DESIGN OF A MATHEMATICAL TEXT ANALYSIS TOOL USING MATLAB FOR A STUDY OF LEXICAL ERRORS / UNA HERRAMIENTA MATEMÁTICA DE ANÁLISIS DE TEXTO EN MATLAB PARA EL ESTUDIO DE ERRORES LÉXICOS[1]

DOI: 10.5281/zenodo.10407631

***Abstract**: The study of the process of foreign language acquisition can be carried out from different and varied perspectives. In this case, the aim is to analyse a corpus of learners of English as a foreign language. In order to achieve this objective, it is necessary to have at least two tools at our disposal: a corpus representative of a given population and a computer application that provides an answer to our research questions. In our case, we have the Canary Islands Corpus of Written English (Wood et al., 2004) but we do not have a computer application that would allow us to analyse student errors in this corpus. To solve this problem, we have opted to use the Matlab programming and numerical calculation platform. This programming platform is increasingly being used by scientists and engineers, while its use in the humanities is progressing more slowly. Taking into account that Matlab is a programming platform, the first thing we have done is to develop a code that we make freely accessible, as well as detailing and exemplifying the functions that this programme offers for an error analysis.*

***Keywords**: foreign language acquisition, error analysis, learner corpus, Matlab, free access code*

***Resumen**: El estudio del proceso de adquisición de una lengua extranjera se puede realizar desde distintas y variadas perspectivas. En este caso el objetivo es hacerlo desde el análisis de un corpus de aprendices de inglés lengua extranjera. Para cumplir este objetivo es necesario disponer de al menos dos instrumentos: un corpus representativo de una población determinada y una aplicación informática que dé respuesta a nuestras preguntas de investigación. En nuestro caso disponemos del Corpus Canario de Inglés Escrito (Wood et al., 2004), pero no de una aplicación informática que nos permita analizar los errores del alumnado en dicho corpus. Para resolver este problema, hemos optado por usar la plataforma de programación y cálculo numérico Matlab. Esta programación está siendo usada cada vez más por científicos e ingenieros, mientras que su uso en el campo de las humanidades avanza más lentamente. Teniendo en cuenta que Matlab es una plataforma de programación, lo primero que hemos hecho es desarrollar una programación que ponemos en acceso libre, además de detallar y ejemplificar las funciones que dicha programación ofrece para un análisis de errores como eslabón del proceso de adquisición.*

***Palabras clave**: adquisición de una lengua extranjera, análisis de errores, corpus de aprendices, Matlab, código de acceso libre*

1. Introduction

The description of learner language is one of the fields of study in research related to foreign language acquisition (Ellis, 2008: 5). Within this field, the analysis of errors and their evolution as the acquisition process unfolds has been of particular relevance. Several paradigms were used in the second part of the last century: contrastive analysis (Lado, 1957), error analysis (Corder, 1967) and interlanguage (Selinker, 1972). In the present century, we find that studies related to errors continue to be carried out. The differentiating aspect is that we have now opted for the performance analysis model, in which errors are taken into account as well as correct productions of the language. Studying both aspects allows us to analyse how the learner's interlanguage develops. An

[1] Begoña CAMPOS ESTARÁS, Universidad de Las Palmas de Gran Canaria, Spain, begona.campos101@alu.ulpgc.es; Adrian PENATE-SANCHEZ, Universidad de Las Palmas de Gran Canaria, Spain, adrian.penate@ulpgc.es

example of this type of analysis can be found in De Alba Quiñones (2008) and Alexopoulou (2011)

Obviously, over the last 60 years, multiple typologies have been created and modified, but despite this there is general agreement about classifying errors, from the point of view of their cause, into several main categories (Navarro Rodríguez, 2008: 59). These categories are those that arise from the following three types of errors: interlanguage errors due to the influence of the mother tongue, intralingual errors due to the influence of the target language, and errors produced by the use of communication or compensation strategies.

Interlingual errors are perhaps the best known since they are considered to be the result of discrepancies between the mother tongue and the foreign language. Intralingual errors have their origins in the incomplete application of a grammar rule, ignorance of the exceptions to that rule, overgeneralisation or incorrect hypothesis. Errors caused by communication or compensation strategies to fill in a linguistic gap may result in the coining of new terms in an attempt to give them an appearance that matches the foreign language, circumlocution, use of mother tongue, etc. (Oxford, 2016; Oxford and Amerstorfer, 2018).

In the category of interlinguistic errors, we find spelling mistakes which are usually given little importance as they are considered merely formal errors or slips of the tongue. They are considered as such because they are based on the idea that the learner learns the word by rote and then has to write it correctly. However, the approach changes substantially when students know the word orally and gradually internalise the English spelling by establishing hypotheses, meaning that the error becomes intralingual. This paradigm shift has also occurred in English as a mother tongue (Kolodziej and Columba, 2005). It is precisely this point of view that justifies the inclusion in our corpus of the error next to the correct word in order to be able to carry out statistical analyses and detect which type of error is involved in each case.

