A STUDY OF SPELLING ERRORS IN WORD PROCESSING: DETECTION AND CORRECTION

by

María I. Díaz-Figueroa

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
JUNE 2006

Approved by:

_____	_____
José Borges, PhD Member, Graduate Committee	Date
_____	_____
Manuel Rodríguez, PhD Member, Graduate Committee	Date
_____	_____
Néstor Rodríguez, PhD President, Graduate Committee	Date
_____	_____
Mercedes Ferrer-Alameda, PhD Representative of Graduate Studies	Date
_____	_____
Isidoro Couvertier, PhD Chairperson of the Department	Date

ABSTRACT

A STUDY OF SPELLING ERRORS IN WORD PROCESSING: DETECTION AND CORRECTION

By

María I. Díaz-Figueroa

June 2006

Chair: Néstor Rodríguez, PhD
Major Department: Electrical and Computer Engineering

This document presents a research study that identifies spelling errors committed by people writing in Spanish and how the errors are corrected. A usability test was conducted in which 20 people were asked to write a document in Spanish using Microsoft Word. During the writing process, the interaction of the participants with Microsoft Word was recorded. The recordings were then analyzed to identify the errors committed by the users and how these were corrected. Errors were classified in eleven different types. The most important finding was that a large proportion of the errors were related with words that include á, é, í, ó, ú, or ñ characters. The study revealed that three fourth of the errors are corrected using one out of four possible techniques. The large majority of the errors were identified and corrected on the spot using the backspace key. Many of the uncorrected errors were errors that were not detected by the word processor. However, with simple algorithms these errors can be detected and corrected. We developed algorithms to detect four of these types of errors. The algorithms were able to detect and provide correct alternatives for all the errors in words that currently exist on the dictionary used to implement the algorithms.

RESUMEN

UN ESTUDIO DE ERRORES ORTOGRÁFICOS EN PROCESADORES DE TEXTO: DETECCIÓN Y CORRECCIÓN

Por

María I. Díaz-Figueroa

Junio 2006

Consejero: Néstor Rodríguez
Departamento: Ingeniería Eléctrica y Computadoras

En este documento se presenta un estudio que identifica errores ortográficos cometidos por personas escribiendo en español y cómo los errores fueron corregidos. Se condujo un estudio de usabilidad en el que se pidió a 20 personas que escribieran un documento en español utilizando Microsoft Word. Durante el proceso de escritura la interacción de los participantes con Microsoft Word fue grabada. Las grabaciones fueron analizadas para determinar los tipos de errores cometidos y cómo fueron corregidos. Once tipos de errores fueron identificados en el estudio. El hallazgo más importante fue que un gran número de errores estaban relacionados con palabras que incluyen las letras á, é, í, ó, ú, o ñ. El estudio reveló que tres cuartas partes de los errores fueron corregidos y que se utilizaron cuatro técnicas para corregirlos. La mayoría de los errores fueron corregidos en el momento en que se cometieron utilizando la tecla "backspace". Muchos de los errores que no fueron corregidos fueron errores que no fueron detectados por el procesador de palabras. Sin embargo, con algoritmos simples estos errores pueden ser detectados y corregidos. Nosotros desarrollamos algoritmos para detectar cuatro tipos de estos errores. Los algoritmos pudieron detectar y proveer alternativas correctas para todos los errores de palabras que existían en el diccionario utilizado para implementar los algoritmos.

This thesis has been dedicated to God, family (especially to my parents: José Díaz and Carmen Figueroa) and friends for all their support, love and understanding.

ACKNOWLEDGEMENTS

During the development of my graduate studies at the University of Puerto Rico at Mayagüez, several persons and institutions collaborated directly and indirectly with my research. Without their support it would be impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

First of all, I want to thank the creator of the universe for giving me the opportunity of walking another huge step in my life: my graduate studies. Thanks God for all your blessings and for being all the time with me. Also, I want to say thank you to my parents and family for all their support, help and understanding during the last three years. I also want to say thank you to a very important person in my life, my boyfriend, Navendu Jain. Thank you for all your support, help and love. If I finished this today, it was because of you. Thanks for being my inspiration. I cannot forget my advisor, Dr. Néstor Rodríguez because he gave me the opportunity to research under his guidance and supervision. I received motivation; encouragement and support from him during all my studies. Néstor, thanks for your example, support, motivation, inspiration and your faith on me. I also want to thank Dr. José Borges and Dr. Manuel Rodríguez, my graduate committee, for all their help and support. Also I want to thank my friends: Luz Acabá-Cuevas, Marlyn Claudio-Vázquez, Edgard Vélez-Morales, Christian Nieves-Santiago, and Omar Valenzuela-Agosto. You guys are awesome! Thanks for all your support and help during the last years. You guys have a special corner in my heart. I hope that we can be always friends as we are and we can keep in touch with each other for all the oncoming years. Also, I want to say thanks to Iglesia Gethsemaní, AIC in

Vega Baja and its pastor Rev. Juan González-Alicea for all their support and help during the last years. Pastor: you were of great motivation during all this time. Thank you for all your advices and support and for being, more than a pastor, a good friend to me. Thanks to the people that take some time from their busy time to help me with the usability test. I will always be thankful to all of you. THANK YOU.

The GEM Fellowship sponsored by IBM, NSF Grant, and the PRIDCO Grant provided the funding and the resources for the development of this research. Thanks a lot to Dr. Domingo Rodríguez for the financial support and for the good lessons you taught me during the last years. I learned a lot from you.

Table of Contents

Table List

Tables **Page**

Figure List

Figures **Page**

Chapter 1

INTRODUCTION

1.1 Motivation

Over the years, the spell checker functionality of word processors has gained importance in many natural languages. Any person that has to write a text document uses this software package to sanitize them. The use of spell checking software is widely prevalent in many fields ranging from academia such as schools and universities, government departments such as law and finance, business enterprises such as banks and mortgage companies, to social institutions such as churches and hospitals. Further, in today's information age, the information retrieval task primarily depends on indexing the keywords in the documents available on the web. Therefore, for a search engine to return effective results to a user's query, the keywords must be spelled correctly in the documents.

Realizing the widespread use of this spell check functionality in everyday life, researchers have been working to improve the quality of spell checker software in order to offer end-users the ability to create documents without grammatical or contextual errors. Existing spell checker software, however, still does not capture all errors committed by the users as shown by two recent studies. In the first study, Huang and Powers [Huang02] identified six types of common errors that users commit using word processor with spell checking. These errors are classified as typographical, homophone, grammatical, frequency disparity, learners, and idiosyncratic. Typographical errors typically manifest when a user types a letter that is adjacent in the keyboard to the correct letter the user wants to type.

Homophone errors are words that sound similar but they have a different meaning (i.e. piece and peace). An example of a grammar error is “among” and “between” words. As both words have a very similar meaning, many people tend to commit errors when writing them because they does not know when to uses them. The frequency disparity errors are when a user is typing the abbreviation of words but, the user can be confuse and type a similar but unintended word. For example, consider the following sentence: “They are here”. That sentence can be abbreviated as: They’re here but the user can be confused and then type Their here. The learner errors are those committed by users that are learning and writing in a language that is not their first language. The idiosyncratic error corresponds to those that have been committed for an unknown reason [Powers97].

In the second study [Galleta05], the authors showed that spell checker software detects some errors wrongly and still cannot detect all the existing errors. In other words, the study revealed three types of things that can result from using spell checkers: correctly identified errors, False Positives errors and False Negatives errors. When the spell checkers detect a real error, this error is a “correctly identified error”. “False Positives errors” are those where spell checkers indicate an error but it is not a real error. “False Negatives errors” are errors that are present in a text but the spell checkers do not detect them. What is different about this research work compared to Huang’s research [Huang02] is that they focused on software performance not in misspell words. The [Galleta05] research revealed that when the spell checkers correctly identified errors, it helps low verbal people (people with less experience in a given natural language) to write almost as a high verbal person. On the contrary, high verbal users that rely on the spell checkers end up committing more errors, than when they

don’t use spell checkers. This is due to the false negative and positive errors caused by the spell checkers.

Our proposed research is motivated by this need for identifying and classifying new kinds of errors that are not captured by existing spell checkers. We want to conduct a usability study to detect errors that users commit during their interaction with word processor software. We want to focus on those errors that are not commonly detected and corrected by spell checkers. Specifically, we aim to study a special class of spelling errors in Spanish that are not detected by Microsoft Word, a popular spell checker software.

1.2 Objectives

The goal of this work was to identify and classify the types of spelling errors committed and how were corrected by writers while writing in Spanish using a commercial word processor. One of the objectives of the study was to identify errors that are not detected by the spell checkers, errors that are caused by the spell checkers and the strategies exercised by the users to correct errors. Another objective was to develop algorithms to detect and correct errors that are not typically detected by spell checkers.

1.3 Contributions

The main contributions of this work were: (1) the generation of a profile of spelling errors committed by writers while writing in Spanish using a commercial word processor; (2) the identification of the spelling error correction strategies used by writers; (3) the development of algorithms to detect Accent, Special Accent, Ñ and Transposition & Disorder

spelling errors. The most important finding of the study was that almost half of the errors were related with words that include á, é, í, ó, ú, or ñ characters. Another important finding was that the large majority of the spelling errors were corrected on the spot using the backspace key. The algorithms developed to detect and correct Accent, Special Accent, Ñ and Transposition & Disorder spelling errors were able to detect and provide correct alternatives for all the errors of words that existed on the dictionary used to implement them.

1.4 Thesis Outline

The structure of this thesis will be as follows: Chapter 2 presents the literature review concerning to previous research work on the study of spell checkers. Chapter 3 describes the results of a study of spelling errors in word processing in Spanish Natural Language. Chapter 4 describes some algorithms developed to detect and correct spelling errors. Chapter 5 presents a discussion of the results obtained in this research work and compares them with other results obtained in other research works. Chapter 6 presents the conclusions of the study and the suggestions for future work.

Chapter 2

THEORETICAL BACKGROUND

2.1 Literature Review

There have been various studies dealing with error recognition in word processing and for the improvement of spell checkers. Huang and Powers [Huang02] came up with a solution for automatically learning contextual knowledge for spelling and grammar correction. They aim particularly to deal with cases where the words are in the dictionary and it is not obvious that there is an error. Those errors are specifically related with the context of the word. They focus their research work in solving the context-sensitive spelling correction issues where the main problem consider is the resolution of lexical ambiguity, syntactic and semantic, based on the surrounding context. They reduced the searching time by defining two keys to define a word instead of using the dictionary to look for the meaning of the words.

In other research work, Fallman [Fallman02] used the World Wide Web as a database to correct grammar and spell checks errors. This application is implemented as a client/server system. The client sends a string or a phrase to the server and the server make a search using a search engine on the web. The system counts the occurrence of that word or phrase in the web and let the user know the number of hits of the incidence of that word or phrase. This is a good solution but the user needs to be connected to Internet all the time in order to use it.

Durham, Lamb and Saxe [Durham83] wanted to learn how useful a spell check could be in a user interface. They conducted their study observing a user interacting with software called RdMail that is an electronic mail system. In this software, they replace the keywords lookup routine with a spelling corrector. The corrector builds a vector of string pointers and forces the user to select one of the words provided by the corrector. After running an experiment during 41 days, they concluded that their mechanism solve 27% of the errors made by the user and let an open question of how to solve the other 73%. In addition, they concluded that spell checkers are straightforward for human computer interfaces.

