

CHAPTER X

CLEC - Colombian Learner English Corpus: first learner corpus of written production in English online in Colombia

CLEC - Corpus Colombiano de Aprendices de Inglés: primer corpus de producción escrita de aprendices de inglés en Colombia disponible en línea

María Victoria Pardo Rodríguez^a & Antonio Jesús Tamayo Herrera^b
Universidad de Antioquia (^a) – *Colombia*; *Instituto Politécnico Nacional* (^b) – *México*

Abstract: This article aims to introduce CLEC's web application (Colombian Learner English Corpus) to the research community. This application was created to search for information within a learner corpus labeled with error tags to add, modify and eliminate data. After having the corpus collected and tagged, it was necessary to create a tool that systematically searches for information within the labeled data. The compilation of the learner corpus followed the guidelines of the Computational Corpus Linguistics (McEnery & Hardie, 2011) and the parameters of learner corpus Granger (2002), Gilquin (2015). The result is a web app designed to seek error tags within a context that can be easily revised and expanded through the system administrator. This corpus is available online, and it is open to any researcher who wants to consult it or contribute with data to enhance the corpus.

Resumen: Este artículo tiene como objetivo presentar la aplicación web de CLEC (Colombian Learner English Corpus) a la comunidad investigadora. Esta aplicación

fue creada para buscar información dentro de un corpus de aprendices etiquetado con etiquetas de error para agregar, modificar y eliminar datos. Luego de haber recolectado y etiquetado el corpus, fue necesario crear una herramienta que hiciera búsquedas sistemáticas de información dentro de los datos etiquetados. La compilación del corpus de aprendices siguió las pautas de la Lingüística de Corpus Computacional (McEney & Hardie, 2011) y los parámetros de los corpus de los aprendices Granger (2002), Gilquin (2015). El resultado es una aplicación web diseñada para buscar etiquetas de error dentro de un contexto que se puede revisar y expandir fácilmente a través del administrador del sistema. Este corpus está disponible en línea y está abierto a cualquier investigador que quiera consultarlo o que quiera aportar nuevos datos para aumentar el corpus.

1. Introduction

Learner corpora (LC) emerged in the late 1980s (Granger *et al.*, 2015) as a valid scientific way to analyze learners' output and has the same characteristics attributed to other corpora with the difference that the source of data is the output of language learners. Defined as "electronic collections of natural or almost natural data produced by foreign or second-language students (L2) and gathered according to explicit design criteria" by Granger (2002, p.7) and Gilquin (2015, p.1). LC has gained significance in the analysis of students' production. Regarding the authenticity of the data produced in a classroom, it is important to remember that the environment is not completely natural because the activities to obtain that input involve some kind of "artificiality" (Granger, 2002, p.8). Also, special attention must be paid to the criteria to build the corpus. The learner corpus' metadata, such as students' characteristics and the task they develop, are important factors for data collection.

The growth of LC in the late 1980s was in part to its potential to investigate authentic output from students. This methodology gives researchers access to outstanding amounts of data samples to do searches for collocations, patterns, and statistics. In the field of research on second and foreign language acquisition and teaching, learner corpora give access to learners' errors when they have been previously tagged, facilitating the analysis of such errors.

Error Analysis (EA) appeared in the early 1970s, and Corder (1967) was the first author to propose the idea that second language learners generated an autonomous linguistic system that he called "*transitional competence*". The author argued that learners gradually modify their native language rules towards target language rules, probably using a univer-

sal grammar or what he called a “*built-in syllabus*”. Later, Selinker (1972) called the built-in syllabus *interlanguage*, and this is the term that has prevailed in time. It refers to the version of language produced by a learner. The analysis of the interlanguage of learners can be performed through the analysis of errors. Error analysis is “the investigation of the language of second language learners” (Corder, 1971, p.14). These analyses can be done using electronic learner corpora to obtain statistics and patterns and analyze what learners lack or need in their learning process. A learner corpus can be very useful when it has error labels to facilitate extensive studies.

Although the usefulness of a corpus of learners’ language with error labeling is undeniable, it does not, on itself, facilitate extensive studies that could be carried out on it. For that reason, taking advantage of the fact that this corpus has a marking of errors in a set of texts, a collection of documents was generated and later uploaded into a database. After having the corpus collected in electronic format, there was a need for a tool that allowed researchers access to the corpus and provided the possibility of making queries with different filters.

The present paper starts with a brief description of the previous related work in learner corpora. Then, it describes the theoretical framework that supports this work along with the process followed during the compilation of the present corpus and the error tagging process. Afterwards, it narrates how the CLEC’ app was designed and how it works to obtain its best performance. This project was developed with the research group Translation and New Technologies (TNT) of the School of Languages at Universidad de Antioquia and makes part of the products of a doctoral thesis.

2. Previous work

There are numerous corpora of English learners that contain samples of learners who have Spanish as their mother tongue, UC Louvain, (2018). Some of them are the Written Corpus of Learner English (WRICLE) Mendikoetxea *et al.*, (2009); the Santiago University Learner of English Corpus (SULEC) Santiago University, (2002); the Gachon Learner Corpus (GACHON) Carlstrom and Price, (2012); the NON-native Spanish corpus of English (NOSE) Díaz-Negrillo, (2012); the International Corpus of Learner English (ICLE) Granger, (2003). The ICLE and the NOSE can be highlighted as corpora of English language with samples of learners who have Spanish as their mother tongue. The ICLE is considered a pioneer in the field of learner language corpus. It has a relatively large collection (approx-

¹ CLEC can be accessed via this URL: <https://grupotnt.udea.edu.co/clec>

imately 3.7 million words) of learners' written output from 16 different mother tongues, including Spanish. A CD containing the collection of texts must be purchased along with a desktop software to carry out searches and analysis on them to have access to this corpus. On the other hand, the NOSE (The NON-native Spanish Corpus of English) has a collection of approximately 1000 argumentative and descriptive texts from students at the University of Granada and University of Jaen. It has labeling of errors under the EARS system Diaz-Negrillo, (2009). Apparently, this corpus had a web interface for its consultation allowing filtering by subject, text type, and parameters of the student's profile, but it is currently not accessible. Most of these corpora lack error labeling, and none of them currently has an accessible interface for researchers or the public to allow searches on them.