From a methodological point of view, error studies are also related to corpora of learners of a foreign language, which, in turn, have their origins in error analysis studies carried out in the last century before the generalisation of computer media (Granger, 1998). The corpus of learners (Sánchez Rufat, 2015) is framed within the linguistic corpora (McEnery and Hardie, 2011) and both, thanks to computer technology, have undergone a very important development over the last twenty years or so, being able to analyse millions of words (McEnery et al., 2019: 74).

Already at the beginning of this century, there were authors who warned that language corpora were based on native speakers and did not respond to the needs and realities of foreign language learners (Grant and Ginther, 2000: 141; Granger, 2012). Therefore, the first learner corpus was created in 2002, namely the ICLE-International Corpus of Learner English (Granger et al., 2015). Subsequently, corpora of learners of different foreign languages have been created: LINDSEI Louvain International Database of Spoken English Interlanguage (Gilquin et al., 2010), CAES Corpus de aprendices de español (Instituto Cervantes, 2015), etc.

In this article, we first explain the corpus of learners we have used. We then list the problems with this corpus and its software application (*FreconWin*). The rest of this article is devoted to explaining the modifications made to this corpus and the software tool we have developed using *Matlab* in order to obtain the data relevant to the objective of our subsequent study: the analysis of lexical errors in primary and secondary school pupils in the Canary Islands. In short, we have come to respond to a need already raised almost 20 years ago, which indicated that the researcher on corpus analysis should have the ability to write at least partially his own computer program (Conrad, 2002: 87).

2. The original corpus and its computer processing

Thanks to the collaboration with the Government of the Canary Islands, it was possible to compile this corpus, which presents a statistically representative sample of pupils at the end of Primary Education, Compulsory Secondary Education (CSE) and Baccalaureate; in other words, students who completed each of the three pre-university educational stages. The tool used was a written essay with topics appropriate to the age and level of the pupils.

As we have already indicated, the corpus includes a statistically representative sample of pupils from the seven Canary Islands. For the sixth year of primary school, the population was 1521 pupils, the required sample being 317 pupils. However, as there was awareness that the size of the written productions at this level was small, it was decided to increase the sample by 50%, resulting in a total of 475 essays for this educational level. For the fourth year of CSE, the population was 2076, and the required sample was 335 essays. Finally, there was a population of 1940 Baccalaureate students, so the required sample was 332 essays.

The creation of this corpus of written English was motivated by the interest in carrying out a lexicometric study at the three specified educational levels. This objective meant that a number of decisions had to be taken when making the transcripts in order to avoid distorting the lexicometric data.

To this end, the *FreconWin* programme (Guerrero Martín, 1991) was used because all words between square brackets are not taken into account. This function was used for each of the lexical errors, thus avoiding the distortion of the lexical richness measures. If this function had not been used, each lexical error would have been counted as a new word, which could result, for example, in primary students having a larger lexicon than the rest of the levels. The square brackets were also used so that personal data (names of pupils or their families, place of residence, etc.) were not considered.

In 2004, a first study of this corpus was published (Wood et al., 2004). It can be divided into two clearly differentiated parts. In the first part, the lexical forms that appear in the corpus are established and presented in two formats: in lexicographical order and in lexicometric order. In the second part of the study, the *FreconWin* programme carries out a study of the specificity of lexical forms, both positive and negative, for each of the three parts of the corpus.

3. Problems with the original corpus and its computer processing

As we have already indicated above, the *FreconWin* programme provided us with two types of lists: lexicographic and lexicometric. The former lists all the lexical forms appearing in the corpus, arranged in alphabetical order, and specifying the total number of occurrences. In the lexicometric list, the lexical forms are ordered according to the total number of occurrences. However, several problems arise when we try to work with these two lists. When analysing the lexicographic list, we found out that, of the total of 3186 different lexical forms in this corpus, only 726 have at least 10 occurrences but when we analysed the lexicographic list, we could see that since we had used the lexical form as unit of analysis, the programme did not allow us to add lexical forms with a common root, as can be seen in Table 1.

Lexical form	*Total number of occurrences*
affect	34
affected	44
affecting	2
affects	1

Table 1: Example of lexical forms with a common root

It is clear from this that these data are scattered and would leave out the lexical forms "affecting" and "affects" because they do not reach the minimum number of 10 occurrences mentioned above. In order to solve this problem, we decided to make a software tool that allows using the "lemma" as the unit of analysis, that is, "the uninflected form of a word that is likely to serve as a headword in a typical dictionary entry". (Kusseling and Lonsdale, 2013: 442)

Moreover, Table 1 does not provide information on the grammatical category of each of the lexical forms. This will force us to disambiguate the corpus, but doing it manually does not seem to be a solution in keeping with the times we live in.