Bolshavok [Bolshakov03] and Gelbukh proposed a solution for *malapropism* – writing words with similar sound but different meaning to what the user intended. For example, "the boy is eating a *peace* of pizza". The real word that applies to this sentence is piece. They used collocations and a search engine to correct this kind of problem. Collocations are phrases composed of words that co–occur for lexical rather than for semantic reasons. They use a collocation database. If a specific combination of words does not exist in the collocation database, they use a search engine to search that combination and they state that if a combination of words occur several times in the web, it is correct. The problem with this is that not all the English web pages are developed by people who know the language and there could be errors as well.

Hodge [Hodge01] and Austin developed an algorithm to solve the four main problems of spelling: insertion, deletion, substitution, and transposition (double substitution) using the AURAL modular neural system. AURAL uses a supervised learning rule and do

not require an additional memory allocation because its architecture uses correlation matrix memory (CMMs). They implement an Information Retrieval (IR) system and an algorithm to calculate the distance of the words using the Hamming Distance to find out the errors in the insertion and deletion process and the n-gram technique to find the errors in the substitution and transposition process. They take the input word and convert it into binary. The system compares it with the lexicon in order to find a word that matches the input of the user. If the system finds an exact match for that word, then the word is correct but if it finds something that varies on a word, it is identified as an error. The system then use hamming distance and a shifting n-gram to generate two sets of potential matches. After that, the system makes a union of the two sets and give to the user the alternatives to fix the misspell word. They compared their algorithm with other general algorithms that solve this problem to see its performance in terms of speed retrieval and use of memory. In the results, the running time of this methodology was of *O(1)* for an exact match and in terms of memory, the methodology used low memory compared to other technique called *n*-gram.

Chapter 3

A STUDY OF SPELLING ERRORS IN WORD PROCESSING IN SPANISH

This chapter presents a study conducted to identify the errors that user commits while they are typing in Spanish Natural Language. The methodology used for the study is presented in section 3.1. The results of the study are presented in section 3.2. The techniques used by user to fix the errors committed are showed in Section 3.3.

3.1 Methodology

The goal of this work was to study the interaction of typical users of word processors with respect to spelling errors and correction. As a first step in this research work, a study was conducted in which twenty people were asked to write for an hour using MS Word. The participants were college students and recent college graduates. They were asked to write in Spanish about something related with their lives. They were asked to type as they normally do. All the participants used Microsoft ® Office Word 2003.

The participants' interaction with the computer was recorded using the TechSmith Morae software. This software records and synchronizes user and system data for usability analysis. The software consists of three components: Morae Recorder, Morae Remote Viewer, and Morae Manager. Morae Recorder is the component of the software that captures the interaction of the user while he/she is using the computer. This part of the software was installed in the users' computers. This component can be configured to capture important

things from the screen, keyboard, and the mouse to be used in the analysis of the interaction. The Morae Remote Viewer allows experimenters to watch the interaction of a user remotely through Internet. For this study, this component was not used because it was not necessary to monitor the participants while interacting since the recording of the interaction provided the necessary data for the study. Finally, Morae Manager was used to analyze the recorded interaction of every participant. Morae Manager allows the researcher to place markers on the recording, so he/she can easily move to that point of the recording while reviewing it. The software allows the researcher to do search in the entire recording per marker.

The Morae Recorder software was configured to record the keystrokes (input from the keyboard), screen text, and mouse clicks (highlight mouse cursor, left and right mouse clicks). It was set to record the user activity for a period of one hour. The recording was done once for each participant.

The collected data was analyzed to identify the type of errors committed by users while typing. The behavior of the spell checker was also studied to identify words automatically corrected. In addition the strategies used by the participants to correct misspell words was also studied. We ran a pilot test to identify the types of errors that can be committed while users are typing.

From pilot tests eleven errors categories were identified. Some of these errors such as typographical, homophone, transposition, extra letter, wrong letter and missing letter have been previously identified in the literature for the English language [Power97, Durham83]. The description of these errors follows.

- **Extra Letter** - The user types an extra letter in a word (i.e. “estuudio” instead of estudio).
- **Missing Letter** – The user does not type a letter in a word (i.e. “esudio” instead of “estudio”).
- **Homophone** - Words that sound similar but they have a different meaning (i.e. “ciervo” and “siervo”).
- **Typographical** – The user types an adjacent letter in the keyboard instead of the correct one (i.e. “sin” instead of “son”).
- **Transposition & Disorder** – Disorder corresponds to the case where the user types a word with all it letters but in an incorrect order (i.e. Aoccdrnig instead of According), while transposition is a special case of disorder in which two adjacent letters in a word are exchanged (i.e. “etsudio” instead of “estudio”).
- **Wrong Letter** – The user types a wrong letter in a word (i.e. “estgdio” instead of “estudio”).

Through pilot studies four additional error categories were identify. Three of these common categories (Accent, Ñ Error and Special Accent) are inexistent in the English Language but very common in the Spanish language. The description of these four additional errors follows:

- **Caps Lock** – The user is typing the first letter of a word in a sentence and turns on the Caps Lock and continues typing in capital letters.
- **Ñ** – The word has a ñ or Ñ but the user does not type it and usually type a n instead.

- **Accent** – When a user types a word that must have a vowel with an accent and writes the vowel without the accent.
- **Special Accent** – The writer does not place an accent on a word that should have an accent or the writer places an accent on a word that should not have an accent (i.e. cambie, cambié). In both cases, both words are correct words of the dictionary but only one of them is correct in the context of a sentence.

3.2 Results

A summary of the number of errors identified for each participant by category is presented in Table 3.1. In order to make a fair analysis of the results the number of errors committed by each of the participants for each category were normalized by dividing the number of errors by the total number of words written (see Table 3.2).

Table 3.1 Errors Committed by the Participants

USERS	Transposition & Disorder	Extra Letter	Wrong Letter	Missing Letter	Typographical	Homophones	Grammatical	Caps Lock	Accent	Special Accent	Ñ	Total Errors	Total Words
User 1	6	17	9	10	15	3	1	0	35	3	1	100	521
User 2	19	22	46	55	36	22	0	1	194	53	0	448	2478
User 3	11	30	33	30	13	15	0	1	48	36	13	230	873
User 4	4	29	17	15	35	0	0	1	46	19	12	178	1029
User 5	2	23	13	13	12	3	0	2	45	7	4	124	715
User 6	1	19	12	9	10	3	0	5	45	13	7	124	690
User 7	15	14	11	27	12	0	0	1	67	29	11	187	1372
User 8	13	32	25	34	15	7	0	0	160	34	21	341	2155
User 9	1	13	6	9	4	2	0	2	39	18	6	100	850

User 10	10	18	8	18	6	1	0	0	89	19	3	172	1182
User 11	6	12	6	21	10	2	0	2	53	30	11	153	1021
User 12	13	36	53	67	66	20	2	5	92	58	8	420	1225
User 13	11	48	27	25	8	3	0	1	49	24	7	203	960
User 14	19	20	25	36	16	3	0	0	59	33	3	214	1118
User 15	15	32	46	24	17	19	0	0	170	66	18	407	2650
User 16	23	47	28	56	22	8	0	7	80	91	15	381	2059
User 17	17	25	21	32	16	2	0	0	56	14	1	190	803
User 18	9	9	15	11	12	2	0	3	28	13	0	102	927
User 19	1	14	20	25	13	0	0	0	2	2	1	78	1350
User 20	34	27	20	53	11	0	0	3	8	1	0	157	1532

Table 3.2 Errors Committed by the Participants Normalized by Total Number of Words Written.

USERS	Transposition & Disorder	Extra Letter	Wrong Letter	Missing Letter	Typographical	Homophones	Grammatical	Caps Lock	Accent	Special Accent	Ñ
User 1	1.15%	3.26%	1.73%	1.92%	2.88%	0.58%	0.19%	0.00%	6.72%	0.58%	0.19%
User 2	0.77%	0.89%	1.86%	2.22%	1.45%	0.89%	0.00%	0.04%	7.83%	2.14%	0.00%
User 3	1.26%	3.44%	3.78%	3.44%	1.49%	1.72%	0.00%	0.11%	5.50%	4.12%	1.49%
User 4	0.39%	2.82%	1.65%	1.46%	3.40%	0.00%	0.00%	0.10%	4.47%	1.85%	1.17%
User 5	0.28%	3.22%	1.82%	1.82%	1.68%	0.42%	0.00%	0.28%	6.29%	0.98%	0.56%
User 6	0.14%	2.75%	1.74%	1.30%	1.45%	0.43%	0.00%	0.72%	6.52%	1.88%	1.01%
User 7	1.09%	1.02%	0.80%	1.97%	0.87%	0.00%	0.00%	0.07%	4.88%	2.11%	0.80%
User 8	0.60%	1.48%	1.16%	1.58%	0.70%	0.32%	0.00%	0.00%	7.42%	1.58%	0.97%
User 9	0.12%	1.53%	0.71%	1.06%	0.47%	0.24%	0.00%	0.24%	4.59%	2.12%	0.71%
User 10	0.85%	1.52%	0.68%	1.52%	0.51%	0.08%	0.00%	0.00%	7.53%	1.61%	0.25%
User 11	0.59%	1.18%	0.59%	2.06%	0.98%	0.20%	0.00%	0.20%	5.19%	2.94%	1.08%
User 12	1.06%	2.94%	4.33%	5.47%	5.39%	1.63%	0.16%	0.41%	7.51%	4.73%	0.65%
User 13	1.15%	5.00%	2.81%	2.60%	0.83%	0.31%	0.00%	0.10%	5.10%	2.50%	0.73%
User 14	1.70%	1.79%	2.24%	3.22%	1.43%	0.27%	0.00%	0.00%	5.28%	2.95%	0.27%
User 15	0.57%	1.21%	1.74%	0.91%	0.64%	0.72%	0.00%	0.00%	6.48%	2.49%	0.68%
User 16	1.12%	2.28%	1.36%	2.72%	1.07%	0.58%	0.00%	0.34%	3.89%	4.42%	0.73%
User 17	2.12%	3.11%	2.62%	3.99%	1.99%	1.00%	0.00%	0.00%	6.97%	1.74%	0.12%
User 18	0.97%	0.97%	1.62%	1.19%	1.29%	0.11%	0.00%	0.32%	3.02%	1.40%	0.00%
User 19	0.07%	1.04%	1.48%	1.85%	0.96%	0.00%	0.00%	0.00%	0.15%	0.15%	0.07%
User 20	2.22%	1.76%	1.31%	3.46%	0.72%	0.00%	0.00%	0.20%	0.52%	0.07%	0.00%
Mean	0.91%	2.16%	1.80%	2.29%	1.51%	0.48%	0.02%	0.16%	5.29%	2.12%	0.57%
SD	0.61%	1.11%	0.98%	1.15%	1.18%	0.50%	0.05%	0.19%	2.14%	1.28%	0.44%
Median	0.91%	1.78%	1.69%	1.94%	1.18%	0.32%	0.00%	0.10%	5.39%	2.00%	0.67%
Minimum	0.07%	0.89%	0.59%	0.91%	0.47%	0.00%	0.00%	0.00%	0.15%	0.07%	0.00%
Maximum	2.22%	5.00%	4.33%	5.47%	5.39%	1.72%	0.19%	0.72%	7.83%	4.73%	1.49%

Figure 3.1 presents the percentages of all errors that each category constitutes. These percentages were calculated using the averages of the normalized number of errors committed by the participants for each category. The results revealed that the Accent error category is the one with the highest occurrence with over 30%, while the grammatical error category is the one with the lowest occurrence. The errors that are unique when writing in Spanish (Accents, Special Accents and Ñ) constitutes over 46% of the errors committed by the participants.