The corpus of the present analysis has a collection of documents labeled with error tags. It lets researchers, students, and teachers carry out searches systematically and with the possibility of filtering errors on different categories and types. Also, with this app, it is possible to obtain examples of these errors and their corrections. For the case of errors that represent more than one error category, a new functionality was developed to change error tags when necessary. This development results from a long process of trial and error, plus tests to achieve an app that allows adding, modifying, or eliminating errors or documents. These functionalities are carried out with a corpus management system that is powerful, versatile, and friendly. Initially, the development of this app was carried out in a technology called Django, which makes use of the Python language, but it was determined that the app should allow not only to consult but also to comply with all the initials of the CRUD concept (James, 1980) (Create, Read, Update, Delete). Therefore, to carry out this scalability process, an architecture and a technology analysis exercise were developed to enable the web application to perform these functions.

3. Corpus collection process

There are several options to collect a learner corpus. It can be collected as part of an academic activity in which all students participate, e.g., as an exam with its corresponding permission for data use. Another option is to ask students to volunteer their work if they are willing to participate. In this second option, attention must be paid not to introduce a bias considering that the most successful students would be more willing to participate than those with a low performance, which would compromise the balance and representativeness of the data.

Regardless of how a corpus is collected, texts in a learner corpus do not occur strictly in a natural way because they are produced in a classroom context and are the result of

activities designed to improve the learners' skills in the target language. In the present research, the output collected results from elicitation techniques that searched for the most natural output from students. The output resulted from questions that elicited students' information or opinions from current situations that affect their daily lives. Participating students were able to choose their own words to express their opinions in their compositions. The present research was based on the analysis of a written corpus from a cross-sectional study.

A written corpus can start with handwritten or typed texts. In the case of handwritten texts, the researcher must make sure the transcription is accurate; therefore, in typing, it is essential to trace the texts for any involuntary addition or loss of data. When all texts are collected, they should be coded, indicating a reference and information that make them traceable. Attention must be paid to quotations that do not belong to the learners' production. Guilquin (2015, p.19) recommends to "remove quotations (which do not represent the learner's own use of language and may therefore have to be excluded from the analysis of the corpus)." In the present work, quotations were not removed to keep the entire context from errors. In some cases, removing quotations would mean losing fundamental parts of the text indispensable to understand the context. On the contrary, they were kept, but close attention was paid to not analyze those parts. On the other hand, in the case of direct computerized versions of learners' texts, they can be kept in files as TXT texts to make sure they can be uploaded in the most appropriate software to conduct the tagging process.

The principles of learner corpora guided the collection of the present corpus (Pardo, 2020). These are some of the guidelines that should be taken into account when designing a corpus of learners, according to Granger, (2002), see Table 1.

Tabla 1. Guidelines for designing a learner corpus (Granger, 2002, p.9).

Learner	Task settings
Learning context	Time limit
Mother tongue	Use of reference tools
Other foreign languages learned	Type of test
Level of performance of English as a Foreign Language (EFL)	Audience / speaker
(The researcher could add other information that consider relevant)	(The researcher could add other information that consider relevant)

After having the institution's permission to carry out the research, several stages were needed to accomplish the collection process. Students did a placement test consisting of an online test supplied by Oxford University Press (Oxford University Press, 2017) and

available at www.oxfordenglishtesting.com. After a brief registration and the introduction of a password, the student starts a one-hour test of about 100 questions that the system sorts out with different degrees of difficulty to determine the student’s language level. This test type guarantees that students are classified according to their performance following the Common European Framework of Reference for Languages (Europe, 2001).

In Table 2 it can be observed how the population of the present study was distributed. Participating students in this study were registered in different semesters from several BA programs offered by the university: Architecture, Basic Sciences, Health Sciences, Law, Politic Sciences, International Affairs, Business School, Humanities and Social Sciences, Engineering, Education Studies, and Mathematics. All participants share the same mother tongue: Spanish and their average age is 23.

Table 2. University classification according to CEFR (Pardo, 2019).

U. Norte Levels	Intro- ductory Level		Level						
	1	2	3	4	5	6	7	8	
CEFR	A1	A2	A2	B1	B1	B1	B2	B2	B2
Number of Students	110	496	439	409	325	356	377	335	286
				Pre- Intermediate	Interme- diate	Intermediate	Intermediate II	Upper- Intermediate	

After the files were collected, they were processed in different ways because they were submitted in different formats. For instance, and because their final work was handwritten, for level B1 the process started with the scanning followed by the texts’ typing. External assistants did the typing of texts in their final year of their BA in languages at Universidad de Antioquia. They were given clear instructions regarding neither adding nor subtracting any words from the original handwritten compositions. After all texts were transcribed, they were thoroughly checked for mistakes and to make sure they were exactly as the original. Next, they were converted into TXT texts to do error annotation. Students from level B2 directly did the digital version; therefore, those texts were immediately converted into TXT format for the error tagger. The handwritten files were in total 373, and the process of typing lasted approximately seven months. After all the previous preparation, all files were ready to start annotation.

3.1. Error annotation process

As any other kind of corpora, learner corpora start as raw texts of electronic versions or transcribed texts from spoken learner output. Van Rooy (2015, p.79) mentions three advantages of using learner corpora to do research in language teaching: size, variability, and automation. *Size* refers to the amount of data that can be processed (computerized corpus allows analyses of great amounts of data). *Variability* refers to the possibility of having more individuals and more text types to include in a corpus. This advantage is also linked to the possibility of having a computerized corpus. Finally, *automation* refers to some automatic aspects of data analyses possible thanks to information technologies (IT).

Corpus annotation is “the practice of adding interpretative, linguistic information to an electronic corpus of spoken and/or written language data” (Wynne, 2005, p.25). The added information comes in the form of tags, which can be defined as single entities added to one part or parts of the speech. Tags are unique and can identify features of the analyzed learner corpus. There are different types of annotation, and they require different tags depending on the goal of the researcher. For instance, descriptive linguistic uses Part of Speech (POS) tags to obtain grammatical annotation in a corpus. Another example is semantic annotation that requires assigning each word a semantic field used to do refined searches and classifications according to the research purpose. For error analysis, the annotation process is done to identify errors according to various categories and types.

