

NetLangEd, A Web Editor to Support Online Comment Annotation

Rui Rodrigues ✉

Departamento de Informática, University of Minho, Braga, Portugal

Cristiana Araújo ✉ 🏠 

Centro Algoritmi, Departamento de Informática, University of Minho, Braga, Portugal

Pedro Rangel Henriques ✉ 🏠 

Centro Algoritmi, Departamento de Informática, University of Minho, Braga, Portugal

Abstract

This paper focuses on the scientific areas of Digital Humanities, Social Networks and Inappropriate Social Discourse. The main objective of this research project is the development of an editor that allows researchers in the human and social sciences or psychologists to add their reflections or ideas out coming from reading and analyzing posts and comments of an online corpus ¹. In the present context, the editor is being integrated with the analysis tools available in the NetLang platform. NetLangEd, in addition to allowing the three basic operations of adding, editing and removing annotations, will also offer mechanisms to manage, organize, view and locate annotations, all of which will be performed in an easy, fast and user-friendly way.

2012 ACM Subject Classification Human-centered computing → Interactive systems and tools; Software and its engineering → Application specific development environments

Keywords and phrases Online Annotation tool, Document Markup System, Text Editor, Discourse Analysis

Digital Object Identifier 10.4230/OASICS.SLATE.2021.15

Funding This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: PTDC/LLT-LIN/29304/2017.

1 Introduction

This paper discusses a tool developed in the context of the project *NetLang* – The Language of Cyberbullying: Forms and Mechanisms of Online Prejudice and Discrimination in Annotated Comparable Corpora of Portuguese and English. In the last decades, we have witnessed an exponential growth of the Internet, more specifically in the way we communicate with each other, that brought many great things to our society from breaking distances between people to giving voice to those who didn't have one, allowing them to report the many injustices happening in the modern days. However it also empowered anti-social behaviors like online harassment, cyberbullying and hate speech. This type of communication is usually hostile and malicious, expressing discrimination, intimidation and disapproval towards certain characteristics like sex, race, religion, ethnicity, colour, national origin, disability, or sexual orientation of a person or a group of people. The objective of this kind of speech is to injure, harass and degrade the targeted person or group in order to dehumanize them [1, 2, 5, 9, 12, 13]. In order to solve this problem, many big companies made available several options to address this type of speech like flagging, reporting, counter-speaking or simply censor certain words that are commonly used in this type of speech. However these strategies do not always work,

¹ Consisting of texts extracted from various sources of Social Computer Mediated Communication (comment boards of news sites and social networks) which are then converted to a specific *JSON* format.



© Rui Rodrigues, Cristiana Araújo, and Pedro Rangel Henriques; licensed under Creative Commons License CC-BY 4.0

10th Symposium on Languages, Applications and Technologies (SLATE 2021).

Editors: Ricardo Queirós, Mário Pinto, Alberto Simões, Filipe Portela, and Maria João Pereira; Article No. 15; pp. 15:1–15:16



OpenAccess Series in Informatics

OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

generating many cases of outrage like the case of 2013 where several pages were found in Facebook with hateful content towards women like *Violently raping your friend just for laughs* and *Kicking your girlfriend in the fanny because she won't make you a sandwich*. In a matter of weeks a petition was created that aggregated 200,000 signatures and many important companies either removed their ads or threatened removing them from Facebook [10]. In that context, the platform *NetLang* is being developed to support researchers in the human and social sciences or psychologists collecting and making available for exploration a corpus of posts and comments that express this kind of hateful language. To improve the usability of the platform it was decided to develop a tool, like an editor, that would allow the researchers to write their thoughts on specific parts of the comments creating annotations and saving them so they can continue the analysis later. Our initial survey found several tools that offer this type of functionality like *Word*², *Adobe Reader*³, *doccano*⁴, *Hypothes.is*⁵ and *Genius Web Annotator*⁶, differentiating among them in the way the note is created, the method used to save/display them, the text formatting options available, etc.

After studying the problem through the exploration of other works and solutions related to the topic, two objectives were established for the work here reported: to develop an editor that allows users to annotate the text that they are analyzing; and finally to carry out usability, conformance, and performance tests with the developed editor and real end users. As final result, it is expected to have an editor that has a simple and easy to use interface. This project will be very useful for its researchers since it will allow them to have the content under study and their notes located on the same platform, ensuring better organization and accessibility.

The paper is organized as follows. Section 2 will cover topics such as the history of annotations, how people annotate on paper and on the web, annotation functionality and types, advantages and disadvantages of digital annotations and an analysis of the existing solutions. In Section 3 the system requirements will be listed and the system architecture will also be discussed and sketched using block diagrams. Section 4 describes how the editor was developed, showing some screenshots of the final result. Finally, in Section 5 an analysis of the results obtained will be made and conclusions will also be drawn regarding the work done and the future work.