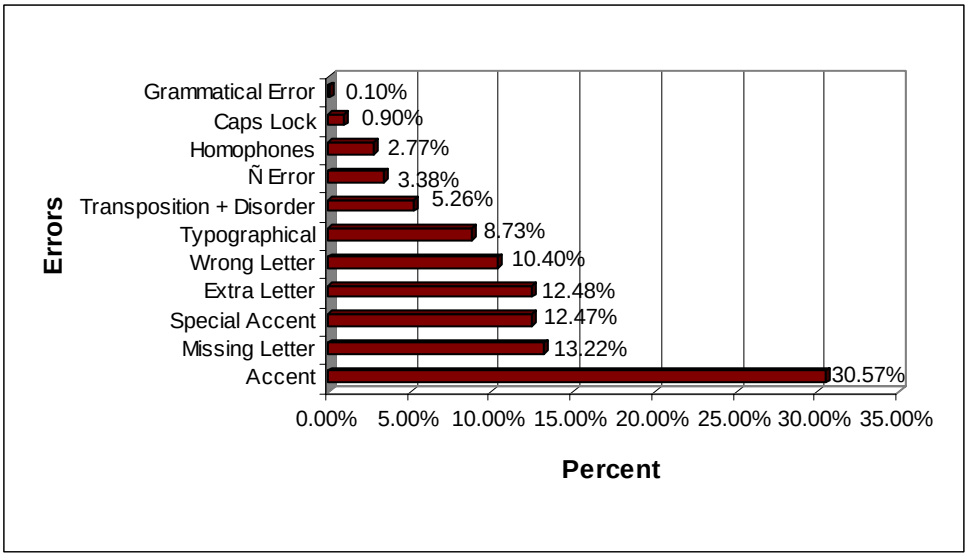


Figure 3.1 Distribution of Errors Committed by the Participants by Category

Figure 3.2 shows the variability in the normalized number of errors committed by the participants. The results indicate that of all the words written by the participants an average of 17.31% were erroneous with a standard deviation of 6.16%. The maximum percentage of errors was 34.29% and the minimum 5.78%.

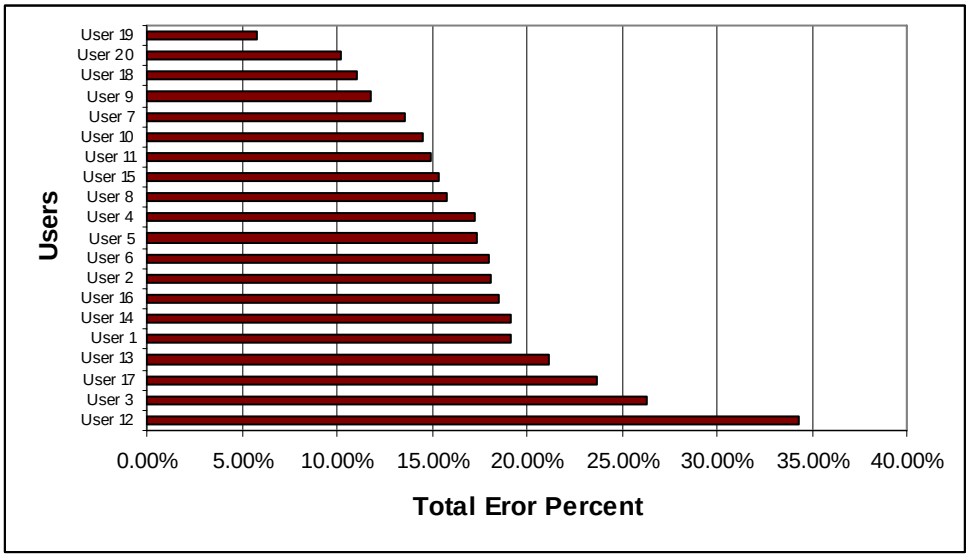


Figure 3.2 Percentage of All Words that Were Errors Committed by Each Participant

Table 3.2 shows the mean, standard deviation, maximum and minimum values for each error category. As shown in the figures of Appendix A the errors committed by the participants in each category where are essentially normally distributed. However, there were participants that did not commit Homophones, Grammatical Caps Locks and Ñ errors.

3.3 Error Correction

An important action observed during the study was how the errors committed by the participants were corrected. The results indicate that an average 73.00% of all the errors were corrected with a Standard Deviation of 19.03%. The variability for the percentage of errors corrected by the participants can be appreciated in Figure 3.3. The user that corrected the fewer number of errors fixed 29.62% of them while the one that corrected the most fixed

100.00% of the errors. The spell checker identified many of the committed errors but some users do not review the document to fix them.

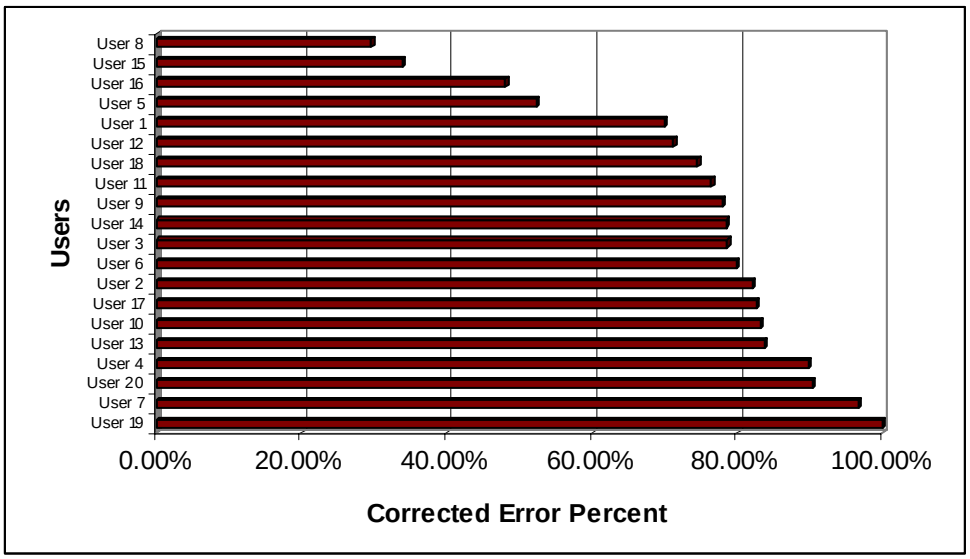


Figure 3.3 Percentage of Errors Corrected by the Participants

The users used different ways to correct the errors they committed while typing. For this study four techniques were identified: backspace, right click, spell checker and undo. The backspace technique was used to correct errors that were detected by the users immediately while typing a word in most of the cases, without the help of the spell checker The right click technique consisted in doing right click on the mouse on a word marked as incorrect by the word processor. When this is done the word processor displays a menu of words from which the user can select the correct word if available and substitute the erroneous word. The spell checker technique is when the user types all the document and go back to correct the erroneous words using the spell checking command in the Tools menu of Microsoft Word. The undo technique consists of hitting the undo button after the word

processor automatically corrects a word that was typed correctly. This technique was used only once during the study.

The percentage of errors corrected with each technique is presented on Figure 3.4. The results reveal that most of the errors were fixed with the backspace technique. Thus, most of the errors were fixed right at the moment they occurred.

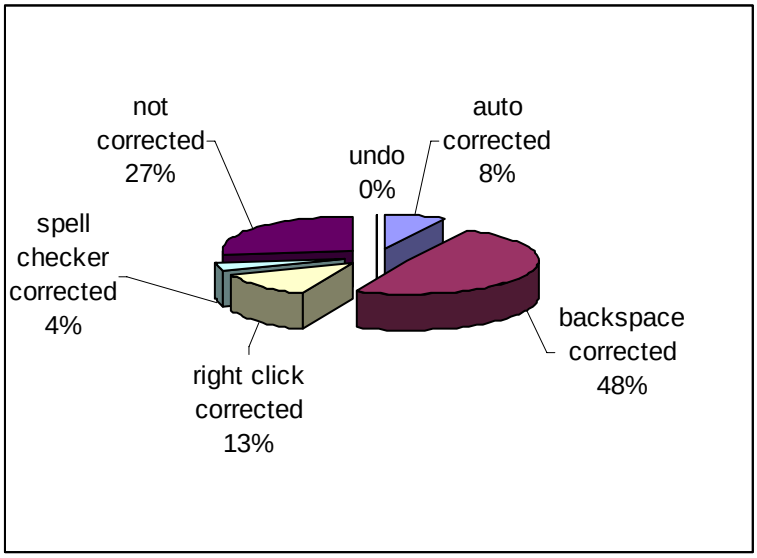


Figure 3.4 Percentage of Errors Corrected with Each error Correcting Technique

Another interesting aspect observed during the study was how many words the spell checker fixed automatically. The results indicate that the word processor attempted to fix 7.57% of the errors automatically. However, approximately 13% of the words automatically corrected were mistakenly corrected (see Figure 3.5).

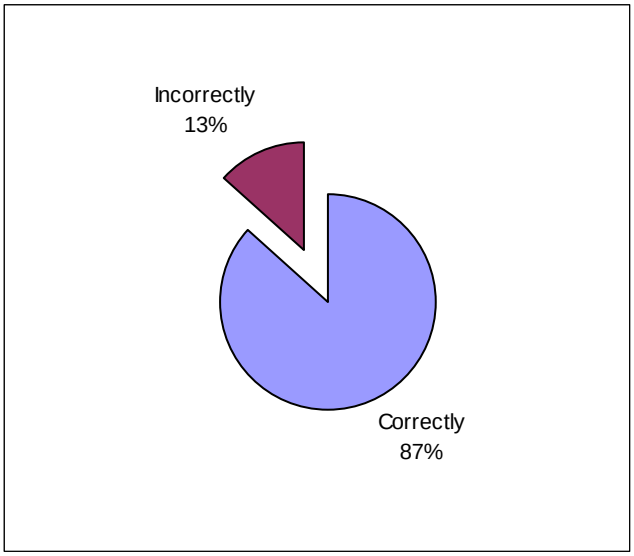


Figure 3.5 Percentage of Errors Automatically Corrected Correctly and Incorrectly by the Spell Checker.

The ways the Transposition & Disorder, Extra Letter, Missing Letter, Wrong Letter and Typographical errors were corrected are very similar as indicated in Figures 3.6 through 3.10 respectively. The large majority of these errors (80% to 91%) were corrected by the users mostly using backspace (76% to 89%). A very small number of errors (0% to 5%) were automatically corrected by the word processor. However, a small amount of these types of errors (0% to 20%) was left uncorrected.