To annotate errors, it is necessary to interpret learners’ choices and decide in what category the error best fits. This entails the construction of one or several target hypotheses that the researcher must test. It is impossible not to interpret data. Only through interpretation, the researcher will find ways to unhide possible hypothesis to do an essential analysis. Assigning a tag to an error means that it was the researcher’s interpretation, and that interpretation is publicly available for the reader. For that reason, when an error-tag is assigned, there could be other interpretations, but the most important is to keep uniformity in the way the tags are used. “The usefulness of error annotated corpora depends on the consistency on the annotation” (Ludeling & Hirschmann, 2015, p.148). Once the present learner corpus was annotated, it was easier to identify and extract data to analyse because the data was organized and ready to be used with software that permits further analyses.

For the present work, the learner corpus was tagged with a standardized error taxonomy that permitted the search and counting of errors analyzing within their context. The software used to extract error tags was WordSmith (Scott, 2005) and LancsBox. (Brezina *et al.*, 2015). WordSmith was used to obtaining the total statistics of errors, the dispersion,

and patterns that most affect the learner's production. LancsBox was used to obtain a more detailed profile of each error type and the corresponding graphics.

Regarding the annotation types in error analysis, there are two different types of annotation: *emendation* and *categorization* (Rosen *et al.*, 2012). In the first case, the researcher establishes one or more target hypotheses and does the correction according to the author's intention. On the other hand, the categorization is done following a previous established list of errors, because error annotation relies on error taxonomies and their categories for error classification. In the present work, after choosing a target hypothesis the researcher did an error categorization, adding predefined tags according to the Manual of Error Tagging from Louvain University version 1.2 (Dagneaux *et al.*, 2005). The corpus contained in the CLEC is a digital collection of 515 written files from English as a Foreign Language (EFL) university students registered in different careers. After the corpus was collected, the files were labeled. When an error was detected, the label was placed just before the error, and the correction followed the error between two-dollar signs: \$ correction \$ as the manual indicates:

Example:

Nowadays, we have seen (GADJN) differents \$different\$ (This error corresponds to the Grammar category and refers to the pluralization of an adjective (ADJN) in English).

The errors labeled and corrected in the CLEC are classified in the following eight categories that grouped a total of 56 error types. Please refer to appendix 1 of the present article to see the error types in detail.

- Form (F): groups the words used that do not exist in English and other errors of a formal type.
- Grammar (G): groups the errors that violate the general rules of English grammar.
- Lexical-grammar (X): errors where the morphosyntactic properties of a word are violated.
- Lexis (L): errors related to the semantic properties of words or sentences.
- Words (W): redundant words, missing words, or wrong word order.
- Punctuation (Q): errors related to punctuation marks.
- Style (S): incomplete sentences and unclear sentences.
- Infelicities (Z): registration problems (related to the field, the mode and the tenor of the speech) and issues of political correctness.

The next step after doing the error labeling was the extraction and alignment of the corpus. This process was carried out using an extraction software that searched for the labels and grouped them according to each error type. Tags were extracted within a context that granted proper analysis. The corpus's alignment was done using WordSmith, Scott, (2005) and LancsBox software, Brezina *et al.* (2015), which permitted the identification of language patterns obtaining statistics of the data with their respective graphs. After this process, the analysis of the findings took place.

3.2. Corpus metadata summary

The following are the main features of the corpus.

- Medium: written production
- Students belong to different university majors
- The EFL courses are 64 hours with an intensity of 4 hours per week for 16 weeks
- Native language of learners: Spanish
- Target language: English
- Genre of texts: there is a combination of genres between opinion paragraphs on different topics for level B1 and argumentative essays for level B2
- Tokens per text: at level B1 a maximum of 200, at level B2 up to 700
- Type: local corpus that seeks to identify needs and failures of learners
- Data compilation: it is a synchronous corpus with data collected in the second semester of 2015
- The incidence analysis was done by calculating the percentage of errors per 100 tokens to guarantee the proportionality of the analysis
- Corpus characteristics 149,325 tokens, 12,164 types and 12,337 lemmas

4. Methodology in the designing of the web application CLEC

After having the corpus collected and labeled with error tags, it was necessary to develop an application that systematically allowed the search of errors with the possibility to filter them according to different categories and types. It was also required that the app could allow changes in the error tags when they overlap among error categories. Therefore, a web application was developed with a frontend and a backend layer. After several tests, the functions of adding, modifying, or eliminating unnecessary data in the corpus were defined to be implemented. The development was possible thanks to a new technology where the frontend and backend responsibilities could be separated, and they were not

codependent. The alternative was a backend developed in Node.js (Dahl, 2009) together with Express.js (a web application framework for Node.js) for its construction as a REST API (Fielding, 2000) and a frontend in a JavaScript-based technology in which the options were React (Walke, 2013). It was decided to develop these technologies as they have excellent documentation and constant updates. Likewise, it was considered that the Node.js and React technologies have better support and a much broader community to guarantee a better response to the problems that arise throughout the development.

During the process, it was decided to use the persistence layer MongoDB (Merriman *et al.*, 2007) database management system (DBMS), which is document oriented because it is consistent with the data of the corpus in the present study. This DBMS allows efficient access when making inquiries. The structure shown in Figure 1, allows to store the contexts after being processed. In this structure, it can be observed how the data is organized by level, name of file, context, error type, and its correction.

```
{
  level:,
  name:,
  context:,
  errors: [{type:,
            error:,
            correction:,
            pos:
          }
        ]
}
```

Figure 1. Document structure in MongoDB.

After defining the technologies to use, the development of the backend started by developing the methods for the search of errors. The additional services were defined and developed to enable the functions to create, read, modify, and delete contexts and create, read, and delete errors.