2 Document Annotation, state-of-the-art

In this section the state of the art regarding annotation will be portrayed. Its initial part will be used to describe the evolution of the annotations over the years. Then, studies conducted by several researchers will be analyzed in order to understand the behavior of readers when annotating on paper and online. Based on these studies and other articles, the purposes and types of annotations will be described in general followed by the listing of the advantages and disadvantages of digital annotation compared to physical annotation. After that, some tools that allow the annotation of digital texts will be listed and compared taking into consideration some dimensions that are considered to be important for the development of *NetLangEd*.

² Available at <https://www.groovypost.com/howto/annotate-in-word/>, accessed in December 2020

³ Available at <https://helpx.adobe.com/acrobat/using/commenting-pdfs.html>, accessed in December 2020

⁴ Available at <https://doccano.github.io/doccano/>, accessed in December 2020

⁵ Available at <https://web.hypothes.is/help/annotation-basics/>, accessed in December 2020

⁶ Available at <https://genius.com/web-annotator>, accessed in December 2020

History of annotations

The usage of text annotations became a prominent activity around 1000 AD in Talmudic commentaries and Arabic rhetorics treatises⁷. It was then used in the Medieval ages to discuss, critique and learn from annotations created from previous readers who also read from the same manuscript. There were also situations where at the time the manuscripts were being copied, their annotations were included in the copy [14]. However, the emergence of the printing press has made this use of annotations obsolete due to having facilitated the circulation of information and the ability to purchase individual copies of text [14]. Nowadays annotation is an activity that is mostly done in private corresponding to the reader's interaction with the text being read. Computer-based technologies also provide many solutions for both individual and shared annotations, allowing to apply this method to online and offline digital documents [14].

How people annotate on paper and on the web

In order to understand the behavior of readers when annotating on paper and on the web, the results described in the papers [11, 7, 3, 8] were studied, obtaining several conclusions for both cases.

In the case of paper annotation, it was verified that highlights were the most used type of annotation and the most common purpose was to remember, thus being able to relate these two since highlights are normally used to help in the memorization process and to make it easier to find the important parts of the text in a later reading. Another reason for this predominant use of highlights is because they allow the reader to stay focused on the task of reading since it is a method of quick execution compared to the other ones. Still regarding the highlights, another important characteristic of these are their colors, which may have additional meanings and can facilitate detection. In one of the analyzed experiments, it was also found that the purpose of reading greatly influences the way in which readers annotate their documents, namely in the types of annotations they use. A final observation is that the annotations with text that are shared are written in a more explicit way so that other readers understand them more easily.

Regarding annotation on the web, it is difficult to determine which types of annotations are most used since it depends on the features that the systems provide. That being said, in all cases, readers were careful to place the notes as close as possible to the respective parts of the document to which they referred. Another interesting observation is that, in cases where it was possible to highlight, this tended to be the most common choice, confirming the popularity of this type of annotation. In this context, it was discovered that they had the purpose of signaling parts of the text that were not understood, that they wanted to remember or because they were important. As in the paper case, here the colors of the highlights are also of great importance, allowing readers to better structure their annotations. Regarding the notes that were made with the purpose of being shared, it was also possible to verify the same situation that was described in the paper case. These are more developed and explicit than the private notes, which are more short and ambiguous. Through this, it is possible to conclude that the readers when writing private notes are only concerned with their significance to themselves while in the case of shared notes they are written so that the other readers have no problems in understanding them.

⁷ Available at https://en.wikipedia.org/wiki/Text_annotation, accessed in December 2020

Functionality of annotations

There are several benefits that are obtained through private and public annotations. Taking into account how people annotate on paper and on the web, as explained in the previous section, and based on the work done by Jindia and Chawla [6] it was possible to identify the following ones: facilitate the current or future reading process; facilitate the later writing process; understand the insights of another reader; provide feedback to the text writer or other readers; help with the memorization and recall process; draw attention to certain parts of the text that are considered to be important for future reference or reading and correct a specific part of the text.

Types of annotations

Annotations can take different forms where some of them assume a textual representation and others consist of graphic effects. Thus, through the same sources of knowledge used in the previous subsection, the following types of annotation were identified:

- **Mark:** The method of marking an important word/phrase through visual effects. These can be highlights, underlines, strikeouts, figures, etc.
- **Paraphrase:** It consists of reproducing in a simpler and more accessible way the central ideas of the original text, without changing its meaning.
- **Comment:** This type is based on the formulation of comments to specific parts of the text that are directed to the writer or self-directed. These may be of agreement/disagreement, questions, responses, connection to ideas from other texts, personal experience, adding explanation, etc.

Note that both the *Paraphrase* and *Comment* type can be combined with the *Mark* type in order to be able to contextualize them in the text.

Advantages and Disadvantages of Digital Annotations

In this subsection we will discuss the advantages and disadvantages of using digital annotations in comparison to paper-based annotations, since one of the main reasons that led to the development of this work was the fact that the users of the platform *NetLang* were using paper to write their comments/analysis of the text being read. To point out pros and cons of digital annotation, the considerations previously mentioned and the work of Glover, Xu and Hardaker [4] were taken into account.

The **advantages** that have been identified are as follows: better organization due to not having the notes spread over several sheets; adding annotations won't damage the original text; easier to change the content of annotations; allows the removal of annotations without leaving marks in the document; it allows you to write notes with more extensive content without having to worry about the space they will occupy and easier to locate annotations on the document through mechanisms.