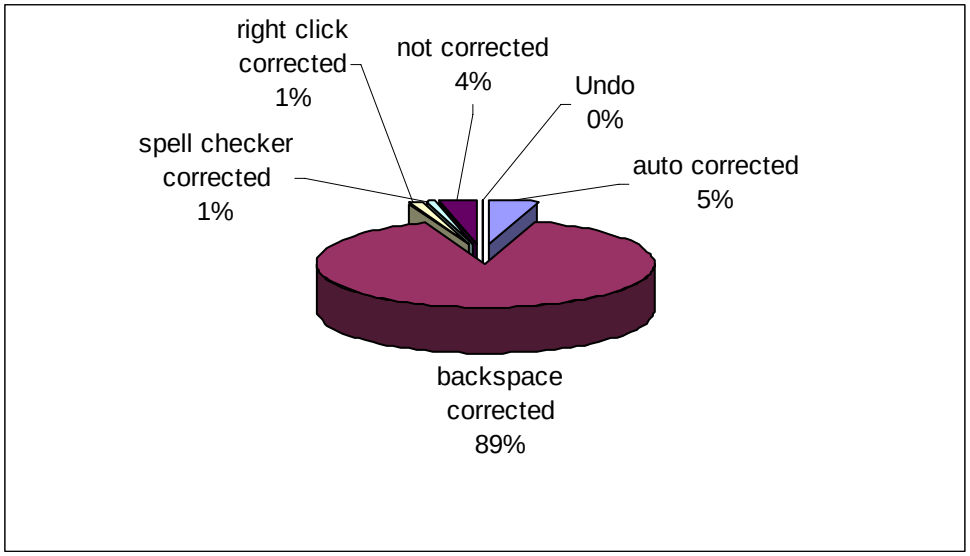


Figure 3.6 Correction of Transposition and Disorder Errors

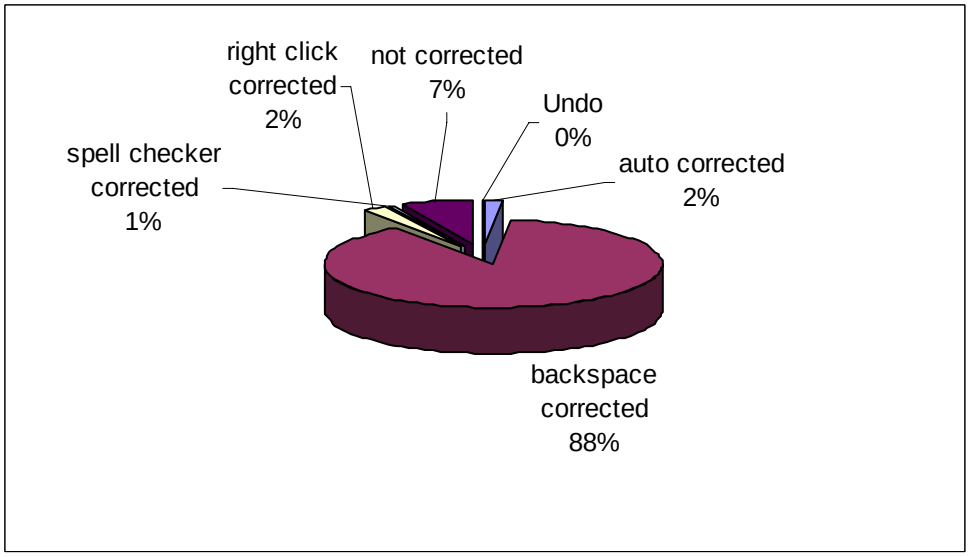


Figure 3.7 Correction of Extra Letter Errors

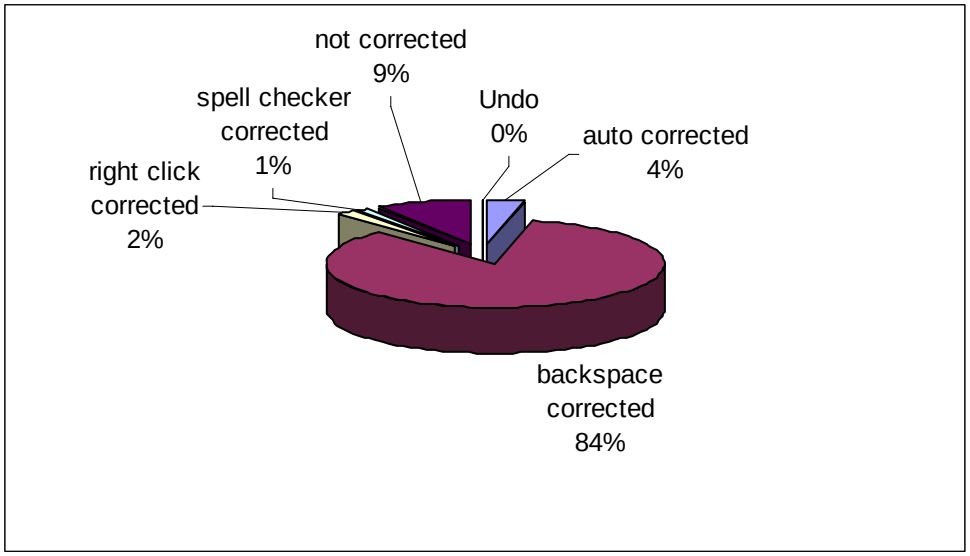


Figure 3.8 Correction of the Wrong Letter Errors

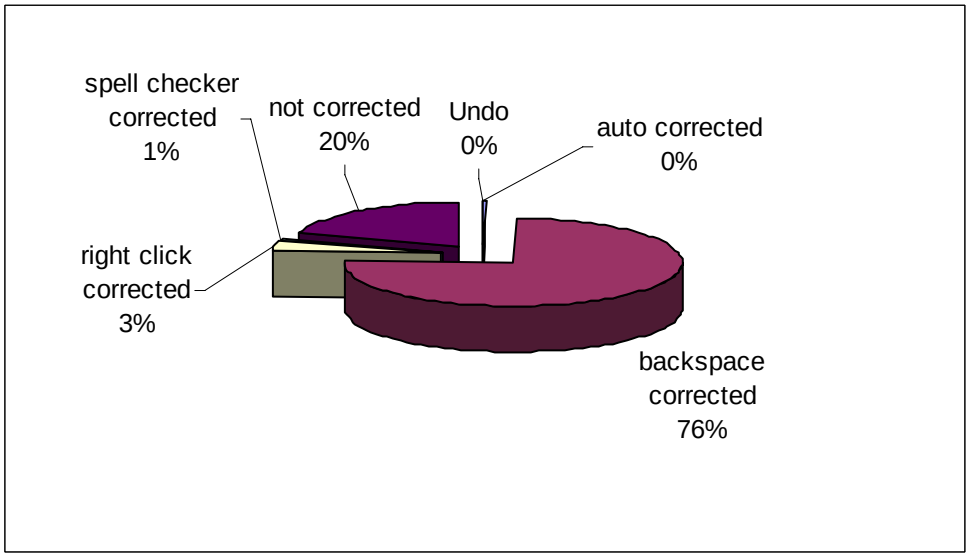


Figure 3.9 Correction of the Missing Letter Errors

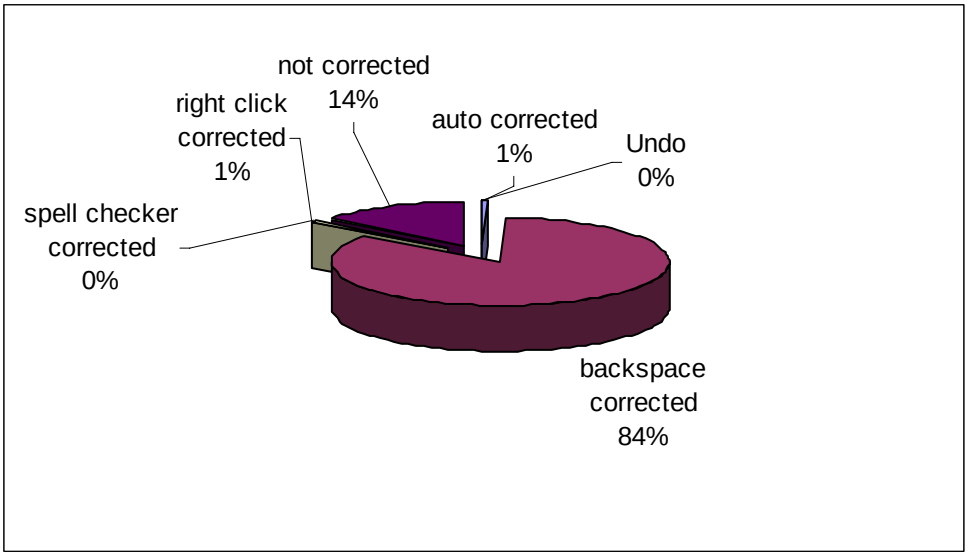


Figure 3.10 Correction of the Typographical Errors

As shown in Figure 3.11 the Homophone errors were corrected by the participants with different techniques and the auto correction feature of the word processor. However, about one third of the errors were not corrected.

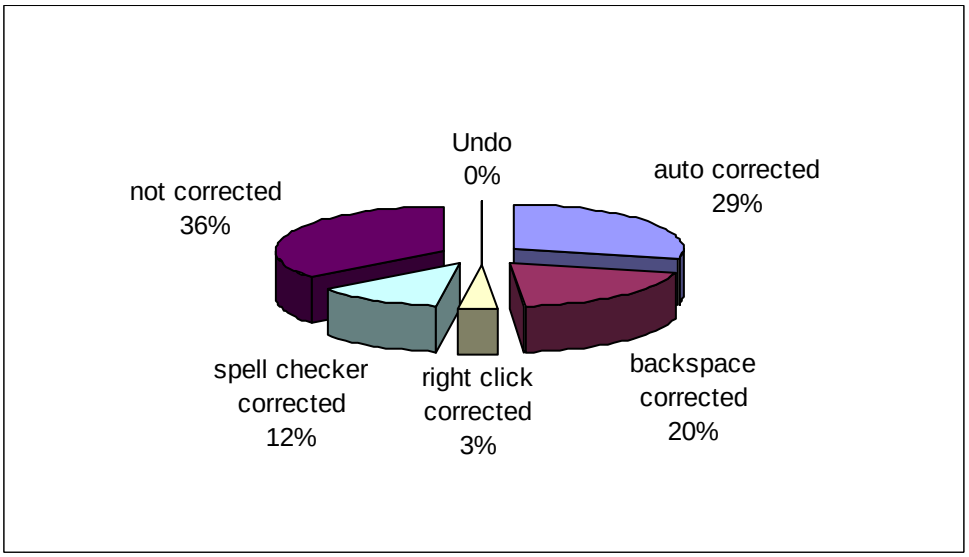


Figure 3.11 Correction of the Homophones Errors

Figure 3.12 shows that Grammatical Errors were never corrected. These errors were classified and identified by Spanish knowledge. There were three identified errors of this kind of error for the entire usability test. An example of this kind of error is in the following sentence: "… era algo de las cosas que nos atan para *acércanos* a Dios..." In this sentence the word "*acércanos*" must be substitute by the word "*acercarnos"*.