In this case, the method for modifying errors was left out as this meant an unnecessarily large load for processing due to the data's nature. Instead, it was decided to leave this functionality implicit as a combination of elimination and addition of errors. The database of contexts was populated with the help of preprocessing Python scripts that allowed structuring the data in the way it was previously defined. The new method of creating contexts included all this preprocessing that was required for new contexts.

In Figure 2, it is shown the architecture of the system described above.

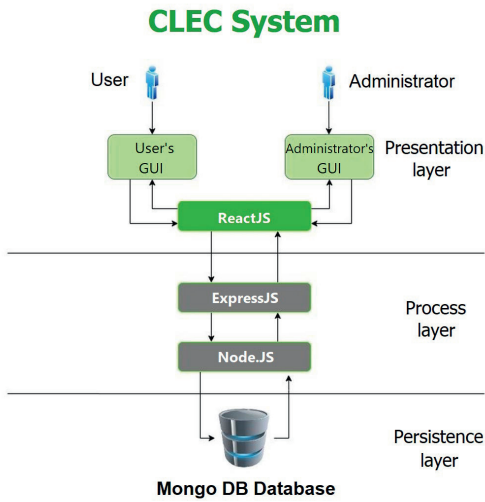


Figure 2. CLEC System Architecture.

As may be observed in Figure 2, the proposed system has two roles: administrator and user. The administrator can modify the application's data, whereas the user can only use the application. The most important use cases for both administrator and users are shown below in figure 3 and 4, respectively.

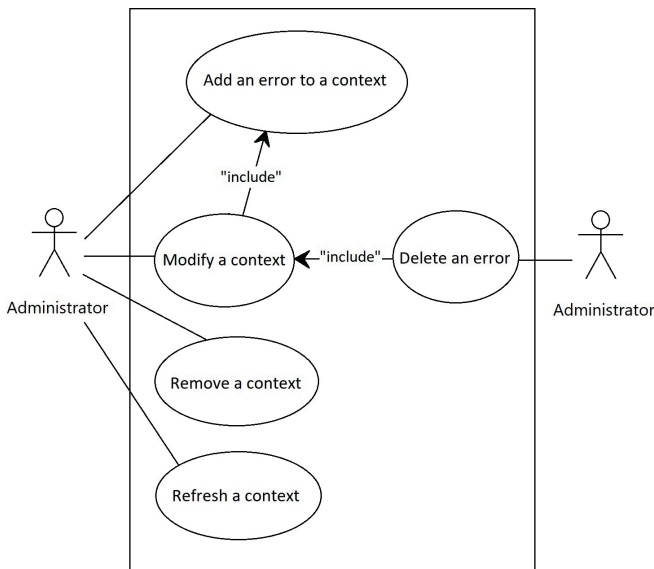


Figure 3. The administrator's use cases.

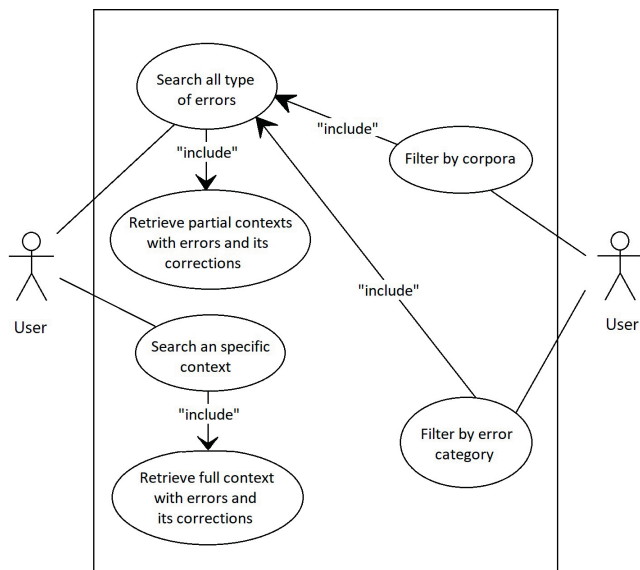


Figure 4. Use cases available for all users.

Each of the use cases depicted above will be illustrated below.

There were two ways to obtain the text contexts, one that displayed all the texts for a general view of different errors within their contexts, and one that obtained a specific text for a detailed view of each error within its context. Let us see the general view of different errors in Figure 5.

UNIVERSIDAD DE ANTIOQUIA GRUPO FIN		C L E C COLOMBIAN LEARNER ENGLISH CORPUS		Corpus	Contexts	Tags	Credits	Login
Complete Corpora		All error categories	100 results	5 tokens	Reset search			
Error type	Error	Correction	Actions					
LS	the place because the place no is is not from	the place because the place not is is not from	GO TO CONTEXT					
WO	because the place not is no is from their country.	because the place not is is not from their country.	GO TO CONTEXT					
LS	not is is not of their country.	not is is not from their country.	GO TO CONTEXT					
QM	We o as people in the world	We , as people in the world	GO TO CONTEXT					
WO	the places although the places not are from our country.	the places although the places are not from our country.	GO TO CONTEXT					

Figure 5. General view of different error types with their corrections (Pardo *et al.*, 2018)

In Figure 5, for every sentence, it can be observed at the right side of the menu a button link that redirects the search to see each error's whole context. Clicking that button implies seeing the text's whole context that contains the error mentioned at the left side of the sentence. When you hit the button "go to context," you will see what is shown in Figure 6, the same error within the full context, and the correction in green.

The screenshot shows the CLEC interface with a text passage and its corrections. The interface includes the Universidad de Antioquia logo and navigation links: Corpus, Contexts, Tags, Credits, Login.