Another problem we encountered when using the *FreconWin* application was that the lexical forms which were spelt incorrectly were transcribed in double brackets, the word spelt correctly being placed just before them. This is positive in that, as noted above, it does not give incorrect information about the lexical richness of the learner, which would be the case if each error were considered as a separate lexical form. It is also positive because it allows to see the types of lexical errors made. However, the software cannot provide any data on these errors, as they are not taken into account. In other words, when this corpus of learners was created, decisions were made that helped to provide answers to the questions of interest (lexical richness in this case), but they do not help with aspects that were not anticipated and are of interest to us now. (McEnery et al., 2019: 80)

4. Objective

Our aim is to develop a software tool that allows different analyses of a corpus with errors. Specifically, we expect it to offer the possibility to: disambiguate, lemmatise, analyse the frequency of words and lemmas, establish the distance between the correct word and the error, and define the importance of each word in its document.

5. Modification of the learner corpus and development of our *MATLAB*-based software application

In this section we describe and document the work done to prepare and analyse the Corpus of Primary, Secondary and Baccalaureate students. The different parts of the software development we have carried out will be outlined to show the possibilities presented by using our system. In particular, we will describe the functionalities developed using *MATLAB*. There are several sources for getting started in *MATLAB*-based development, especially tutorials offered by the company itself and, and more specifically for this area. In other words, introductory books on *MATLAB* for researchers (Madan, 2013).

MATLAB is not a closed tool that provides functionality as is common in computing tools used by academics in the humanities. *MATLAB* is more like a programming language from which to develop a software application. The code we have created for this tool is freely available on Github (https://github.com/andarinneo/textProcessingHumanities).

MATLAB's interface is separated in two main parts: the console window and the editor window; examples of the console and editor can be seen in the tutorials provided for MATLAB[1,2]. In the console window we can execute and control the flow of how we execute the code scripts that implement the functionality that we have developed. In the editor window we write the code that becomes the software tool that we will use for the study of the Learner Linguistic Corpus. *MATLAB* is a better option for research tasks in

[1] https://es.mathworks.com/help/matlab/learn_matlab/desktop.html
[2] https://es.mathworks.com/videos/using-the-live-editor-117940.html

the humanities compared to a classical programming language such as C++ or Java, and there are three reasons for this: 1. Rapid prototyping, 2. *MATLAB* function libraries that support text analysis and 3. The general versatility for development with high uncertainty as is the case with research tasks. Also, as *MATLAB* has a much simpler syntax and general functioning, it allows researchers that are not proficient in programming to modify and add functionality to it. To facilitate the comprehension of *MATLAB* for those not familiar with it, we provide a link to the basic guide that pinpoints its basic elements (see footnotes 2 and 3).

5.1. Corpus Cleaning tasks

Before proceeding to develop the tool that will analyse the text, the corpus had to be modified. This is because the annotations, marks and other miscellaneous symbols previously entered did not fit into regular expressions (Kleene and Shannon, 1951). Such regular expressions are required to allow an automatic system to process the text and create the data structures required for text processing.

Examples of how the text had to be modified include: removal of "+" identifiers, removal of carriage returns (ASCII decimal code 13) in students' texts, separation of words that had been transcribed without leaving a space between them, disambiguation of double-bracket identifiers to identify whether they represent an error, anonymization of personal data or a location.

5.2. *Parsing* of the Linguistic Corpus

Once the spurious symbols that had been introduced into the Corpus have been corrected, it is possible to create the code that will break the text into its different parts and note the corrections made, as well as the original form of the words before the correction. This process requires what is known in computer science as “Grammatical Analysis”, commonly known in English as Parsing. We have chosen to use the colloquial term to avoid confusion when referring to the process we have carried out.

In the context of our work processing the Corpus, “Grammatical Analysis” or “Parsing” simply consists in being able to automatically separate each student’s essay. All essays are contained in a single Word document. First, all PRIMARY students’ essays can be found with a title that indicates that these are the primary school students’ texts. Secondly, we find the SECONDARY students’ essays, and finally, all the BACCALAUREATE students’ essays. All essays with a student ID number. What “Parsing” does is simply break down the Word document into each individual essay and properly annotate the student’s ID and the grade to which it belongs (PRIMARY, SECONDARY, BACCALAUREATE).

The grammatical analysis (parsing) to be carried out consists of processing the parts that make up the text as a Chomsky type 3 grammar (Chomsky, 1956). In this grammar created by us the parts will be: PRIMARY, SECONDARY and BACCALAUREATE. These in turn are divided into the texts of each student, each text consisting of a header and a body. The header identifies the centre/province and the student. The body is the text in question which we seek to analyse; the body in turn may contain elements marked as special which the grammar must capture. The body of the text that had the special elements removed is hereafter known as a document.

There are several details that need to be detailed to fully understand how we work automatically processing the Corpus:

1. Lexical errors are tagged by hand by a teacher, who then annotates the error using double brackets, plus a specific letter code, so the original form and the corrected form can be automatically processed by our software code.
2. We include corrected words to make it possible to analyse how errors are committed within the whole Corpus.

3. Parsing is needed to be able to fully automate the processing of the whole Corpus. This will allow the possibility to produce elaborate statistical analysis as we will show later in the paper.
4. In order to process the document as a Chomsky type 3 grammar, composed of the previously described elements, we have made use of regular expressions (Kleene and Shannon, 1951). Once the document has been processed, a data structure containing the following has been created:
   Grade: Primary/Secondary/Baccalaureate
   School ID/Province: Identifier of the school of origin for Primary and Secondary students, and the province of origin for baccalaureate students.
   Student ID: Identifier of the author of the text.
   Header: Original text of the header of the text.
   Body: Original text of the body of the text.