The **disadvantages** identified were the following ones: it is not possible to directly manipulate the document requiring to follow a specific process that is more complex than simply using the pencil to write and draw on the paper; the forms that digital annotations can take are limited to those offered by the tool used; although a simple click of a button is enough to share digital annotations with other users; the circulation of a physical document will always be the most intuitive way to carry out this process and requires the user to learn how to use the tool.

Existing Tools

To conclude this section, the documentation of several tools that have annotation mechanisms was analyzed. Of the several that were found, the most popular or that have the most interesting characteristics are the following ones: *doccano*⁸, *Word*⁹, *Adobe Reader*¹⁰, *Weava*¹¹, *LINER*¹², *Diigo*¹³, *Hypothes.is*¹⁴ and *Kami*¹⁵. The *A.nnotate*¹⁶ tool was also considered, but since it is not as popular as the tools mentioned before it will not be covered here as well as the tools *Inception*¹⁷, *Prodigy*¹⁸, *UBIAI*¹⁹ and *LabelStudio*²⁰ since they are very similar to the *doccano* tool. In order to be able to compare them, some points that were considered to be important for the editor were stipulated, being the following: *Annotation of web pages*, *Free*, *Multiple forms of annotation*, *Text formatting*, *Search facilities* and *Export and import*. After defining everything, the tools were compared in Table 1.

■ **Table 1** Tool comparison.

Tool	Annotation of web pages	Free	Multiple forms of annotation	Text formatting	Search facilities	Export and import
<i>doccano</i>	No	Yes	Limited	No	No	Yes
<i>Word</i>	No	No	No	Yes	Limited	No
<i>Adobe Reader</i>	No	Yes	Yes	Limited	Yes	Yes
<i>Weava</i>	Yes	Limited	No	No	Yes	No
<i>LINER</i>	Yes	Limited	No	No	Yes	No
<i>Diigo</i>	Yes	Limited	Limited	No	Yes	No
<i>Hypothes.is</i>	Yes	Yes	No	Yes	Yes	No
<i>Kami</i>	No	Limited	Yes	Yes	Limited	No

Through Table 1 it is possible to conclude that, although there is no perfect tool, the most complete is the *Adobe Reader* since it is the one that satisfies most of the requirements in a satisfactory way. However, it does not fulfill the most important requirement for this work, this being the ability to annotate web pages. Thus, if we look at the tools that fulfill this requirement, the best one would be *Hypothes.is* due to fulfilling the greatest number of the remaining requirements. Another factor that should be mentioned is that this tool is the only one of the four web based tools that have been analyzed here that does not restrict some of its functions through free and paid accounts. The only negative aspects of this tool is that it only allows the use of highlights, to which it is possible to associate comments, as a form of annotation and doesn't have any mechanism that allows to export the work done so that it can be imported later to continue it.

⁸ Available at <https://github.com/doccano/doccano>, accessed in April 2021

⁹ Available at <https://www.microsoft.com/en-us/microsoft-365/word>, accessed in April 2021

¹⁰ Available at <https://get.adobe.com/br/reader/>, accessed in April 2021

¹¹ Available at <https://www.weavertools.com/>, accessed in April 2021

¹² Available at <https://getliner.com/>, accessed in April 2021

¹³ Available at <https://www.diigo.com/>, accessed in April 2021

¹⁴ Available at <https://web.hypothes.is/>, accessed in April 2021

¹⁵ Available at <https://www.kamiapp.com/>, accessed in June 2021

¹⁶ Available at <http://a.nnotate.com/>, accessed in June 2021

¹⁷ Available at <https://inception-project.github.io/>, accessed in June 2021

¹⁸ Available at <https://prodi.gy/demo>, accessed in June 2021

¹⁹ Available at <https://ubiai.tools/>, accessed in June 2021

²⁰ Available at <https://labelstud.io/>, accessed in June 2021

3 NetLangEd, the proposed architecture

In this section, the work proposal will be described in detail starting with a list that contains the features that the editor to be developed must have followed by the diagrams of the system architecture. Based on the information collected during the literature review phase to understand the *State of the Art*, discussed in Section 2, the features that the editor should present will now be defined.

The editor will have the following **Functional Properties**:

- **Marks on the text:** The part of the text that the user notes should be marked in some way.
- **Highlight color options:** The user must be able to customize the color of the highlight.
- **Tooltips in annotated text:** When hovering the cursor over the annotated text, a tooltip containing at least the respective comment must be presented.
- **Text formatting:** The user must be able to format the comment text in different ways, such as changing the font size, changing the font family, creating lists, etc.
- **Annotated comments list:** The user must be able to see all comments on the annotations made in the document he is analyzing.
- **Removal:** There should be options that allow the removal of annotations and their comments.
- **Editing:** As with removals, there should also be options that allow editing annotation comments.
- **Location discovery:** There must be a mechanism that allows a quick location of the annotation of the respective comment in the document.
- **Search:** The user should be able to search a word/phrase in the comments of the annotations.
- **Filter:** The user must be able to filter comments according to various criteria.
- **Import:** The user must be able to import from his computer the annotations and their respective comments that were previously made in the document that is currently being analyzed.
- **Export:** The user must be able to export to his computer the annotations and their respective comments that were made in the document.
- **Clear the document:** The user must be able to remove all annotations in the document at once.

