3rd Int. Symp. Wearable Comput.*, Oct. 1999, pp. 3–9.
[39] B. Laufer, ''The contribution of dictionary use to the production and retention of collocations in a second language,'' *Int. J. Lexicography*, vol. 24, no. 1, pp. 29–49, Mar. 2011.
[40] P. Pérez-Paredes, M. Sánchez-Tornel, J. M. A. Calero, and P. A. Jiménez, ''Tracking learners' actual uses of corpora: Guided vs non-guided corpus consultation,'' *Comput. Assist. Lang. Learn.*, vol. 24, no. 3, pp. 233–253, Jul. 2011.
[41] J. H. Goldberg and A. M. Wichansky, ''Eye Tracking in Usability Evaluation. A Practitioner's Guide.,'' in *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, A. Duchowski, Ed. London, U.K.: Springer, 2003.
[42] A. Voßkühler, V. Nordmeier, L. Kuchinke, and A. M. Jacobs, ''OGAMA (Open gaze and mouse Analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs,'' *Behav. Res. Methods*, vol. 40, no. 4, pp. 1150–1162, Nov. 2008.
[43] A. Gilmore, ''Using online corpora to develop students' writing skills,'' *ELT J.*, vol. 63, no. 4, pp. 363–372, Oct. 2009.
[44] R. W. Todd, ''Induction from self-selected concordances and self-correction,'' *System*, vol. 29, no. 1, pp. 91–102, Mar. 2001.
[45] J.-Y. Chang, ''The use of general and specialized corpora as reference sources for academic english writing: A case study,'' *ReCALL*, vol. 26, no. 2, pp. 243–259, May 2014.
[46] H. Yoon and A. Hirvela, ''ESL student attitudes toward corpus use in l2 writing,'' *J. 2nd Lang. Writing*, vol. 13, no. 4, pp. 257–283, Dec. 2004.
[47] K. Pernice, K. Whitenton, and J. Nielsen, *How People Read on the Web: The Eyetracking Evidence*. Fremont, CA, USA: Nielsen Norman Group, 2014.
[48] S. Balasubramaniam, G. A. Lewis, S. Simanta, and D. B. Smith, ''Situated software: Concepts, motivation, technology, and the future,'' *IEEE Softw.*, vol. 25, no. 6, pp. 50–55, Nov. 2008.

ALESSIO BELLINO received the Ph.D. degree in computer science from the University of Milano-Bicocca. He is currently an Assistant Professor with the School of Informatics and Telecommunications, Universidad Diego Portales, Chile. His research interests include interaction design, human–computer interaction, and ubiquitous computing.

DANIELA BASCUÑÁN received the M.A. degree in Applied Linguistics and Language Teaching from Lancaster University, U.K. She also teaches Academic Writing at Pontificia Universidad Católica de Chile. She is currently the Co-Founder of WriteBetter, a writing assistant which helps users get suggestions while writing. Her research interests include language ideologies, corpus linguistics, and computer-assisted writing.