This first data structure does not yet contain the student's text in a state that allows processing. It is from this first structure that a more sophisticated analysis of the text for pre-processing begins.

5.3. Pre-processing

This pre-processing consists of the following phases: **1** Tokenisation, **2** Phrase detection, **3** Punctuation removal, **4** Disambiguation of discourse elements, **5** Obtaining the morphological Theme (Stem) of each word and **6** Lemmatisation of each word.

5.3.1. Tokenization

The tokenisation process consists of automatically detecting each of the atomic elements that make up the text. The methodology we have used is the one defined in David and Chapman (2020) and in the standards defined in UNICODE (ICU, 2020). At the same time, we have modified all the proper names, which for privacy reasons cannot be shown, and have been changed to the proper name LENNON. The letters "x" and "p" that we can see between the brackets are codes used to identify that "x" is a lexical error, and that "p" is a proper name. Below is an example of a text before and after tokenisation:

Before:

"In the USA are there many- [x[very]] tornados, volcanic eruptions and hurricanes. The Canary Islands- [x[Islain]] there were volcanic eruption, the last- [x[finis]] volcanic eruption- [x[eruptión]] was [p[Teneguía]] in [p[the Palma]]I like films- [x[filmes]] about- [x[the]] nature, volcanic eruptions- [x[eruptións]] etc. The new film is Volcano. The film is very good for- [x[by]] the peoples but- [x[bat]] I don't like ..... film."

After: (64 tokens)

"In" "the" "USA" "are" "there" "many" "tornados" "," "volcanic" "eruptions" "and" "hurricanes" "." "The" "Canary" "Islands " "there" "were" "volcanic" "eruption" "," "the" "last " "volcanic" "eruption" "was" "Teneguía" "in" "the Palma" "I" "like" "films " "about" "nature" "," "volcanic" "eruptions " "etc" "." "The" "new" "film" "is" "Volcano" "." "The" "film" "is" "very" "good" "for " "the" "peoples" "but " "I" "don't" "like" "." "." "." "." "." "film" "."

During the tokenization process errors are detected by finding the double brackets and the letter code; an “x” for error tagging. The letter code is found between the first and second opening bracket as follows: [x[YYYY]]. The previous word is tagged with a dash symbol “-” before the word as follows: -XXXX. So when an error appears we have the following format: -XXXX [x[YYYY]]. XXXX is the corrected form of the word and YYYY is the original form of the word.

Using the same format as with errors we code personal nouns that need to be erased in order to keep anonymity of the students. We tag such nouns as follows: [p[YYYY]].

That word is substituted with a different personal noun, in our case "Lennon". This helps to keep the meaning and structure of the text to allow automatic processing afterwards. Errors are stored within the data structure and both original and corrected forms are kept.

The data structure containing the original data is modified at this point and the clean text and the corrections that have been detected are introduced. An example of how the data structure of the previous texts looks like is shown in Table 2.

Grade	'Baccalaureate '
Province	[0×1 string]
Student id	1
Header	"[id[province: LPA; student: 1]]"
Body	"In the USA are there many- [x[very]] …
Corrections	[9×2 string]
Clean text	[1×1 tokenizedDocument]

Table 2: Example of data structure used to store processed text

5.3.2. Sentence detection

Once the text has been segmented into its components, each of the sentences that make up each document are identified. This is a necessary process in order to be able to effectively disambiguate the elements of the discourse later on. Punctuation marks are used to automatically detect the sentences.

5.3.3. Elimination of punctuation marks

The next step is to remove punctuation marks [, . : ;] present in the text so that they are not considered in the subsequent analysis. It is important that this step is carried out after sentence detection as punctuation symbols provide valuable information in sentence detection.

5.3.4. Disambiguation of discourse elements

In order to disambiguate each of the elements of the text, we make use of the dictionaries and lexicons of the English language provided by *MATLAB* to identify to which class each word belongs. The classification of each word will allow us to carry out a richer statistical study of how errors are presented in the corpus studied in this work.

5.3.5. Obtaining the Morphological Theme (Stem) of each word

In order to obtain the Morphological Theme, we have made use of Porter's algorithm (1980). This algorithm is well established in the scientific communities of natural language analysis and therefore we have considered that the use of this functionality in our context can enrich the study of the corpus and subsequent works using this tool.

5.3.6. Lemmatisation of each word

As a last pre-processing step of the texts to be analysed, all words of each text have been reduced to the form in which they appear in the dictionary (Lemma). Through the statistical study of Lemmas and Themes we can extract error patterns in a more effective way. By using lemmas in the study of errors, it can be better managed if learners make errors in specific verbs that without the root word would be difficult to group.

5.4. Text processing tools

To facilitate the study of the errors made by the students, we have created a series of functionalities that aim to facilitate the researcher's task. We can see two types of tools.