The **non-functional properties** that the editor will possess are as follows:

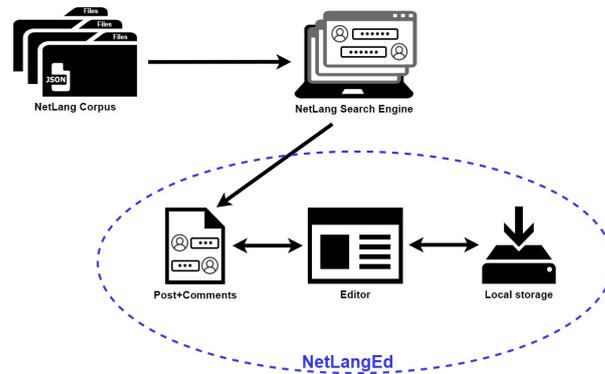
- **Simple and clear interface:** The interface should not take up too much space so as not to distract the user and its content must be simple and explicit so that the user does not feel confused when using it.
- **Simple and clear functionalities:** The functionalities must be easy to understand and to execute so that the user does not have difficulties in using the tool.
- **Quick add functionality:** This functionality should be quick to perform, being done in the smallest number of steps possible.
- **Quick edit functionality:** Editing annotation comments should be possible both in the annotation comments list and in the document, thus allowing the user to remove them in any situation. This functionality should also be quick to perform, being done in the smallest number of steps possible.
- **Quick removal functionality:** As in the case of editing, the option to remove comments from annotations should be possible to execute both in the comment list of annotations and in the document, thus allowing the user to remove them in any situation. This functionality should also be quick to perform, being done in the smallest number of steps possible.

- **Annotation overlay:** The parts of the text where annotations overlap should be properly treated so that the annotations involved can be easily distinguished.
- **Annotation comment representation:** An annotation comment must be presented in the same way both in the annotation comment list and in the document's tooltips.

After the identification of the functional and non-functional requirements for the comments annotation editor under discussion, its architecture is sketched.

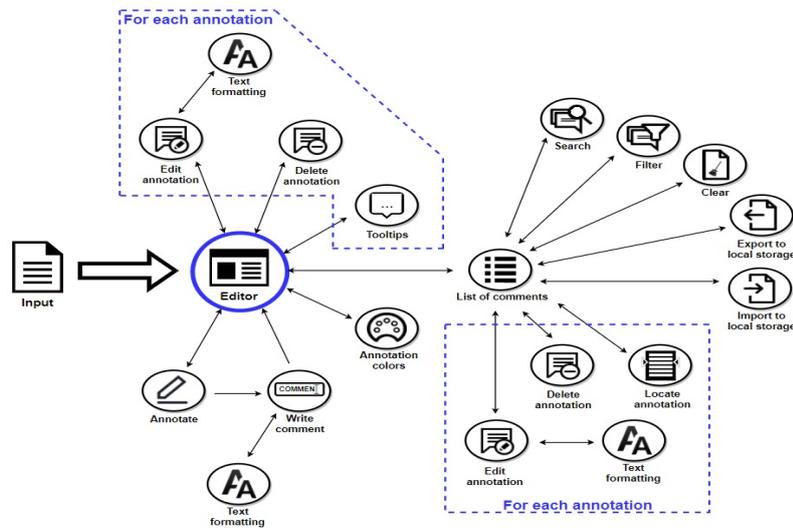
3.1 System Architecture

In this subsection, several diagrams will be presented that will allow to better understand the architecture of the system. Thus, Figure 1 presents the architecture of *NetLangEd* and its integration with *NetLang* platform.



■ **Figure 1** *NetLangEd* Architecture and Integration with *NetLang* platform.

Observing Figure 1 it is possible to see that the editor will be accessed from the pages that contain the posts and comments that are stored in the *NetLang* repository. Another aspect that can be seen in Figure 1 is the possibility of exporting and importing the work done, these being done to and from the user's computer respectively. Finally, Figure 2 presents how the functionalities are organized in the system and how they can be accessed.



■ **Figure 2** *NetLangEd* usage diagram.

15:8 NetLangEd, A Web Editor to Support Online Comment Annotation

As can be noticed in Figure 2, some of them (add, edit and remove annotations, choose the color of the annotations and tooltips) can be executed directly on the text; the remaining are executed on the menu.

4 NetLangEd, development

In this section it will be described how each of the requirements that were listed in Section 3 has been achieved and how the integration process with the *NetLang* platform was carried out.

4.1 Functional properties

The functional requirements that were implemented will be discussed in this subsection.