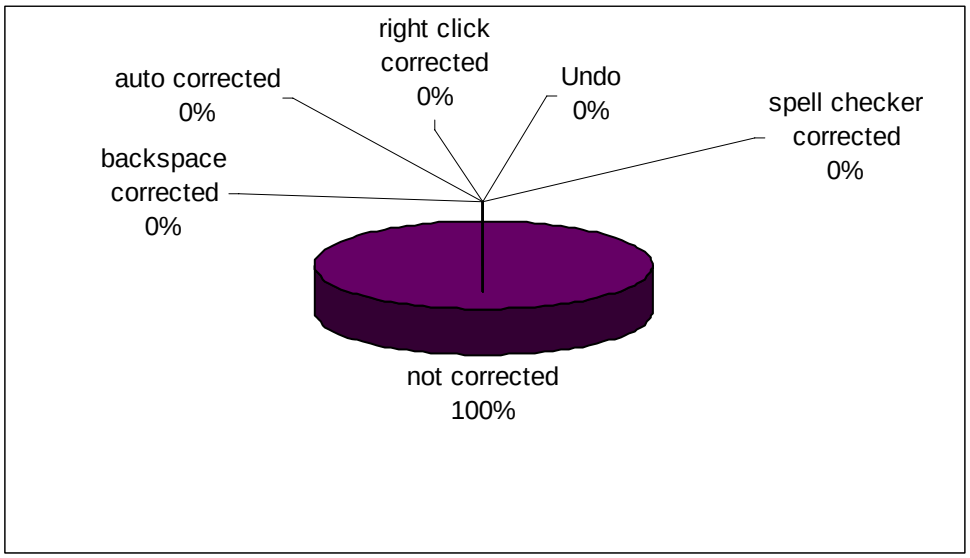


Figure 3.12 Correction of the Grammatical Errors

Figure 3.13 shows the different ways that Caps Lock errors were corrected. All of these errors were corrected. The large majority were corrected by the users using backspace and approximately one fourth of them were automatically corrected by the word processor.

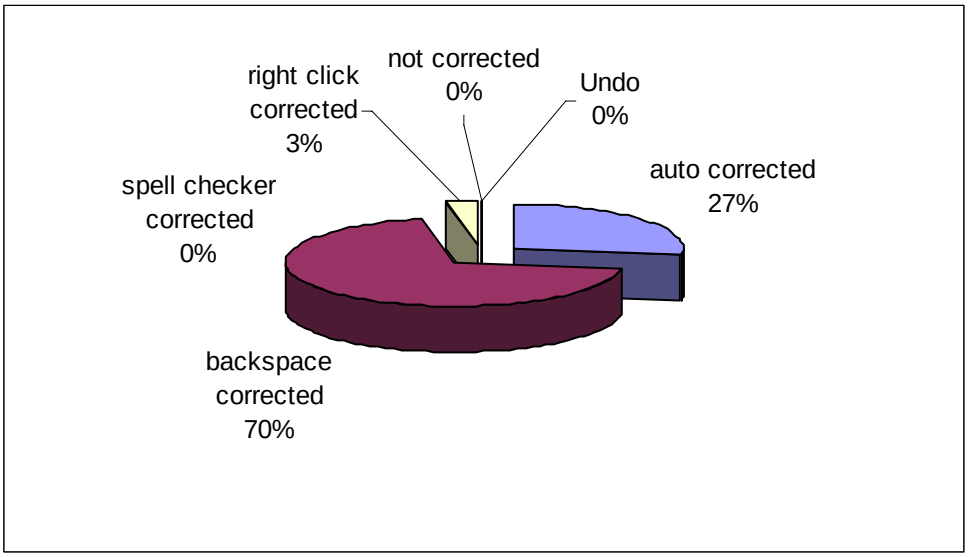


Figure 3.13 Correction of the Caps Lock Errors

The ways in which the Accent errors were corrected is shown in figure 3.14. Almost half of the errors were corrected by the participants using different techniques. A small number of these errors were automatically corrected by the word processor. However, more than one third of these errors were never corrected.

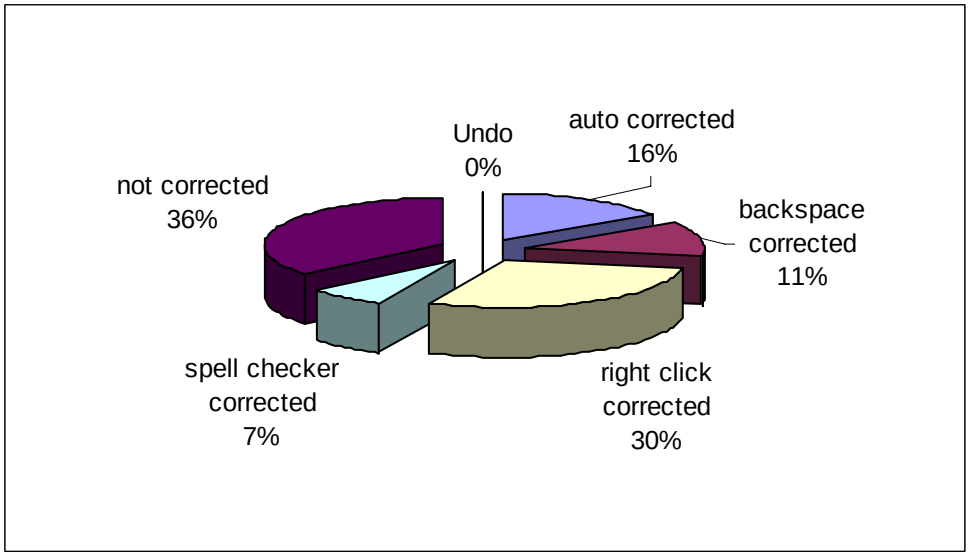


Figure 3.14 Correction of Common Accent Errors

Figure 3.15 shows that approximately three fourths of the Special Accent errors committed by the participants were not corrected. The rest of the words were corrected with different techniques.

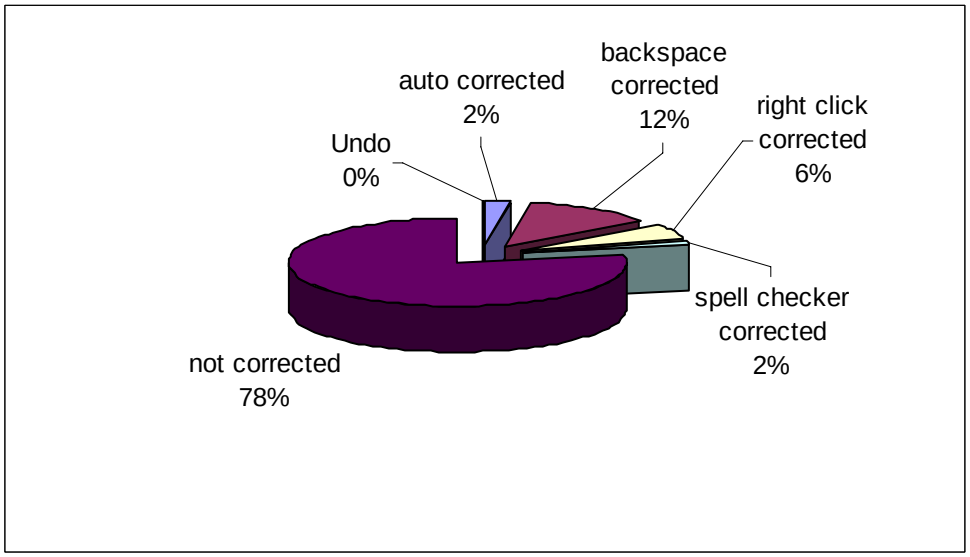


Figure 3.15 Correction of Special Accent Errors

Finally, figure 3.16 shows the different ways that the Ñ errors committed by the participants were corrected. Approximately half of these errors were corrected by the participants and a very small number by the word processor. However, almost half of the Ñ errors were left uncorrected.

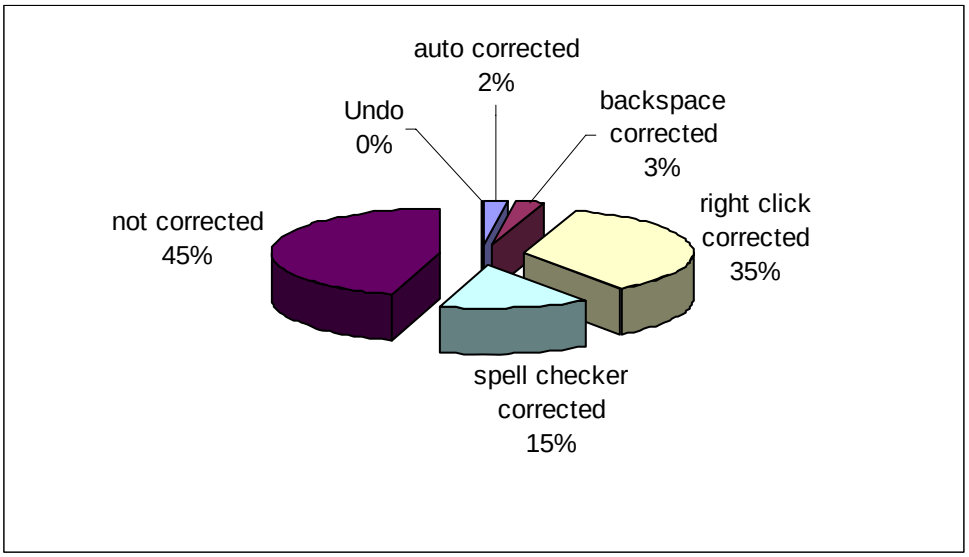


Figure 3.16 Correction of Ñ Errors

As we can see from this section, a large portion of special accent, common accent and, Ñ Error were uncorrected by the users. Some of them were not corrected because the spell checker was incapable of detect them. For this reason, we developed new algorithms to reduce the number of uncorrected errors. The algorithms are presented in chapter number four.

Chapter 4

ALGORITHMS

This chapter presents three algorithms that were developed to detect and help correct some of the errors identified with this study. Section 4.1 presents an algorithm that addresses the transposition and disorder letters problems. An algorithm to address the special accent problem is presented in section 4.2. Section 4.3 presents an algorithm to address the Ñ error.

4.1 Algorithm to Detect and Help Correct Transposition and Disorder Errors

Transposition is a very common error that a user commit while typing a document. This problem consists of exchanging the position of two adjacent letters. This kind of error can happen in any language. Another similar problem that we identify is words that have its entire letters but the letters are in disorder. For example, consider the following text:

“Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.”

When the previous text is spell checked by Microsoft Word the correction suggested for each word is summarized in Table 4.1. Microsoft Word does not suggest the correct

word in 16% of the misspell words. From that 16%, the speller did not offer any suggestion for 5% of the words and an incorrect suggestion for the other 11%.

Table 4.1 Suggestions Provided by Microsoft Word to Some Misspell words.

Incorrect Word	Correct Word	Suggestion by the MS Word Spell Check
Aoccdrnig	*According*	*Occurring*
Rscheearch	*Researcher*	*(no spelling suggestion)*
Cmabrigde	Cambridge	Cambridge
Uinervtisy	*University*	*(no spelling suggestion)*
Deosn't	doesn't	Doesn’t
Mttaer	Matter	Matter, mutter
What	What	What, with, want, wrath, watt
Order	Order	Order
Ltteers	Letters	Letters, litters, liters, letterers, otters
Word	Word	Word, wood
Only	Only	Only
Iprmoetnt	*Important*	*Pigment*
Thing	Thing	Thing, ting
That	That	That, tat, taut, tart, tact
Frist	First	First, frits, frost, fist, frit
Lsat	Last	Last, slat
Ltteer	Letter	Letter, litter, latter, later, liter
Rghit	Right	Right, grit
Pclae	Place	Place, plea, plane, plate
Rset	Rest	Rest, reset, ret
Total	Total	Total
Mses	Mess	Mess, muses, messes, messy, miss
Still	Still	Still, sill
Raed	Read	Read, raid, reed, rated
Wouthit	*Without*	*Outfit, outwit, worthy, within, wrought*
Porbelm	Problem	Problem
This	This	This, tins, ties, tics, tips
bcuseae	*Because*	*Busier, busied, busies, bureau, buckeye*
Human	Human	Human
Mind	Mind	Mind, main
Deos	Does	Does, duos, demos
Raed	Read	Read, raid, reed, rated
Ervey	Every	Every, eve
Lteter	Letter	Letter, litter, letters

Istlef	Itself	Istle, itself, isle
Word	Word	Word, wood
Wlohe	Whole	Whole, woe

The transposition problem is a special case of the disorder type of errors. Considering the nature of transposition and disorder errors it is possible to develop an algorithm that can detect both of them and suggest a correct word in most of the cases. With this objective in mind we developed an algorithm to address these types of errors. The first step in the implementation of the algorithm is to take all the words of a dictionary and put them into a hash table. This hash table contains two columns: Key column and the Value column. Each individual word will be placed in the Key column with the letter in alphabetical order. For example, for the word ***according*** the key column will be set as ***accdginor*** (See Figure 4.1). A vector of words that share the same key is placed in the Value column.