Original Text (Left Panel):

Final Work: In my opinion, Commercials on TV are not honest. First, **Commercials** do not show the disadvantages of the product. For example, **Commercials** about fast food ; **they** just say **the** fast food is delicious, but they do not say, fast food is not healthy. Second, **Commercials** do not say the truth. For example, Mc Donald's commercial shows the biggest hamburguers of the world, but it is a lie, their hamburguers are small. Finally, **Commercials** are bad for your economy. For example, the bank's commercials, **they** just want to convince you **0** get a credit card, for this, they make **offerts for** you can get **it** . In conclusion, Do not let **Commercials** convince you about any product; if you are **interesting** in something, ask for it. Commercials are dangerou

Corrected Text (Right Panel):

Final Work: In my opinion, Commercials on TV are not honest. First, **commercials** do not show the disadvantages of the product. For example, **commercials** about fast food **00** just say **0** fast food is delicious, but they do not say, fast food is not healthy. Second, **commercials** do not say the truth. For example, Mc Donald's commercial shows the biggest hamburguers of the world, but it is a lie, their hamburguers are small. Finally, **commercials** are bad for your economy. For example, the bank's commercials, **0** just want to convince you **to** get a credit card, for this, they make **offers so** you can get **them**. In conclusion, Do not let **commercials** convince you about any product; if you are **interested** in something, ask for it. Commercials are dangerou

Figure 6. View of errors with full context and corrections (Pardo *et al.*, 2018).

Considering the nature of the data and these functionalities, the possibility of modifying contexts only to the parts of each text that did not contain errors was added. This was done in case the researcher wants to focus only on the text with errors. There were two methods to achieve this goal, one that creates lists of both context parts that contained and did not contain errors, and a second method that receives similar lists with the modifications made.

Similarly, the services corresponding to creating, reading, and eliminating errors were developed. All of them included verifications so that the rest of the errors did not enter conflict for their positions and/or for their content. For this part of the process, the service to modify errors was left out because it resulted in multiple cases in which some verifications of the data required excessive processing. This was replaced by a new possibility to modify errors by eliminating a previous error and adding a new one. It was an easier function, both for the development process and for the end-user.

Down, on the right side of Figure 7, 4 buttons allow changes in the corpus: add error, modify context, remove context, and refresh context.

The screenshot shows the CLEC interface. At the top left is the logo of Universidad de Antioquia. The main header contains 'CLEC' and 'CORPUS CONTEXTS TAGS CREDITS SIGN OUT'. Below the header are two text boxes. The left box contains a paragraph with several errors highlighted in red: 'no is no is of', '0', 'not are of', 'the', 'visit', and 'tourist don't care the environmen'. The right box contains the same paragraph with corrections highlighted in green: 'not is is not from', '0', 'are not from', 'a', 'visits', and 'tourists don't take care of the environment'. Below the text boxes is a control bar with four buttons: 'ADD ERROR' (green), 'MODIFY CONTEXT' (yellow), 'REMOVE CONTEXT' (red), and 'REFRESH CONTEXT' (black).

Figure 7. View of buttons to make modifications in the corpus.

These new functionalities are a plus in case there is need for a more detailed work in the corpus or to focus on specific parts of the texts.

A view of the search filters can be viewed in Figure 8. These filters were grouped by level: the corpus was divided into 4 levels of English A1, A2, B1, B2. They were arranged in an element of type selected:

- Basic (A1)
- Pre-intermediate (A2)
- Intermediate (B1)
- Advanced (B2)

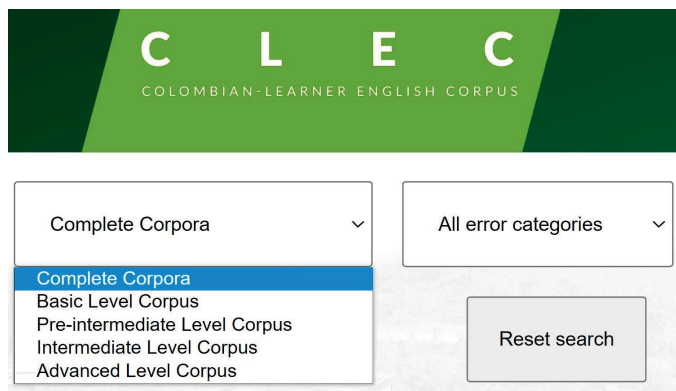


Figure 8. View of levels in the corpus.

In Figure 9, it can be noticed how the error types explained in the corpus collection section of this article were arranged as an element of type select.

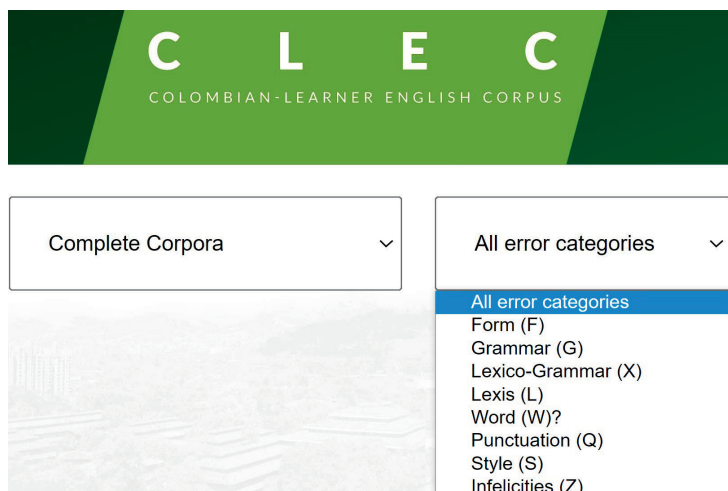


Figure 9. View of error categories (Pardo *et al.*, 2018).

In Figure 10, it may be noted how a condition was created so that check boxes with the corresponding class error types would be displayed when the selection was changed. In all this process, it can be noted how the system's graphic design was created, selecting the university's institutional colors (dark and light green).