Marks on the text

This requirement, that consists of allowing users to annotate HTML text without affecting its original formatting, was the most important objective to be achieved in this project. Several solutions were tested, with the vast majority of them failing with some tags such as lists, links, etc or simply making it difficult to add, edit and remove annotations. Taking these problems into account, the only solution that avoided most of them was to surround each word, with the exception of HTML tags, with a *span* tag, after which it was decided to surround each character in order to be able to annotate any part of the words instead of being forced to annotate it entirely. These *span* tags have a unique identifier in order to be able to change their properties more easily, and for that it was necessary to develop a function that returned the identifiers of the *spans* that are present in the selection made by the user. It should be noted that this approach is not without problems, being dependent on the text to be annotated to remain unchanged so that the identifiers remain constant and other problems that arose during the integration phase with the *NetLang* platform that will be discussed in Subsection 4.3. Having said that, the adding process begins with the user selecting the part of the text that he wants to annotate followed by clicking on the pop-up that appears after completing the selection, in order to confirm the intention to annotate the selected part of the text. After that, the addition box will appear in the sidebar in which the user can enter the comment, with the possibility of associating tags to it, completing the process using the button that allows to save the comment. This whole process can be seen in Figure 3.



■ Figure 3 Text annotation functionality.

Highlight color options

This feature was implemented as a palette with the color options that are available (see Figure 4). Initially it was possible to pick any color, however this method would complicate not only the process of filtering the annotations by their color because it would lead to too many options but it would also complicate the process of picking a color since the user would have to either find the desired color or would have to save the color he picked for later use.

Tooltips in annotated text

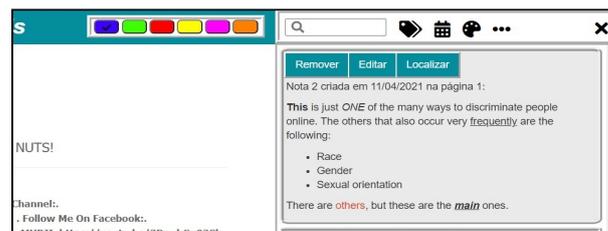
This feature can be seen whenever the user places the mouse cursor on top of an annotation presenting the result visible in Figure 8b. In addition to containing the comment that is associated with the annotation, the number of the note is also shown to make it possible to identify them more easily. However, there are some restrictions regarding the display of comments in the tooltips. In order to make sure that the tooltip is always visible, it can only be shown up to six lines of the comment. In cases where there is more content to show, an ellipsis is shown.

Text formatting

Concerning text formatting, there are many options that can be included, as *undo*, *redo*, *clear formatting*, *bold*, *italic*, *underline*, *strikethrough*, *superscript*, *subscript*, *fonts*, *font sizes*, *numbered list*, *bullet list*, *text color*, *background color*. From a technical point of view all of them can be included. However only fifteen (as can be seen in Figure 3) were implemented because the others were difficult to display in the tooltips.

Annotated comments list

As can be seen in Figure 4, all the annotations that were made on the document that is currently being analysed will be listed on a sidebar. Another important aspect of this sidebar was its simplicity, clarity and convenience. To this end, it was decided to keep at the top of it all the options that are used on a recurring basis separated, like the filters that will be talked about later, and group in one option all the options that wont be use as regularly, like exporting, importing, etc. Another decision that was made was to place all the options that can be made on the comments that are present in the sidebar on top of each one of them.

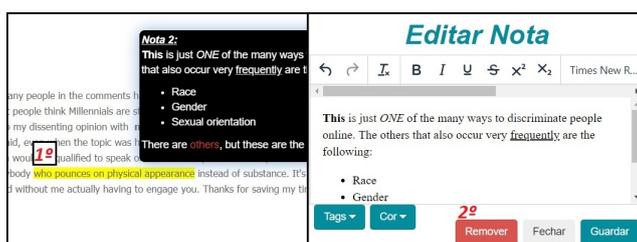


■ **Figure 4** List of comments in the sidebar.

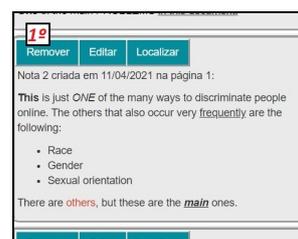
Removal

This requirement has been implemented and can be executed in two ways, where one is performed by clicking on the desired annotation, that is present in the document, showing the edition box that has the button that allows its removal and the other way is through the removal button that is located over the annotations present in the sidebar. Both alternatives can be seen in Figures 5a and 5b.

15:10 NetLangEd, A Web Editor to Support Online Comment Annotation



(a) From the document.



(b) From the sidebar.

■ **Figure 5** Annotation removal functionality.

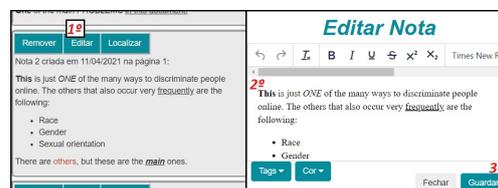
Note that, in both ways, the text box will be precariously filled with the current comment of the respective annotation so that the user can make the decision to remove the annotation more clearly.