Key	*Value*
accdginor	according
aceeehrrrs	researcher
abcdegimr	Cambridge
ahtw	thaw, what
alst	last, salt, slat
⋮	⋮
keyword	Word1, Word2,…, Wordn

Figure 4.1 Configuration of a Hash table for Detecting and Help Correct Transposition and Disorder Errors

We studied the words in the English dictionary of OpenOffice in order to know how many keys exist. We found 55, 260 keys out of 62,076 words in the dictionary. Table 4.2 shows the frequency of keys with one or multiples words matching it. We found that 81% of the keys have just one word matching it. The extreme case found was a key with 9 matching words.

Table 4.2. Frequency of Keys with One or Multiple Words Matching It for an English Dictionary.

Number of Matching Words	Frequency of Key Words
1	50,244
2	3,848
3	759
4	270
5	76
6	46
7	13
8	3
9	1
10 or more	0

A similar study was conducted with a Spanish dictionary. In this case we found 64, 926 keys out 71, 934 words in the dictionary. Table 4.3 shows the frequency of keys with one

or multiples words matching it. We found that 90% of the keys have just one word matching it. The extreme case found was a key with 9 matching words.

Table 4.3 Frequency of Keys with One or Multiple Words Matching It for a Spanish Dictionary.

Number of Matching Words	Frequency of Key Words
1	58, 579
2	4, 417
3	900
4	260
5	86
6	34
7	14
8	4
9	2
10 or more	0

The algorithm developed takes a word and arrange the letters in alphabetical order. Then it hashes the resulting string with the hash table. If the table has a key that matches the string, then all the words stored in the value column of the hash table become alternatives for the misspelled word. If there are more than one alternative for the key word, the algorithm takes each of the alternatives and calculates the Hamming distance of the original word with

every the possible alternatives. The hamming distance is defined as the measure of the difference or "distance" between two words of equal length. The words are listed as alternative in order of Hamming distance. The alternative with the highest Hamming distances is the first one displayed.

To test the effectiveness of the algorithm, we fed it the incorrect words listed on Table 4.1 and our algorithm provided the correct alternative for each and every words. This contrasts with the Microsoft Word speller that could not provide the correct alternative for 16% of the words.

4.2 Algorithm to Detect and Help Correct the Special Accent Errors

A common problem found when writing in Spanish was the special accent error. This problem appears when a user types a word that has an accent and that word has an equivalent word without accent (i.e. cambie, cambié). The problem happens when a writer does not place an accent on a word that should has an accent or the writer places an accent on a word that should not have an accent. In both cases, both words are correct words of the dictionary but only one of them is correct in the context of a sentence. Since both words are correct the spell checker does not provide any indication that the written word is erroneous.

An analysis of the Spanish dictionary of OpenOffice revealed that there are 13,157 words with accents of which 3,307 have an equivalent word without accent. In addition there are 36 words out of those 3,307 for which there is another word with an accent placed in another vowel (i.e. este, éste, esté). There are also 19 with accents that have another

equivalent word with an accent in another vowel but no equivalent without accent. The remaining 9,723 words do not have an equivalent word without an accent.

We developed an algorithm to detect the special accent error as well as the common accent error. The algorithm creates a hash table with two columns: the key column has the words without accent and the value column have the corresponding words with accent for that specific keyword including the word without accent if there exist one. To create the hash table, we verified first if the word has an accent. If the word has an accent, we create the key of the word by simply removing the accent from that word. If the word does not have an accent, it is left unchanged. In order to create the hash table the algorithm creates a key for every accented word in the dictionary and a corresponding value object that is compose of two things: a vector of words and a Boolean flag that is initially set as False. For each word, the algorithm finds out is there is a key for that word on the hash table. If the key already exists, the algorithm adds the word to the vector of words and then set the Boolean flag to True. If the key does not exist, the algorithm creates the new key for the hash table. The accented word is added to the value object as well as the word without accent if there exist one in the dictionary. After this process is performed for all the words of the dictionary, the algorithm scans the table and removes all the words that have the Flag set as False. Figure 4.2 presents the configuration of the hash table.

Key	*Value*
informacion	información
multiplique	Multipliqué, multiplique
titulo	Título, tituló, titulo
informaran	Informarán, informaran
continue	Continúe, continué
⋮	⋮
keyword	**Word1, Word2,…, Wordn**

Figure 4.2 Configuration of a Hash table for Detecting and Help Correct Special Accent Errors

To detect special accent errors the algorithm takes a word a removes the accent if it was typed with an accent. Then it uses the word without accent to verify if there is a key in the hash table that matches that word. If a matching key is found, then the algorithm returns the words on the value column of the hash table as the possible alternatives for the word written by the user. Both, correct and incorrect words will have a warning to let the user know the different ways to write that word.

From the study, we identified 564 erroneous words that corresponded to the Special Accent error category. Of those, 419 have their corresponding correct word in the OpenOffice Dictionary. We ran the algorithm described above on those words and it produced warnings for all of them. The algorithm is able to detect 100% of the Special Error as long as the correct word is existing in the dictionary. This result contrasts very notably with the results of the study. In the study Microsoft Word was only able to detect 6.7% of these Special Accent errors.

The algorithm described in this section also captures Accent errors. From the study we identified 1365 erroneous words that corresponded to the Accent error category. Of those, 901 have their corresponding correct word in the OpenOffice dictionary. We ran the algorithm described above on those words and it produced warnings and the correct alternative for all of them. Thus, the algorithm is able to detect 100% of the Accent errors as long as the correct words are in the dictionary. This result is consistent with the results of the study because MS Word also detects 100% of the words that have the correct alternative in its dictionary.

4.3 Algorithm to Solve the Ñ Error

Another problem identified with the study was the Ñ Error. This error occurs when the writer has to type the ñ or the Ñ character and is unable to do so because in some cases the keyboard does not has a key for that character and in other cases the writer does not remember the combinations of keys that need to be entered to generate the character. Since the writers do not know how to type it, in most of the cases, they just type a "n" instead or a combination of symbols such as "~n" or "n~". This type of error is similar to the special accent error because in Spanish there are words that only differ from one another in that one has a ñ and the other an n instead (i.e. caña, cana). The OpenOffice dictionary has 1,279 words that have the ñ letter and fifty-four of them have another correct word that has an "n" instead of the "ñ".

We developed an algorithm to detect and help correct the Ñ and the special Ñ error. The implementation of the algorithm requires the creation of a hash table with two columns: a key column that has the words with an “n” instead of a “ñ” character, and a value column that consists of a vector of words and a Boolean flag that is initially set to False. To create the hash table, all the words in the dictionary are checked to determine if the word has a ñ character. If it does, the “ñ” letter is substituted for an “n” letter. Then the algorithm checks if a key for the resulting word exists. If the key already exist, the algorithm adds the word to the vector of words and sets the value of the flag to True. If the key does not exist, the algorithm creates a new key for the hash table. After this process is performed for all the words of the dictionary, the algorithm eliminates any key from the hash table who’s flag is set to False. Figure 4.3 presents an example of the configuration of the hash table for detecting the Ñ errors.

Key	*Value*
nino	niño
nina	niña
manana	mañana
mano	maño, mano
ordenar	Ordeñar, ordenar
⋮	⋮
keyword	**Word1, Word2,…, Wordn**

Figure 4.3 Configuration of a Hash table for Detecting and Help Correct Ñ Errors

To detect Ñ and special Ñ errors the algorithm takes a word and removes the ñ letter and substitutes it with an n. If the user types one of the following combination "~n" or "n~", the algorithm removes the tilde and create the word without the tilde. Then it uses the word without tilde to verify if there is a key in the hash table that matches that word. If the algorithm detects a key that matches the word, then it uses the words in the value column as the possible alternatives for that word.

From the study we identified 142 erroneous words that corresponded to Ñ error category. Of those, 72 have their corresponding correct word in the OpenOffice Dictionary. We ran the algorithm described above on those words and it produced warnings and the correct alternative for all of them. Thus, the algorithm is able to detect 100% of the Ñ errors as long as the correct words are in the dictionary. In comparison Word was able to detect 87% of the normal "Ñ" errors and only 3% of the special "Ñ" errors.

Chapter 5

DISCUSSION

The most significant finding of the study presented in this document was that a large number of the errors committed while writing in Spanish is related with words that have accented vowels (á, é, í, ó, ú) or a "ñ" character. The study revealed that the errors related with typing words with these characters (Accents, Special Accents and Ñ errors) constitute over 45% of all the errors committed by the participants. For obvious reasons these types of errors do not happen when writing in English. The errors that are common in the English language are the other eight error types identified with this study. In our study, if the errors related with the special characters are removed, the combination of Transposition, Wrong Letter, Extra Letter or Missing Letter errors constitute 77% of all the errors committed by the participants. These results are very similar to the Damerau study [Damerau64], that revealed that these four error types constitutes over 80% of all the error committed by the writers.

The Accents, Special Accents and Ñ errors occur mostly because the methods for typing the vowels with the accent and the ñ are very cumbersome in most computer systems and the writers do not remember how to do it. In the Windows platform these characters can be typed by pressing the Alt key a sequence of digits. Another way of entering these characters is using the English International Keyboard Setting. With this setting the writer holds the Right Alt key and presses the vowel they want to accent. It also allows writing the ñ letter just by holding the Right Alt and the n letter. On a system with a Spanish keyboard, the writer place accents by pressing the accent key and the vowel to be accented. Also there is a ñ key on the

Spanish keyboard. MS Word offers a Symbol map that includes the accent letters. In addition, Windows platforms have a Character Map under System Tools that include special characters such as letters with accent.

The result of the study revealed that approximately two thirds of the errors committed by the participants were corrected. In the large majority of the cases the errors were corrected using the backspace key. The remaining words were corrected using the spell checking features of the MS Word.

The study revealed that the ways in which the errors are corrected varies with the types of errors. The large majority of Wrong Letter, Extra Letter or Missing Letter, Typographical, Transposition & Disorder errors are corrected by the writers using backspace. This is because these types of errors are easy to identify by the writers and they corrected most of them on the spot. All the Caps Lock Errors were corrected either by the writers or the word processor. This is due to the fact that words in capital letters are easy to spot and users can identified them easily. On the contrast, none of the Grammatical Errors were corrected. This is because Grammatical Errors are errors in which another word is written instead of the intended word. Since the word written is a correct word the word processor does not detect the error and it passes unnoticed by the writers.