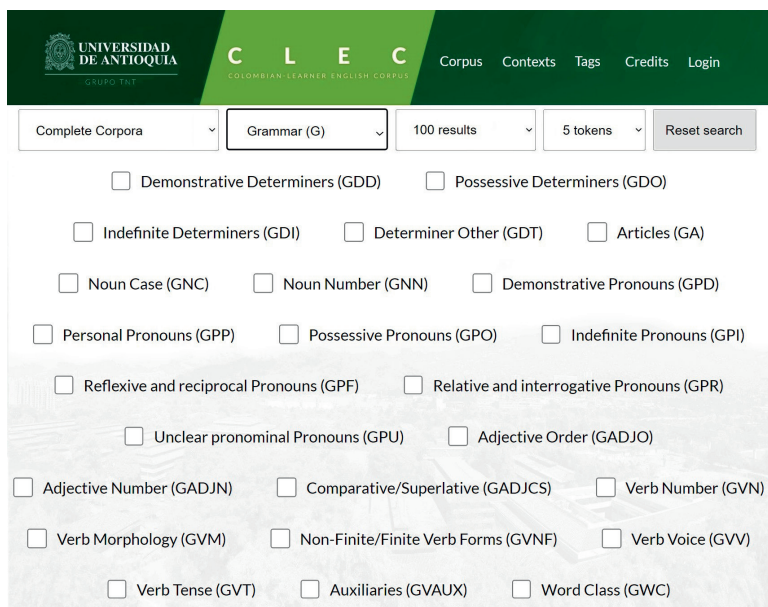


Figure 10. Check boxes to choose error types to analyze in the Grammar category.

In this case, Figure 10 shows error types from the grammar category, but if the category changes, the error types will correspond to the chosen category.

In Figure 11, it is possible to observe errors within the context of one sentence. The errors are in red and in front of the whole text with the corrections in green.

Error type	Error	Correction	Actions
GADJO	If the system judicial in Colombia Guilty severely punishes	If the judicial system in Colombia Guilty severely punishes	GO TO CONTEXT
GADJO	0 people like see 0 commercials fabulous .	0 people like see 0 fabulous commercials .	GO TO CONTEXT
GADJO	Fraud is a crime serious .	Fraud is a serious crime .	GO TO CONTEXT

Figure 11. View of errors within a small context.

The same errors can be viewed in the whole context when hitting the button “go to context.” In Figure 12, we may note the view of the whole context for one of the errors.

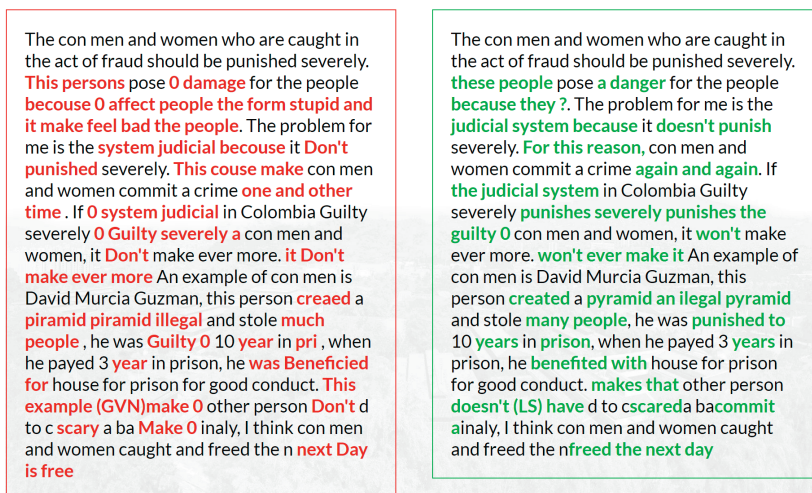


Figure 12. View of errors in one file.

It is necessary to clarify that the view of errors in Figure 12 shows all the different errors the student made in his composition, for that reason, there are several categories and types of errors.

All the previous functionalities were oriented for the use of all users, including unauthenticated ones. For authenticated users (administrator role), additional components were made available for the other functionalities, including a button, in the context view, for each error that would allow the possibility to eliminate them if necessary. Let us see the detail in Figure 13.

Type	Error	Correction	
GDD	This	these	DELETE ERROR
FS	persons	people	DELETE ERROR
GA	0	a	DELETE ERROR

Figure 13. View of the button to delete errors (Pardo *et al.*, 2018).

Besides, a set of buttons were included at the bottom of the whole contexts, and the buttons are: Add, Modify, Remove and Refresh. By displaying a pop-up window, the user selects

the context section on which he/she wants to introduce a modification. The same process is followed for each case. There is another button to remove the context and the last button to refresh the context with the changes made. Let us see Figure 14.

The screenshot shows the CLEC web interface. At the top, there is a navigation bar with the logo of Universidad de Antioquia and the text 'CLEC COLMBIAN LEARNER ENGLISH CORPUS'. Below the navigation bar, there are two context panels. The left panel shows a paragraph of text with several errors highlighted in red. The right panel shows the same paragraph with corrections highlighted in green. Below the context panels, there are four buttons: 'ADD ERROR', 'MODIFY CONTEXT', 'REMOVE CONTEXT', and 'REFRESH CONTEXT'. Below the buttons, there are two tables. The first table has columns 'Type', 'Error', and 'Correction'. The second table has columns 'Type', 'Error', and 'Correction'.

Type	Error	Correction
GA	The	0

Type	Error	Correction
GWC	advertisements	advertising

Figure 14. View of full contexts and buttons to add, modify and remove data (Pardo *et al.*, 2018)

5. Results

From the previous process, the result was a web responsive application that completely performs searches and does analysis on the tagged corpus of errors. This app contains a learner corpus of English as a Foreign Language (EFL) learners that has the potential of being easily revised and expanded through the role of the system administrator. This new functionality will be very useful to enrich the system that can be used by linguists, teachers, and students who may consider it to do research. This corpus is available in the given URL

and is open to any researcher if you want to consult it or if you want to contribute with learner corpora².

The development of the backend as a REST API allowed the tests to be carried out independently of the frontend, allowing future developers to use this API for new versions or refactoring of the frontend.

Regarding the front end, it was also possible to deliver a design that is very aesthetic and friendly. This will allow that existing method and those that would be open to the public were simplified and more understandable for use.

Finally, the web application was deployed on the Translation and New Technologies (TNT) research groups of Universidad de Antioquia server. The Colombian Learner English Corpus (CLEC) is available online at: <https://grupotnt.udea.edu.co/clec>.

5.1. Graphical view of errors

The findings of errors in the corpus were grouped by category and type. Figure 15 shows a view of errors by category.