Editing

As in the previous *functional property*, the editing process can also be done in two ways. The first consists of clicking on the annotation, present in the document, whose comment is to be edited and the second way is through the button that allows editing that is located on top of each of the annotations present in the sidebar. In both cases the result of the clicks is the display of the edit box where the user can edit the comment, as well as the tags and color associated with it, concluding the process by clicking in the button to save the comment. Both ways can be seen in Figures 6a and 6b.



(a) From the document.



(b) From the sidebar.

■ **Figure 6** Annotation editing functionality.

It should be noted that, in this case, the text box will also be precariously filled with the current comment of the respective annotation so that the user does not have to rewrite the entire text if he only wants to make some changes to it.

Location discovery

The purpose of this requirement was to help the user to easily locate the comment in the document, being implemented through the button at the top of each of the comments present in the sidebar visible in Figure 4. Initially, this feature only scrolled the page to the place where the annotation was located, making it the only visible annotation in order to be more easily detectable. However, after making the decision to display the documents in pages, due to the problems that were detected in the process of integrating the work with the *NetLang* platform and which will be discussed later, this feature now has the ability to switch to the page where the annotation is located in case it was not done on the page that is currently open.

Search

This requirement is present at the top of the sidebar that can be seen in Figure 4. It is important to mention that only the annotations whose comments contain exactly what is entered by the user will be filtered, that is, the search takes into account the small and capital letters, blank spaces, etc. It is also important to note that the result of applying the filter is visible both in the document as in the sidebar.

Filter

This requirement was implemented through the three buttons that are to the right of the search bar present at the top of the sidebar visible in Figure 4, having buttons that allow filtering the annotations by tags, date and color. In addition to these three types of filters, others were discussed, however these seemed the most useful. Note that these three filters can be used together, including with the search functionality. That said, it was necessary to make another decision on how these filters will act together, that is, if it is enough that the annotations comply with one of the filters or if they have to comply with all of them. After some reflection and taking into account that the purpose of filtering is to specify as much as possible a characteristic of something, it was decided that the annotations have to comply with all the filters to be shown. Another characteristic that is worth mentioning is that, as in the previous requirement, the result of applying the filters is visible both in the document as in the sidebar. Regarding the management of the tags, this will be done through the button that allows filtering by tags, through which it will be possible to open the pop-up visible in Figure 7 where the user will be able to create, edit and remove tags.



■ **Figure 7** Tags manager.

Import

This requirement was implemented as a feature that is present in the dropdown that results from clicking on the button that is represented by the three dots visible in Figure 4. While the basic functionality of this requirement was achieved, it was attempted to apply a security mechanism to prevent the user from being able to write malicious HTML code in the part of the comments that would later be run when loading the file. Although it works and prevents the import of a file that has been changed, the user can access the “keyword” and replicate the hash process in order to overcome this barrier. However, it can be said that at least this process cannot be carried out in such an easy way.

Export

This functionality also appears in the drop-down that results from clicking on the button that is represented by the three dots visible in Figure 4. Export option writes in a text file the necessary information so that the user can restart the work where it was left in a next section. Note that the user must click on the *save button* to save the current status of the work done, before exporting it. Otherwise, the work will be lost.

Clear the document

This feature is also present in the drop-down that results from clicking on the button that is represented by the three dots visible in Figure 4. This functionality simply removes all annotations made in the document as well as in the sidebar.

4.2 Non-functional properties

In this subsection, it will be explained how the non-functional requirements were implemented.

Simple and clear interface

This requirement was achieved through several variables that were taken into account in the development of the interface. The first one is related to the way in which the buttons were placed at the top of the sidebar, which were explicitly placed those that are more likely to be used more frequently and those that are used less frequently were grouped in the option characterized by the three dots, as can be seen in Figure 4. Another important aspect is related to the comments of the annotations present in the sidebar. These were embedded inside compartments to better distinguish where a comment starts and ends. Another important decision was to place the operations that can be executed on these comments directly on them in order to be clear to the user to know which comment is applying the operations on. All these decisions can also be seen in Figure 4. The placement of the color palette was also subject of reflection. It was initially planed to be placed in the sidebar, however since it is a tool that the user may want to use at any time, it was decided to keep it fixed next to the sidebar as shown in Figure 4. Regarding the boxes that appear to edit or add a comment, it initially appeared in the center of the screen with the capability to be moved to any other part of the screen. However, since this task of always moving the box is boring and as it is preferable to always be able to see the part of the text that is being commented for contextualization, it was decided that in both cases these boxes would be displayed in the sidebar, as can be seen through Figure 3. Finally, the last point that was considered was the sidebar itself. It was decided to allow it to be hidden or expanded since when it was open it could distract the user. Another detail is related to the fact that it pushes the text that is being annotated to the side instead of overlapping it, thus not covering the text, allowing the user to continue to annotate even with it expanded. This decision can be seen in Figure 4.

Simple and clear functionalities

Most features have a word on their button that clearly describes their purpose, as for example, in the case of Figure 4. There are some exceptions, such as the functionalities at the top of the sidebar that have icons as shown in Figure 4. However, care was taken to use icons that represent the respective functionalities. Thus, only the operations that are executed from

the document, these being the tooltips, removal and editing, are not so obvious to execute. However, in this case it was decided to sacrifice this aspect a little in order to benefit other requirements that will still be discussed here.