About half of the Accent errors are corrected by the writers and another small amount automatically by the word processor. Most of these errors were corrected with the speller features of the word processor (spell checker, right click and autocorrect). However, approximately one third of the errors were no corrected. The large majority of the Special

Accent errors were not corrected. This is because the MS Word spell checker does not detect the large majority of this type of errors and in most of the cases the writer is unaware of the problem. Most of these errors corrected were corrected with the speller features of the word processor (spell checker, right click and autocorrect). In the case of the Ñ errors about half of them were corrected by the writers and almost another half were left uncorrected. Some of these errors can be easily detected because of the presence of a tilde character (~) on the word. However, other cases are difficult to detect by the word processor and the writers because of the writers end up writing the word with an “n” instead of the “ñ” and that word is a correct word of the dictionary.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

The most important contribution of this study is identification of a profile of errors committed by people using a word processor to write in Spanish. Eleven error categories were identified. The most significant finding was that a large number of the errors committed are related with words that have a character such as á, é, í, ó, ú or ñ. These characters are very common in the Spanish language but inexistent in the English language. The errors caused when typing words with these characters were classified as Accents, Special Accents and Ñ errors. Most of these errors were committed because the methods for typing the á, é, í, ó, ú and ñ characters were very cumbersome and the writer usually did not recalled them. Thus, we conclude that the lack of straight forward support for special character of languages such as Spanish can caused a significant number of errors.

Our study produced similar results to the Damerau [Damerau64] study. If the Accents, Special Accents and Ñ errors are not considered, the percentage of the combination of transposition, wrong letter, extra letter and missing letter errors found in our study is very similar to the percentage reported in the Damerau. Thus, the Accents, Special Accents and Ñ errors are additional errors that are associated with the Spanish language. These findings supports the conclusion that writers can commit a significant number of additional errors

when writing in Spanish using Microsoft Word and a standard English keyboard than when they do it in English.

Another important finding of the study was that a substantial number of errors (approximately one third) are not corrected. Most of these errors pass undetected because the word processor does not detect them and thus does not provide a warning to the writers. From the study we identified that writers used basically three techniques to correct errors while writing in Spanish language: backspace, right click and spell checker. The backspace technique was used by the writers to correct approximately two thirds of all the words corrected. The writers used this technique to correct most of the errors that they detected at the moment they committed them. Thus, we conclude that writers correct most of the errors on the spot by recalling how the word is spelled correctly.

The study revealed that the percentage of errors corrected varies with the type of error. Error types that are easily identified by the writers or the word processor such as Wrong Letter, Extra Letter or Missing Letter, Typographical and Transposition & Disorder errors exhibit high percentage of correction. On the other hand, error types that are not detected by the word processor or easily identified by the writers such as Special Accent and Ñ errors exhibit a lower percentage of correction.

As it is documented in Chapter 4, with simple algorithms most of the Accent, Special Accent and Ñ errors can be detected. In addition the writer can be provided with alternatives to correct the error. The adoption of such algorithms by commercial word processor can improve error correction for Spanish writers.

6.2 Further Work

Existing spell checking software have the feature of changing words that consider erroneous for words that are assumed to be the correct words to be used. Word processors like Microsoft Word do no provide persistent feedback to warn the writer that a written word have been changed automatically. This action can cause a false positive situation because it may be the case that the original word was the correct word and the document end up with an incorrect word without the writer knowing it. In order to prevent this kind of errors we suggest a study of techniques for providing persistent feedback when a word has been changed automatically. In addition, the word processor should provide a straight forward way of placing the original word back into the text.

Another recommendation is to further expand the present study to include more participants and larger and diverse texts. We also suggest to conduct a similar study with English writers to study the differences and similarities among them. The study can also be replicated for writers of other languages.

References

[Bolshakov03] I. A. Boshakov, A. Gelbukh, **On Detection of Malapropisms by Multistage Collocation Testing,** 8th International Conference on Applications of Natural Language to Information Systems, June 2003.

[Damerau64] F. J. Damerau, **A technique for computer detection and correction of spelling errors.** *Comm*. ACM 7.3 (March 1964).

[Durham83] I. Durham, D. Lamb, J. Saxe, **Spelling Corrections in User Interfaces**, Communications of the ACM, Volume 26 Issue 10, October 1983.

[Fallman02] D. Fallman, **The penguin: using the web as a database for descriptive and dynamic grammar and spell checking**, CHI'02 Extended Abstracts on Human Factors in Computing Systems, April 2002.

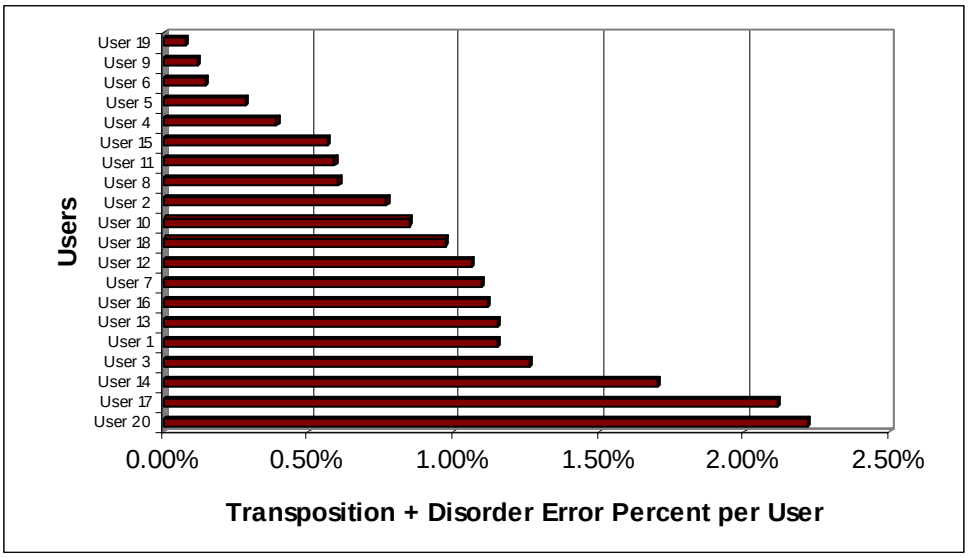
[Galletta05] D. Galletta, A. Ducikova, A. Everard, B. Jones, **Does spell-checking software need a warning label?** Communications of the ACM, Volume 48 Issue 7, July 2005.

[Hodge01] V. J. Hodge, J. Austin, **A Comparison of a Novel Neural Spell checker and Standard Spell Checking Algorithms**. Pattern Recognition, The Journal of the Pattern Recognition Society, Pattern Recognition 35 (2002) 2571 – 2580, July 2001

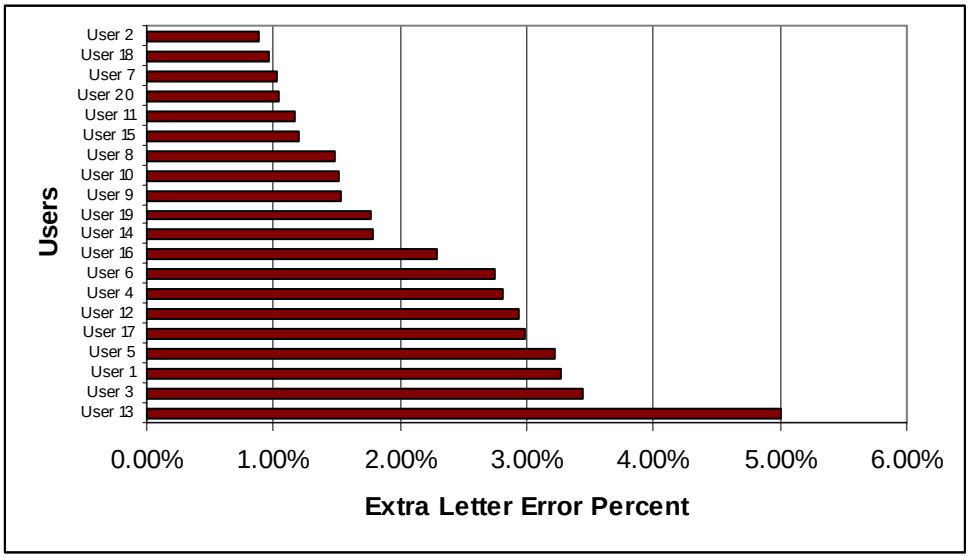
[Huang02] J. H. Huang, D. Powers, **Large Scale Experiments on Correction of Confused Words**. Computer Science Conference, Proceedings 24th Australasian, Pages 77 – 82, February 2001.

[Powers97] D. W. Powers, **Learning and Application of Differential Grammars**, CoNLL97: Computational Natural Language Learning, ACL Association for Computational Linguistics, pages 88 – 96, 1997.

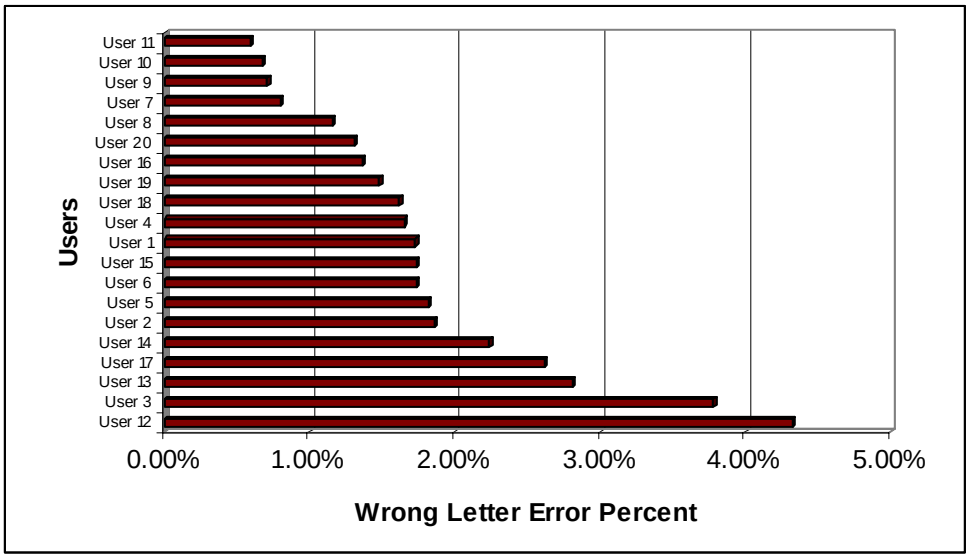
Appendix A:
Proportion of Words with Specific Errors