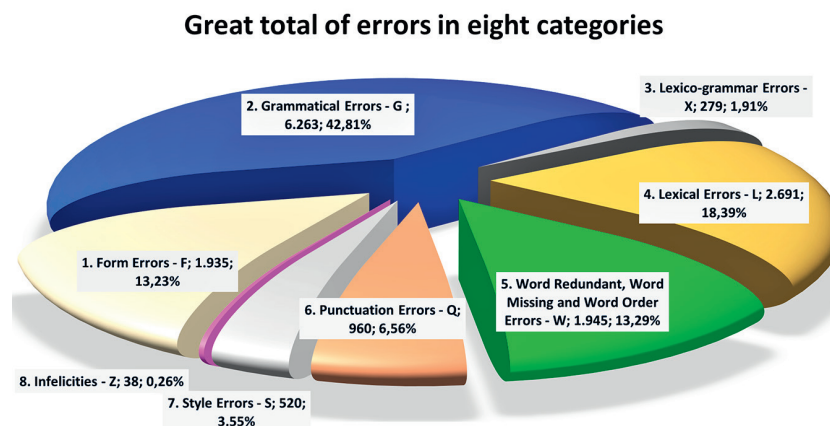


Figure 15. Incidence of errors by category (Pardo, 2019).

It is clear in figure 15 that the category of errors with most frequency in the corpus was Grammar. A more detailed view of errors is displayed by type in Figure 16.

² If you want to contribute with data to this project, please contact the authors.

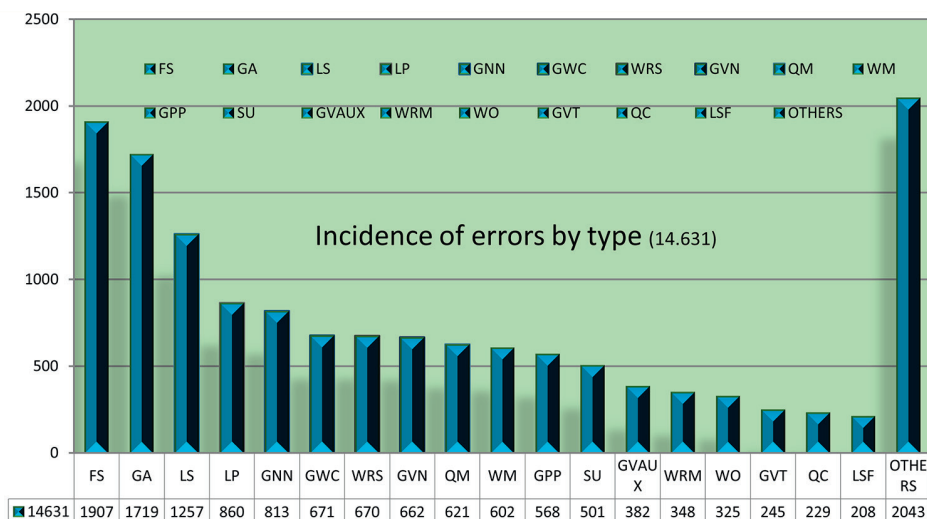


Figure 16. Incidence of errors by type (Pardo, 2019).

In this case, the frequency by type can give us an idea of the frequency of each type of error. All this information can be easily retrieved for its analysis using the CLEC app.

6. Conclusions

This work presented the CLEC app, the first corpus of written production of Colombian students learning English as a Foreign Language available online for the research community. CLEC works with a modern technology that offers agile maintenance options and allows a user interface design that is friendly and allows a satisfying interaction with the app.

Similarly, it was possible to achieve the construction of a complete, friendly, and safe administration system to manage the data of the treated corpus allowing its scalability and maintenance to create, read, edit, and eliminate contexts. These functions give the application an invaluable utility for didactic and research matters.

There were several advantages brought with the technologies used in this project. Using React, future development teams will be able to take over the project and add new functionalities.

Despite the complexity of the structure in which the contexts and errors were handled, it was possible to reduce the complexity of the entire process for the end-user through the correct planning of the development and the views. Now it is an interface that allows the use of its features in a practical way.

Finally, this work gives the academic community an invaluable free access web application, which facilitates the teaching-learning process of English as a foreign language through an efficient and friendly error analysis.

Acknowledgements

Thanks to Universidad del Norte for allowing the collection of the data.

We would like to acknowledge Manuel Gómez and Nicolás Henao for their participation in the design of the CLEC app.

The research reported here was supported by a COLCIENCIAS scholarship.

Appendix

1. Error categories and types according to the manual of Louvain University

FM	Form, Morphology
FS	Form, Spelling
FSR	Form, Spelling, Regional
GDD	Grammar, Determiner, Demonstrative
GDO	Grammar, Determiner, Possessive
GDI	Grammar, Determiner, Indefinite
GDT	Grammar, Determiner, Other
GA	Grammar, Articles
GADJCS	Grammar, Adjectives, Comparative / Superlative
GADJN	Grammar, Adjectives, Number
GADJO	Grammar, Adjectives, Order
GADVO	Grammar, Adverbs, Order
GNC	Grammar, Nouns, Case
GNN	Grammar, Nouns, Number
GPD	Grammar, Pronouns, Demonstrative
GPP	Grammar, Pronoun, Personal
GPO	Grammar, Pronoun, Possessive
GPI	Grammar, Pronoun, Indefinite
GPF	Grammar, Pronoun, Reflexive/Reciprocal
GPR	Grammar, Pronoun, Relative/ Interrogative
GPU	Grammar, Pronoun, Unclear reference
GVAUX	Grammar, Verbs, Auxiliaries
GVM	Grammar, Verbs, Morphology
GVN	Grammar, Verbs, Number
GVNF	Grammar, Verbs, Non-Finite / Finite
GVT	Grammar, Verbs, Tense
GVV	Grammar, Verbs, Voice
GWC	Grammar, Word Class