Quick add functionality

This functionality was implemented being only necessary to perform four actions (select text, click on pop-up, write comment and save comment) as can be seen in Figure 3. There is a way to reduce the number of necessary steps to three, which consisted in removing the pop-up part and display the addition box after completing the selection, however this method could create complications in cases where the user accidentally selects something that he didn't want to annotate. Note that this process may require a greater number of steps if the user wants to associate tags to the annotation, however it is not a mandatory step in the process.

Quick edit functionality

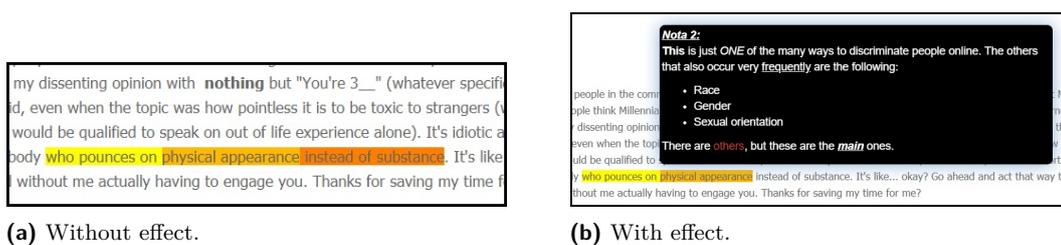
To fulfill this requirement, it was important to make sure that the editing functionality could be performed both in the document and in the sidebar. In this case, both situations only require three actions (in the document it is necessary to click on the annotation, edit and save changes and in the case of the menu it will be necessary to click on the edit, edit and save changes button) to complete the process, both of which are visible in Figures 6a and 6b, without having found any way to reduce this number. Note that, as in the case of addition, this process may require a greater number of steps if the user wants to edit the tags or the color that are associated with the annotation, however these are not mandatory steps in the process.

Quick removal functionality

This requirement, just like in the previous one, was achieved by not only allowing the user to use the removal functionality on the document and on the sidebar but also execute it with the shortest number of clicks. Regarding this last aspect, in the document its only required two steps (click on the annotation and click on the remove button) and on the list of comments its required only one (click the remove button), both of this cases being visible in Figures 5a and 5b. In the case of executing this functionality in the document, it is possible to be done in just one click by right-clicking on the annotation to be removed, however this method could lead to situations where the user could unintentionally remove an annotation.

Annotation overlay

This feature was implemented using a color effect, as can be seen in Figure 8a. Observe that the color effect is present in the part of the text in which the overlap has occurred, which will have a color that will be the result of combining the color of the last annotation made on that part of the text with the color of the new annotation also made on that part of the text. When there are a greater number of overlaps, placing the cursor over an annotation it will only show that specific annotation, as visible in Figure 8b, returning to normal as soon as the mouse is moved out of it.



(a) Without effect.

(b) With effect.

■ **Figure 8** Example of an annotation overlay.

Annotation comment representation

This feature can be seen in Figures 8b and 4, where it is possible to see that the annotation content and its formatting are displayed in the same way in both cases.

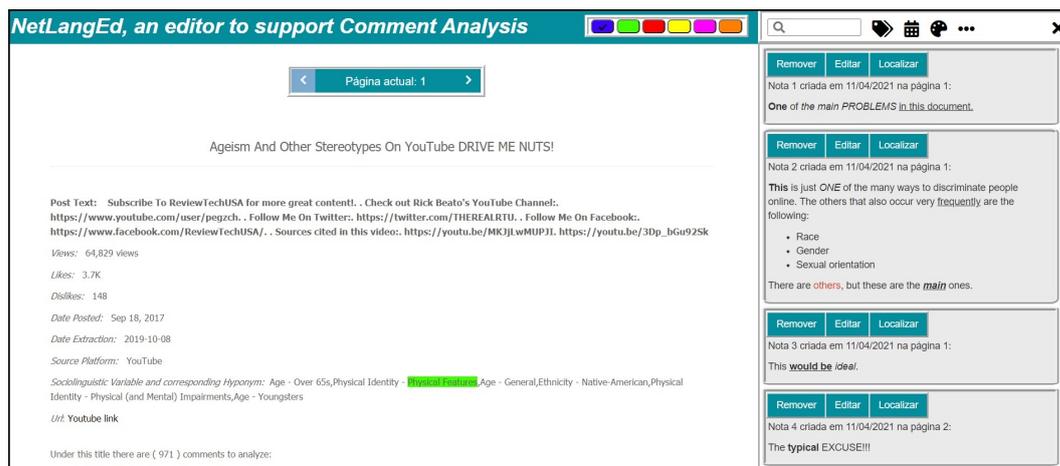
4.3 Integration with the NetLang project

To wrap up this section, where the tool development was discussed, the problems that were encountered during the integration process with the *NetLang* existing platform will be identified and the solutions found will be also presented. The first problem was related to the fact that there are very long texts on which it would be applied the process of placing *span* tags in all of its characters, causing the browser to jam and not load the page. The solution to this problem was to load part of the document instead of loading it all at once and to load more whenever the user pressed a button. Although the previous solution eliminated the problem of not being able to load the page, it did not avoid the second problem that occurred when the page size started to become very large, causing the editor to function slowly. In view of this, the final solution was to load the document as pages that would not exceed a maximum limit, where the user could navigate backwards or forwards one page at a time or even directly open a specific page.