The first type of tool is more common in the field and facilitates the work by efficiently handling the data and presenting it to the researcher. The second type of tool is more elaborate as it uses mathematical methods to perform a statistical/metric analysis of the words in the texts (Barrios et al., 2015).

For both types of analysis, we have resorted to the structure known as *bag-of-words* (Harris, 1954), used in natural language processing applications, which facilitates the required computations and access to information. In turn, this structure can be used to generate descriptors of parts of the text and thus allows a comparison using similarity metrics between them. We will now detail the data management and visualisation functionalities that have been developed to create an efficient use of the studied data.

5.4.1. Word frequency analysis

This functionality focuses on showing the occurrence of each word in the corpus. In Table 3. we show an example of the word count functionality after performing lemmatisation and punctuation removal:

Word	Primary	Secondary	Baccalaureate	Total
"the"	331	997	2763	4091
"and"	1280	591	1101	2972
"is"	1653	838	479	2972
"I"	1313	786	630	2729
"my"	1745	256	225	2226
"a"	407	434	964	1805
"in"	435	332	928	1695
"Lennon"	1181	271	11	1463
"of"	61	171	1135	1367

Table 3: Word frequency after lemmatisation and punctuation removal

5.4.2. Display of words in context

This functionality shows each occurrence of a word with the discourse elements that precede and follow it. This functionality makes it easy to help the researcher to appreciate the contextual details surrounding that word in each situation. In Table 4 we show an example for the high school level of the word *volcanic*:

Context	Document	Word
"usa many tornado **volcanic** eruption hurricane canar"	1	4
"hurricane canary islands **volcanic** eruption last volcanic e"	1	9
"s volcanic eruption last **volcanic** eruption teneguia the pa"	1	12
"like films about nature **volcanic** eruptions etc new film v"	1	20
"ome due island lanzarote **volcanic** eruption tragedy immense"	4	23
"on natural calamity like **volcanic** eruption bad city villag"	5	5
"help canary island many **volcanic** century has been many eru"	5	24
"ry has been many eruption **volcanic** eruption provoke infrast"	5	30

Table 4: Display of words in context for the occurrence of *volcanic*

5.4.3. Visualisation using word clouds

A word cloud consists of a two-dimensional presentation of words that are larger or smaller depending on their frequency of occurrence in the texts. The higher the frequency, the larger the size, the lower the frequency, the smaller the size. We show an example in Figure 1 taken from the corpus under study in this paper after removing the "stop words"[1].

[1]Words like "a", "and", "to", and "the" (known as stop words) can add noise to data.

Figure 1: Word clouds for each part of the corpus with removal of "stop words", such as "a" or "the".

5.4.4. Example of Lemmatisation, Morphological Theme and Disambiguation of discourse elements

Now that Word Clouds have been defined and explained we can make use of them to show further examples of all the functionality that has been introduced. In this subsection we will further showcase our work's functionalities through examples. We will show examples of the effects lemmatization and stemming have. We showcase such examples by using Word Clouds in Figure 2.

Figure 2: Word Clouds of each section when performing Lemmatization and Stemming. In this examples of Word Clouds, "stop words" have been included.

5.4.5. Examples of Disambiguation of parts of speech

We produce an example that we hope clarifies and gives further insight on how the identification of the parts of speech functionality works. In this example we showcase how we have added the types "corrected form" and "personal data" to identify the special cases that we outlined when we either have an error corrected by the teacher or personal data needs to be anonymized. We showcase 3 examples, one for a Primary text (Table 5), one for a Secondary text (Table 6) and another for a Baccalaureate text (Table 7).

Token	Document	Sentence	Line	Type	Language	PartOfSpeech
"My"	1	1	1	Letters	Eng	Pronoun
"name"	1	1	1	Letters	Eng	Noun
"is"	1	1	1	Letters	Eng	Auxiliary-verb
"Lennon"	1	1	1	Personal Data	Eng	Proper-noun
"I'm"	1	2	1	Corrected form	Eng	Proper-noun
"eleven"	1	2	1	Letters	Eng	Numeral
"years"	1	2	1	Letters	Eng	Noun
"I'm"	1	3	1	Corrected form	Eng	Noun

Table 5: Example of Part-Of-Speech automatic detection in a Primary text

Token	Document	Sentence	Line	Type	Language	PartOfSpeech
"Dear"	1	1	1	Letters	Eng	Adjective
"teacher"	1	1	1	Letters	Eng	Noun
"I"	1	2	1	Letters	Eng	Pronoun
"am"	1	2	1	Letters	Eng	Auxiliary-verb
"in"	1	2	1	Letters	Eng	Preposition
"Borley"	1	2	1	Letters	Eng	Proper-noun
"My"	1	3	1	Letters	Eng	Pronoun
"family"	1	3	1	Letters	Eng	Noun