Proportion of Words with Transposition and Disorder Error by Participants

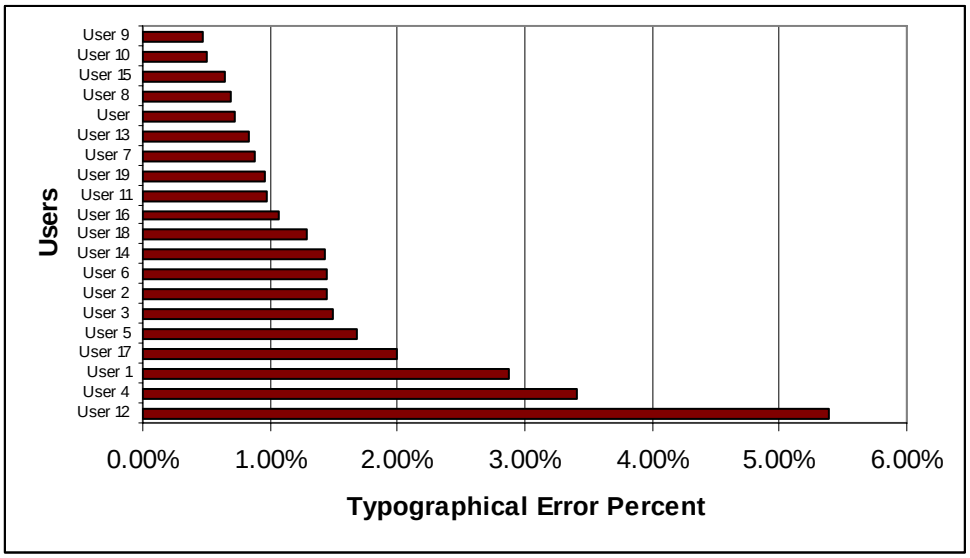


Proportion of Words with Extra Letter Error by Participants

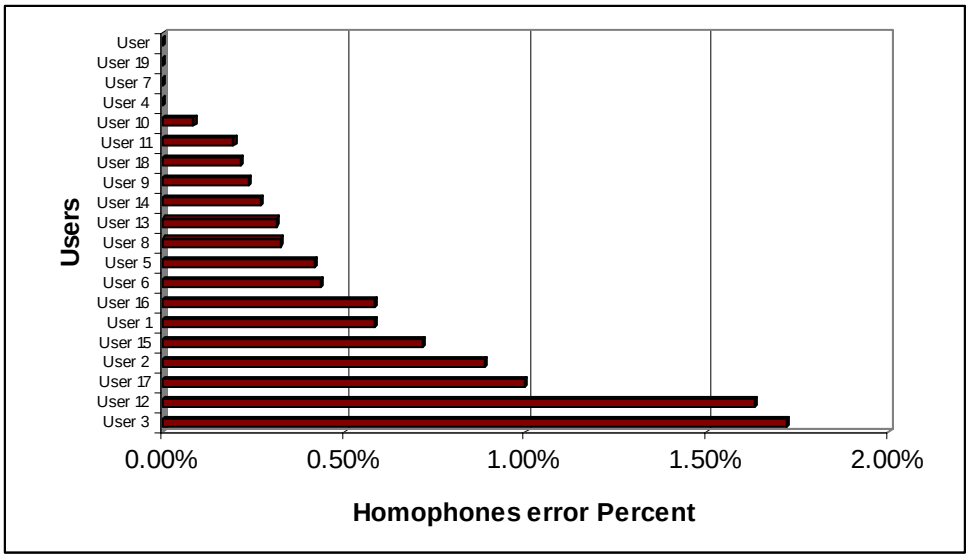


Proportions of Words with Wrong Letter Error by Participants

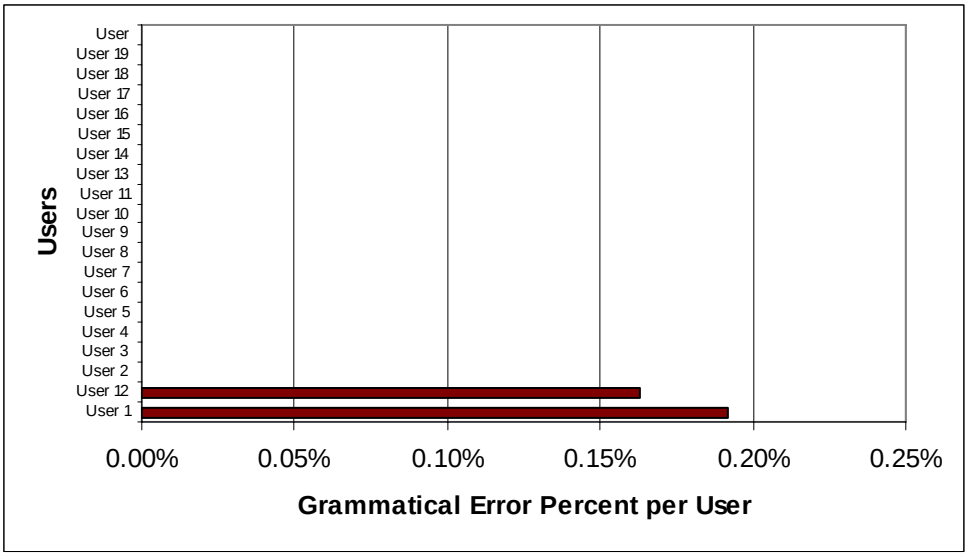
Proportion of Words with Missing Letter Error by Participants



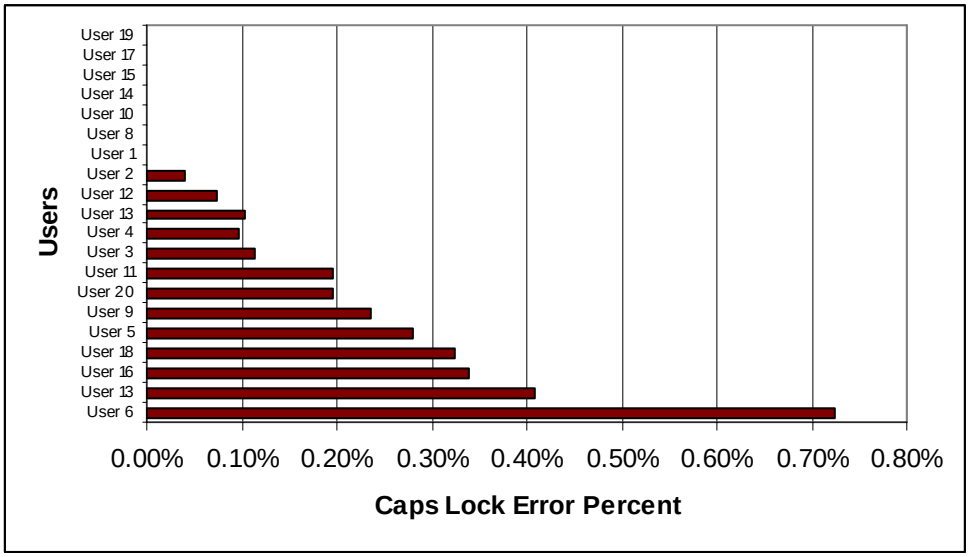
Proportion of Words with Typographical Error by Participants



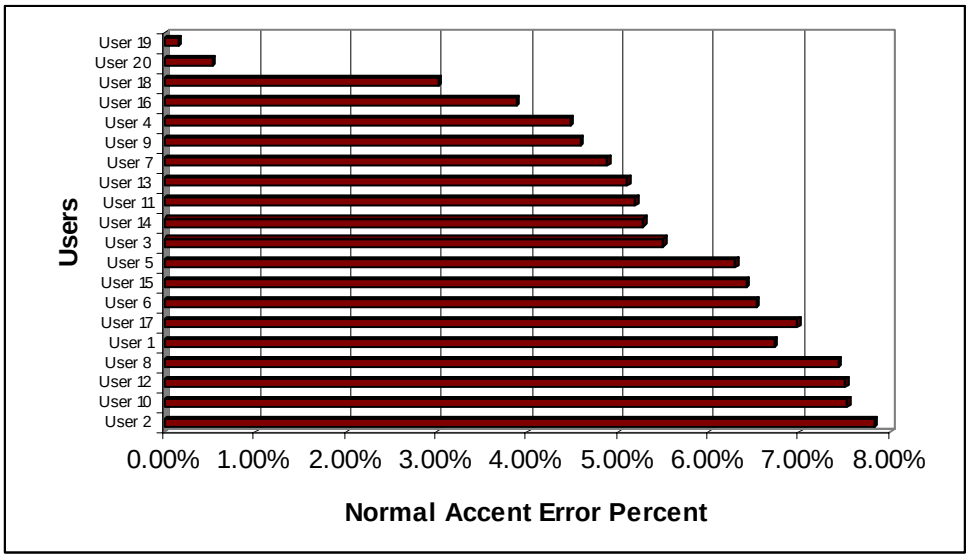
Proportion of Words with Homophone Error by Participants



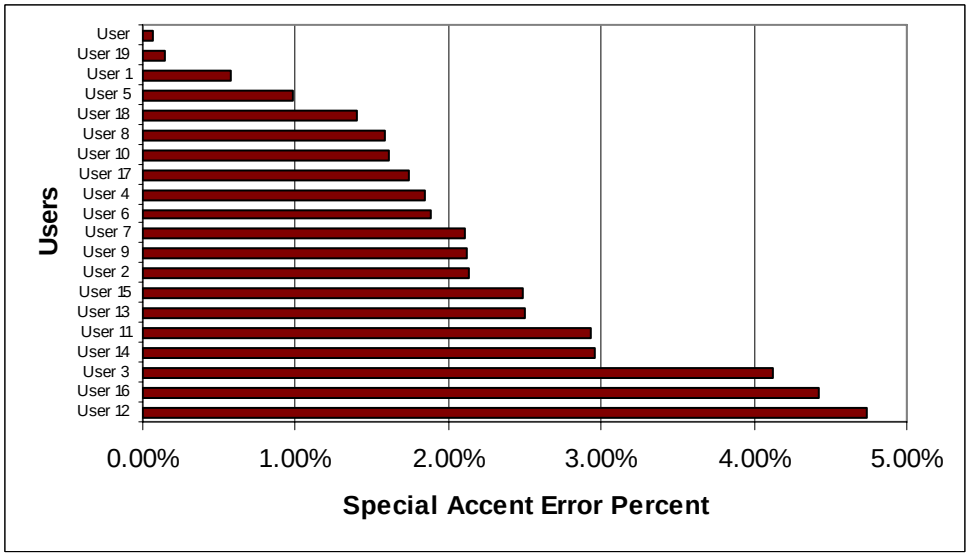
Proportion of Words with Grammatical Error by Participants



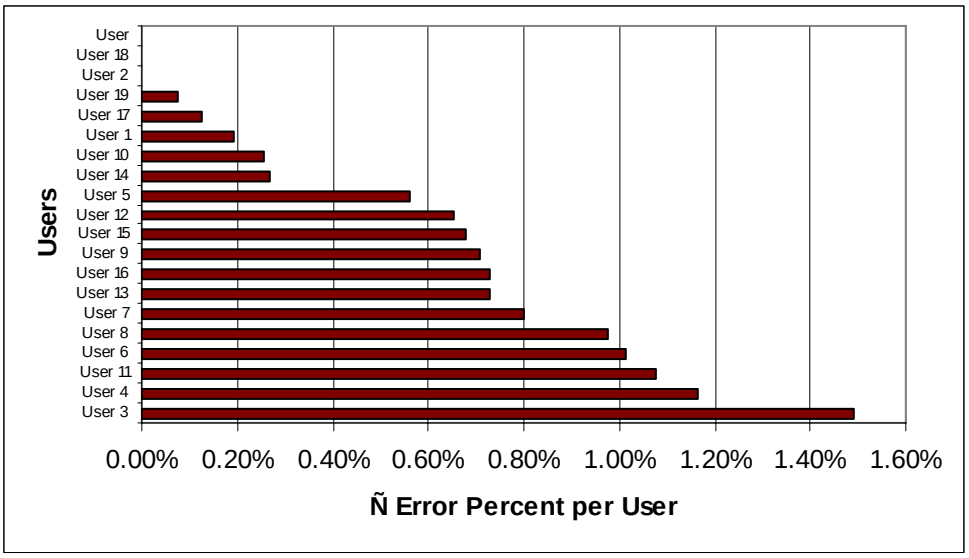
Proportion of Words with Caps Lock Error by Participants



Proportion of Words with Accent Error by Participants



Proportion of Words with Special Accent Error by Participants



Proportion of Words with Ñ Errors by Participants