After overcoming all this problem, the planned tool is operational and can be accessed at <http://bit.ly/NetLangEd>. Figure 9 is a screenshot that illustrates a working session. The image shown exhibits the text window on the left with an annotation in green and with the page browser at the top, the menu on the right side with the filters and other options at the top and with all the comments of the annotations made in the document listed below it and with the color palette to the left of the menu.

5 Conclusion

Along this paper it was presented the evolution that occurred in the use of annotations, from a strict use for sharing knowledge to a primarily personal use. Through the analysis of the studies carried out by several authors it was possible to observe details of the readers' behavior when annotating on paper and online, through which were identified not only the types of annotations used but also their purposes. It was also possible to enumerate some advantages and disadvantages that online annotations have compared to paper annotations, allowing to conclude that there are two main factors that determine the preference of its use. The first is related to the original format of the document to be annotated where in the case of being paper it will be very unlikely that it will be digitized to be digitally annotated and the second factor is related to the functionalities that the annotation system provides where in the case of not covering the readers' main needs of annotation will lead them to prefer to print the document and annotate it on paper. In order to understand the characteristics that



■ **Figure 9** Example of a work session.

a web annotator should have, several existing solutions have been analyzed, being possible to observe some characteristics that can be considered essential and others that are either not very useful or simply do not fit the objectives that are intended to achieve with this work. When comparing these solutions it was possible to conclude that, although none was perfect, the best would be *Hypothes.is* since, of the tools that allow the annotation of web pages, it is the one that fulfills the greatest number of the remaining requirements. That being said, the analysis that was carried out in all of them served as a basis to create a list of requirements that the editor to develop must fulfill by which the diagrams of the system architecture were developed. After finishing the development of the editor, it is possible to affirm that all the requirements that were stipulated here were fulfilled. To the best of our knowledge, *NetLangEd* is the first annotation editor integrated with a corpus browser. However, there are some aspects that can be improved or even added in the future, such as the aesthetic component of the editor, allowing filters to act on the form of a conjunction and developing a more efficient way to make annotations so that it is possible to load to the document in its entirety instead of loading it in pages. Another future work will be to conduct tests with users in order to obtain feedback.

References

- 1 Raphael Cohen-Almagor. Countering hate on the internet. *Annual Review of Law and Ethics*, 2014.
- 2 Karmen Erjavec. “You don’t understand, this is a new war!” Analysis of hate speech in news web sites’ comments. *Mass Communication and Society*, 2012.
- 3 Nor Fariza, Hazita Azman, and Afendi Hamat. Investigating students’ use of online annotation tool in an online reading environment. *3L: Language, Linguistics, Literature*, 19:87–101, 2013.
- 4 Ian Glover, Zhijie Xu, and Glenn Hardaker. Online annotation – research and practices. *Computers & Education*, 49(4):1308–1320, 2007.
- 5 David John Harvey. Dangerous speech – some legislative proposals. *SSRN Electronic Journal*, 2019.
- 6 Anjali Jindia and Sonal Chawla. Annotations in e-learning. *International Journal of Computers & Technology*, 12:3852–3859, 2013.

15:16 NetLangEd, A Web Editor to Support Online Comment Annotation

- 7 Ricardo Kawase, Eelco Herder, and Wolfgang Nejdl. A comparison of paper-based and online annotations in the workplace. In *Learning in the Synergy of Multiple Disciplines. EC-TEL 2009. Lecture Notes in Computer Science*, pages 240–253, 2009.
- 8 Jingyan Lu and Liping Deng. Examining students' use of online annotation tools in support of argumentative reading. *Australasian Journal of Educational Technology*, 2013.
- 9 Silvana Neshkovska and Zorica Trajkova. The essentials of hate speech. *International Journal of Education TEACHER*, 2017.
- 10 Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. *Proceedings of the 25th International Conference on World Wide Web – WWW '16*, 2016.
- 11 Iliia A. Ovsianikov, Michael A. Arbib, and Thomas H. McNeill. Annotation technology. *International Journal of Human-Computer Studies*, 50(4):329–362, 1999.
- 12 Alexandra A. Siegel. *Social Media and Democracy: The State of the Field, Prospects for Reform*, pages 56–88. Cambridge University Press, 2020.
- 13 Stefanie Ullmann and Marcus Tomalin. Quarantining online hate speech: technical and ethical perspectives. *Ethics and Information Technology*, 2019.
- 14 Joanna Wolfe and CM Neuwirth. From the margins to the center - the future of annotation. *Journal of Business and Technical Communication*, 2001.