Table 6: Example of Part-Of-Speech automatic detection in a Secondary text

Token	Document	Sentence	Line	Type	Language	PartOfSpeech
"In"	1	1	1	Letters	Eng	Preposition
"the"	1	1	1	Letters	Eng	Determiner
"USA"	1	1	1	Letters	Eng	Proper-noun
"are"	1	1	1	Letters	Eng	Auxiliary-verb
"there"	1	1	1	Letters	Eng	Adverb
"many"	1	1	1	Corrected form	Eng	Adjective
"tornados"	1	1	1	Letters	Eng	Noun
"volcanic"	1	1	1	Letters	Eng	Adjective

Table 7: Example of Part-Of-Speech automatic detection in a Baccalaureate text

5.5. Mathematical tools for text analysis

We will now look at some examples of text processing by means of elaborated mathematical-statistical analysis. These tools have been used in the past for text analysis from a mainly computational perspective because they have been developed in a different area of knowledge. Nevertheless, the introduction of these techniques, which have been scientifically proven to be effective, to a different area such as the one we are dealing with offers promising avenues for future development.

5.5.1. Distance between words

In order to measure the *degree of error* in the errors made by learners, we propose the use of several distances between words. The first is the Levenshtein distance (Levenshtein, 1966); this distance measures the minimum number of insertions, deletions and substitutions required between two words to be the same. The second proposed distance is the Damerau-Levenshtein distance (Damerau, 1964); this distance

measures the minimum number of insertions, deletions, substitutions and character position changes required between two words. Finally, we propose the use of the Hamming distance (Hamming, 1950) which simply measures the minimum number of substitutions required.

All the previous distances have been used extensively in different fields of computer science for similarity analysis and we consider that introducing the use of these formal metrics can enrich the field of humanities research. For error analysis we advise to use the Levenshtein and Damerau-Levenshtein distances preferably because the Hamming distance requires that the correct word and the error contain the same number of characters. To show this situation and to illustrate in general the behaviour of these distances we show some examples in Table 8.

Correct Word	Student's Word	D(Levenshtein)	D(Damerau)	D(Hamming)
eruptions	erutions	1	1	Infinite
eruptions	erruptions	1	1	Infinite
eruptions	eruptoins	2	1	2
brother	borther	2	1	2
brother	borter	3	2	Infinite
brother	border	3	3	Infinite
family	family	0	0	0
family	famili	1	1	1
family	feimily	2	2	Infinite

Table 8: Examples of the value of each distance on grammatical errors

5.5.2. Term Frequency - Inverse Document Frequency (TF-IDF)

The TF-IDF value represents how important a particular term is to the text to which it belongs. This technique has been shown to provide good results in data mining (Rajamaran and Ullman, 2011) and therefore we believe that its exploration as a metric of the impact of the word in which an error has occurred can be valuable. The use of such a statistical metric can be extended to a more global concept: how we understand the errors a learner makes. Errors are no longer all the same, and they are no longer always equally impactful, even if they are the same. Their impact depends on the nature of the text and the importance of the word in which the error is made for understanding the text.

The TF-IDF value is the product of two statistical metrics: the term frequency and the inverse document frequency. The inverse document frequency tells us how much information a word contributes to a text. In turn, this metric can be modified using several variants of the original method, such as the one presented by Barrios et al. (2015). It is also possible to define the weight that each word has in a particular way, as this will allow us to make statistical studies specific to the cases we will deal with.

As we have seen, the calculation of a TF-IDF metric is a powerful and flexible tool that can provide a more elaborate analysis of the linguistic corpus we are trying to treat. We will show an example of what can be achieved using the TF-IDF metric:

1. First, we create what is called a "Bag of Words" using all the texts in the corpus. A "Bag of Words" simply creates a structure that quantifies the number of times each term appears in a document.

2. In our Corpus we have 1153 documents (texts from primary, secondary, and high school students), for which we have a total of 5697 terms once the texts have been cleaned as explained above.

3. To this data structure we calculate the TF-IDF value. This calculation will determine which words make each text more unique from a statistical point of view. Once the TF-IDF value has been calculated, we have a measure of how important each

word is for each document. That is, the higher the TF-IDF value, the more important that word is for that document in the corpus.

4. To show this phenomenon we show the result of the calculation of the TF-IDF value for documents 1151 and 1152 in Figure 3. For each word that is part of document 1151 and 1152 we have calculated a TF-IDF value that represents how unique that word is with respect to the rest of the corpus. All values have been represented by a histogram. We can observe for example that document 1152 has more words with a high TF-IDF value than document 1151, which indicates that some documents deviate more from the norm than others, etc.