LCC	Lexis, C onjunctions, C oordinating
LCLC	Lexis, C onnectors, L ogical, C omplex
LCLS	Lexis, C onnectors, L ogical, S ingle
LCS	Lexis, C onjunctions, S ubordinating
LP	Lexical P hrase
LPF	Lexical P hrase, F alse friends
LS	Lexical S ingle
LSF	Lexical S ingle, F alse friends
QC	Punctuation, C onfusion
QL	Punctuation, Lexical
QM	Punctuation, M issing
QR	Punctuation, R edundant
SI	S entence, I ncomplete
SU	S entence, U nclear
WM	W ord M issing
WO	W ord O der
WRS	W ord R edundant S ingle
WRM	W ord R edudant M ultiple
XADJCO	LeXico-Grammar, A djectives, C omplementation
XADJPR	LeXico-Grammar, A djectives, D ependent P reposition
XCONJCO	LeXico-Grammar, C onjunctions, C omplementation
XNCO	LeXico-Grammar, N ouns, C omplementation
XNPR	LeXico-Grammar, N ouns, D ependent P reposition
XNUC	LeXico-Grammar, N ouns, U ncountable / C ountable
XPRCO	LeXico-Grammar, P Repositions, C omplementation
XVCO	LeXico-Grammar, V erbs, C omplementation
XVPR	LeXico-Grammar, V erbs, D ependent P reposition
Z	I nfelicities

— *References*

- Brezina, V., McEnery, T., & Wattam, S. (2015). Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics*, 20(2), 139-173. <https://doi.org/10.1075/ijcl.20.2.o1br>
- Carlstrom, B., & Price, N. (2012). *The Gachon Learner Corpus*. Retrieved from <https://app.box.com/s/erq3w1d7v71fq5ze76kzt56lomk3c06>
- Corder, S. (1967). The significance of learner's errors. *IRAL - International Review of Applied Linguistics in Language Teaching*, 5(1-4), 161-170. <https://doi.org/10.1515/iral.1967.5.1-4.161>
- Corder, S. (1981). *Error Analysis and Interlanguage*. Oxford University Press.
- Dagneaux, E., Denness, S., Granger, S., Meunier, F., Neff, J., & Thewissen, J. (2005). *Error Tagging Manual Version 1.2*. Centre for English Corpus Linguistics, Université Catholique de Louvain.
- Dahl, R. (2009). *NODE.JS*. Open JS Foundation. <https://nodejs.org/es/docs/>
- Diaz-Negrillo, A. (2009). *EARS: a User's Manual*. Lincom Academic Reference.
- Diaz-Negrillo, A. (2012). Learner corpora: the case of the NOSE corpus. *Journal of Systemics, Cybernetics and Informatics*, 10(1), 42-47. <https://www.iiisci.org/journal/pdv/sci/pdfs/HEB467AV.pdf>
- Europe, C. of. (2001). The Common European Framework of Reference for Languages: Learning, teaching, assessment. *Common European Framework*. <https://doi.org/10.1093/elt/ccii05>
- Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures* [Doctoral dissertation, University of California, Irvine]. Donald Bren School of Information and Computer Sciences. https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- Gilquin, G. (2015). From design to collection of learner corpora. In S. Granger, G. Gilquin, & F. Meunier (Eds.), *The Cambridge handbook of learner corpus research* (pp. 9-34). Cambridge University Press.
- Granger, S., Gilquin, G., & Meunier, F. (Eds.). (2015). *The Cambridge Handbook of Learner Corpus Research*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139649414>
- Granger, S. (2002). A Bird's-eye view of learner corpus research. In S. Granger, J. Hung, & S. Petch-Tyson (Eds.), *Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching* (pp. 3-33). John Benjamins Publishing Company.
- Granger, S. (2003). The International Corpus of Learner English : A New Resource for Foreign Language Learning and Teaching and Second Language Acquisition Research. *TESOL Quarterly*, 37(3), 538-546.
- James, M. (1980). *Managing the database environment*. Savant Research.
- Ludeling, A., & Hirschmann, H. (2015). Error annotation systems. In S. Granger, G. Gilquin, & F. Meunier (Eds.), *The Cambridge handbook of learner corpus research* (pp. 135-157). Cambridge University Press.
- McEnery, A., & Hardie, A. (2011). *Corpus Linguistics: Method, theory and practice*. Cambridge University Press.
- Mendikoetxea, A., O'Donnell, M., & Rollinson, P. (2009). *WriCLE: A learner corpus for Second Language Acquisition Research*. 2010. http://ucrel.lancs.ac.uk/publications/cl2009/351_FullPaper.doc
- Merriman, D., Horowitz, E., & Ryan, K. (2007). *MongoDB Documentation*. <https://docs.mongodb.com/>
- Pardo, M. (2019). *Error Analysis in a Written Corpus of Spanish Speakers EFL Learners. A Corpus-based Study*. Universidad de Antioquia.

- Pardo, M., Quiroz, G., Tamayo, A., Henao, N., Ortega, M., & . (2018). *CLEC Colombian Learner English Corpus*. <https://grupotnt.udea.edu.co/clec/corpu>
- Rosen, A., Jirka, H., Stindlová, B., Feldman, A., & Svatava, S. (2012). Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation*, 48(1), 65-92. <https://doi.org/10.1007/s10579-013-9226-3>
- Scott, M. (2005). *WordSmith*. Lexically. <http://lexically.net/wordsmith/research/>
- Selinker, L. (1972). Interlanguage. *International Review of Applied Linguistics in Language Teaching*, 10(1-4), 209-232.
- UC Louvain. (2018). *Centre for English Corpus Linguistics*. Learner Corpora Around the World. <https://uclouvain.be/en/research-institutes/ilc/cecl/learner-corpora-around-the-world.html>
- University, T. S. (2002). *The Santiago University Learner of English Corpus (SULEC)*. <https://sulec.cesga.es/>
- Van Rooy, B. (2015). Annotating learner corpora. In S. Granger, G. Gilquin, & F. Meunier (Eds.), *The Cambridge handbook of learner corpus research* (pp. 79105). Cambridge University Press.
- Walke, J. (2013). *React. Una biblioteca de JavaScript para construir interfaces de usuario*. React. <https://es.reactjs.org/>
- Wynne, M. (Ed.) (2005). *Developing linguistic corpora: a guide to good practice*. Oxbow Books.