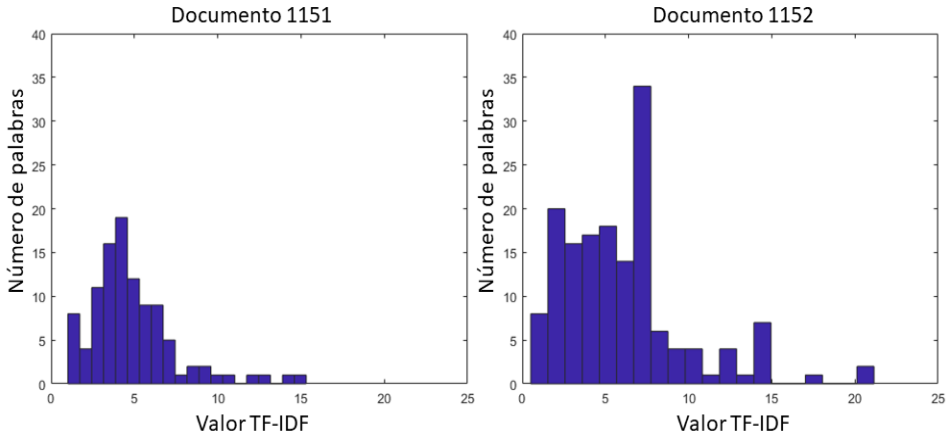

Figure 3. Histograms of the TF-IDF value of each word for documents 1151 and 1152.

In short, these elaborated statistical analyses open up numerous options for analysing a corpus and its documents, going to statistical parameters that allow a comparative analysis unattainable to a human eye. To further clarify the value of the TF-IDF metric we will show specific examples for both documents 1151 and 1152.

In both documents there are 26 shared words of which we show the first 6 shared words in Table 9. It can be seen that for example the word "the" is statistically very important for document 1152 as it has a very high value (TF-IDF of 13.9254) while it is not as statistically important for document 1151 (TF-IDF of 3.3421). While on the other hand the words "very" and "I" are more statistically relevant for document 1151 and quite irrelevant (TF-IDF values of 1.0729 and 0.5082) for document 1152. By using this metric of statistical relevance for each word within a document, elaborate measures can be obtained to measure how unique/rich the writing of each student is.

WORD	*in*	*and*	*very*	*I*	*the*	*my*
Document 1151 (TF-IDF value)	1.0021	2.8320	4.2916	4.5740	3.3421	3.0774
Document 1152 (TF-IDF value)	1.0021	1.2587	1.0729	0.5082	13.9254	1.0258

Table 9: TF-IDF value of first 6 shared words between documents 1151 and 1152

Conclusions

As the main conclusion and to summarise the results and the work presented in this article, we would like to emphasise the numerous possibilities for the future that are opened up by the use of tools such as the one presented here. Sharing the work of this project for use by the scientific community makes it possible to create incremental tools that avoid the inescapable obsolescence that classical software developments produce.

Apart from the benefits of free use of the code of such a tool, from the point of view of preventing tools from becoming obsolete and the work carried out from being lost, this opens up a new avenue for incremental growth and greater sophistication in the analysis of texts through the functionality provided by more advanced mathematics. Examples of how this can make text analysis more sophisticated can be seen in the use of word distances and complex metrics such as the TF-IDF value.

It should again be noted that the code used for this tool is freely available on Github (https://github.com/andarinneo/textProcessingHumanities) under a Creative Commons non-commercial licence (BY-NC). This licence allows use as long as the original source is cited, and it is not used for commercial purposes.

In terms of prospective lines of study, it is clear that this computer tool will enable to carry out a very detailed longitudinal analysis. Moreover, considering that it is an extensive corpus, its usefulness is even more evident.

One aspect that we consider very relevant is the possibility of error analysis taking into account the correct productions that occur in each word. For this purpose, it will be essential to have the percentage of errors, but also the TF-IDF value. Also, within the analysis of error, the measurement of the distance between the correct word and the error will make it possible to determine the type of error and to evaluate the different types of spelling errors in order to check the evolution of the pupils' acquisition process.

Finally, we would like to point out that this tool also allows to carry out other types of studies related to discourse analysis.

References

Alexopoulou, A., 2011, "La función de la interlingua en el aprendizaje de lenguas extranjeras". *Revista Nebrija de Lingüística Aplicada a la Enseñanza de Lenguas, 9,* pp. 86-101.

Barrios, F., López, F., Argerich, L. and Wachenchauzer, R., 2015, Variations of the Similarity Function of TextRank for Automated Summarization [Conference presentation]. Argentine Symposium of Artificial Inteligence (ASAI 2015).

Chomsky, N., 1956, "Three models for the description of language". *IRE Transactions on Information Theory,2,* pp. 113-124.

Conrad, S., 2002, "Corpus linguistic approaches for discourse analysis". *Annual Review of Applied Linguistics,22,* pp. 75-95.

Corder, S., 1967, "The significance of learners' errors". *International Review of Applied Linguistics, 5,* pp. 161-169.

Damerau, F., 1964, "A technique for computer detection and correction of spelling errors". *Commun ACM, 7* (3), pp. 171–176.

David, M. and Chapman, C., 2020, Unicode Text Segmentation. Version: Unicode 13.0.0. https://www.unicode.org/reports/tr29/

De Alba Quiñones, V., 2008, *Análisis de errores y de actuación en producciones escritas de aprendices alemanes de español. Estudio léxico-semántico* [Unpublished master's thesis]. University Pablo de Olavide, Seville, Spain.

Ellis, R., 2008, *The study of second language acquisition*. Oxford University Press.

Gilquin, G., De Cock, S. and Granger. S. (Eds.), 2010, *Louvain international database of spoken English interlanguage (LINDSEI)*. UCL Presses Universitaires de Louvain.

Granger, S., 1998, "The computer learner corpus: A versatile new source of data for SLA research". In S. Granger (Ed.), *Learner English on Computer*,Rouletge, pp. 3-18.

Granger, S., 2012, "How to use foreign and second language learner corpora". In Mac- key, A., and Gass, S. M. (Eds.), *Research methods in Second Language Acquisition: A practical guide* (pp. 7-29). London: Blackwell Publishing.

Granger, S., Gilquin, G. and Meunier, F., 2015, "Introduction: learner corpus research – past, present and future". In S. Granger, G. Gilquin and F. Meunier (Eds.), *The Cambridge handbook of learner corpus research,* Cambridge University Press, pp. 2-5.

Grant, L. and Ginther, A., 2000, "Using computer-tagged linguistic features to describe L2 writing differences". *Journal of Second Language Writing, 9,* pp.123-145.

Guerrero Martín, J.L., 1991, "FRECONWIN: Una explicación informática para el estudio lexicométrico de textos". *Lenguaje y Textos,21,* pp. 51-65.
Hamming, R. W., 1950, "Error detecting and error correcting codes". *The Bell System Technical Journal*, *2*, pp. 147-160.
Harris, Z., 1954, "Distributional Structure". *WORD, 10,* pp.146-162.
ICU, 2020, *ICU International Components for Unicode. Boundary Analysis. IBM (2000-2009).* http://userguide.icu-project.org/boundaryanalysis
Instituto Cervantes, 2015, *Corpus de aprendices de español como lengua extranjera.*https://www.cervantes.es/lengua_y_ensenanza/tecnologia_espanol/caes.htm
Kleene, S. C. and Shannon, C., 1951, *Representation of Events in Nerve Nets and Finite Automata. Automata Studies*. U. S. Air Force Project Rand Research Memorandum (RM-704). The RAND Corporation.
Kolodziej, N.J. and Columba, L., 2005, "Invented spelling: Guidelines for parents". *Reading improvement, 42* (4), pp. 212.223.
Kusseling, F. and Lonsdale, D., 2013, "A corpus-based assessment of French CEFR lexical content". *The Canadian Modern Language Review, 69,* pp. 436-461.
Lado, R., 1957, *Linguistics across cultures: Applied Linguistics for Language Teachers*. Ann Arbor, Michigan: University of Michigan.
Levenshtein, V., 1966, "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady, 10*, pp. 707-710.
Madan, C., 2013, Introduction to Matlab for behavioral researchers. *ISBN:* 9781452255408. SAGE publications, Inc.
McEnery, T. and Hardie, A., 2011, *Corpus Linguistics: Method, Theory and Practice*. Cambridge University Press.
McEnery, T., Brezina, V., Gablasova, D. and Banerjee, J., 2019, "Corpus linguistics, Learner corpora, and SLA: employing technology to analyze language use". *Annual Review of Applied Linguistics, 39,* pp. 74-92.
Navarro Rodríguez, A., 2008, *Análisis de los errores léxicos en la producción escrita de alemán L3 por alumnos de Escuela Oficial de Idiomas.* [Unpublished master's thesis]. Universidad de Las Palmas de Gran Canaria.
Oxford, R. (2016). *Teaching and researching language learning strategies. Self-regulation in Context.* London: Routledge.
Oxford, R. y Amerstorfer, C.M. (2018). *Language learning strategies and individual learner characteristics.* London: Bloomsbury.
Porter, M.F., 1980, "An algorithm for suffix stripping". *Program, 44* (3), pp. 130−137.
Rajaraman, A. and Ullman, J., 2011, *Mining of Massive Datasets*. Cambridge University Press.
Sánchez Rufat, A., 2015, "Análisis contrastivo de interlingua y corpus de aprendientes: precisiones metodológicas". *Pragmalingúística, 23,* pp. 191-210.
Selinker, L., 1972, "Interlanguage". *International Review of Applied Linguistics, 10,* pp. 209-231.
Wood, M., Peñate, M. and Bazo, P., 2004, *FreconWin. Corpus canario de inglés escrito*. Consejería de Educación, Cultura y Deportes del Gobierno de Canarias.

Begoña **CAMPOS ESTARÁS** has a degree in English Philology. After completing the Master's Degree in Teacher Training, she is writing her doctoral thesis on English language learning errors in the University of Las Palmas de Gran Canaria (ULPGC).

Adrian **PENATE-SANCHEZ** received his PhD from the Institut de Robòtica i Informàtica Industrial in Barcelona. In 2014, he joined the Computer Vision group at Toshiba's Cambridge Research Laboratory for a 5-month internship. From 2015 to 2018 he was a research associate at University College London working in the fields of 3D computer vision and machine learning. In 2018 he joined the Robotics Institute at Oxford University as a postdoctoral researcher. In 2020 he joined the ULPGC as a Distinguished Researcher and Lecturer under a Beatriz Galindo grant.

Received: March 14, 2023 | Revised: October 9, 2023 | Accepted: November 4, 2023 | Published: December 15, 2023