A Spellchecker for Dyslexia

Luz Rello
HCI Institute
Carnegie Mellon University
luzrello@cs.cmu.edu

Miguel Ballesteros
LT Institute
Carnegie Mellon University
NLP Group
Universitat Pompeu Fabra
miguel.ballesteros@upf.edu

Jeffrey P. Bigham
HCI and LT Institutes
Carnegie Mellon University
jbigham@cs.cmu.edu

ABSTRACT

Poor spelling is a challenge faced by people with dyslexia throughout their lives. Spellcheckers are therefore a crucial tool for people with dyslexia, but current spellcheckers do not detect real-word errors, which are a common type of errors made by people with dyslexia. Real-word errors are spelling mistakes that result in an unintended but real word, for instance, *form* instead of *from*. Nearly 20% of the errors that people with dyslexia make are real-word errors. In this paper, we introduce a system called *Real Check* that uses a probabilistic language model, a statistical dependency parser and Google n-grams to detect real-world errors. We evaluated *Real Check* on text written by people with dyslexia, and showed that it detects more of these errors than widely used spellcheckers. In an experiment with 34 people (17 with dyslexia), people with dyslexia corrected sentences more accurately and in less time with *Real Check*.

Keywords

Dyslexia; Spellchecker; Real-Word Errors; Spelling Errors.

Categories and Subject Descriptors

K.4.2 [**Computers and Society**]: Social Issues—*Assistive technologies for persons with disabilities*; K.3 [**Computers in Education**]: Computer Uses in Education.

1. INTRODUCTION

Dyslexia is the most frequent language-based learning disability. 10% of the population has dyslexia [13], which represents from 70 to 80% of the language-based learning disabilities. Dyslexia is defined as a neurological specific reading and spelling disorder by the World Health Organization [34]. People with dyslexia have difficulty not only with reading but also with writing.

A main challenge is that people with dyslexia do not consciously detect spelling errors [26]. As a result, the text that people with dyslexia write contains more errors, even if they are trained; and adults with dyslexia keep making spelling errors without noticing [31]. Spelling errors can negatively affect how the people making those errors are perceived. For instance, it can lead to lower grades on school work, and cause people to think those making the errors are less intelligent.

Spellcheckers should help people with dyslexia make fewer errors, but they tend to miss real-word errors – a category of errors that people with dyslexia are especially likely to make. A real-word error is a correctly spelled word that is not the one the user intended to write. For instance, the sentence bellow has six spelling errors, but they are not detected by common spellcheckers because they are real words.[1]

> *We *sow *quit a *big *miss *take we *maid.*
> *We show quite a big mistake we made.*

In fact, a comparison of the most popular spellcheckers on different error types showed that real-word errors were the least flagged by the spellcheckers [23]. While grammar checkers promise to solve such problems in some cases eventually, the grammar checkers available in common software cannot detect many real-word errors now and many real-word errors are grammatical even if statistically unlikely. Moreover, real-word errors are frequent. Considering the errors written by people with dyslexia, 17% and 21% are real-word errors in English and Spanish texts, respectively [25, 28].

In this paper we present a method to detect and correct real-word errors in Spanish. To evaluate the usefulness of the method we performed three experiments. First, we evaluate the accuracy of the method using a corpus of real-world native Spanish speakers with dyslexia. Second, we compared the method with the most widely used spellcheckers. Third, we carried out an experiment with 34 people, 17 with dyslexia, to test the efficiency of the detections and the corrections using real sentences written by people with dyslexia.

The contributions of this paper are:

- a method to detect and correct real-word errors in Spanish;
- *Real Check*, a system that detects more real-word errors than the most widely use correctors;
- a study showing that *Real Check* makes people with dyslexia to correct sentences in a faster and more accurate way.

[1]For instance, Microsoft Word 2013 and LibreOffice 4.2.6.3 on Ubuntu 14.04 do not detect them.

ASSETS '15, October 26-28, 2015, Lisbon, Portugal.

http://dx.doi.org/10.1145/2700648.2809850.

2. RELATED WORK

Our work is related to *(i)* spellcheckers for dyslexia based on user modelling, and *(ii)* spellcheckers based on natural language processing (NLP) that target real-word errors.

2.1 User Modeling for Dyslexia

The first approach to apply user modeling to error correction for users with dyslexia was an adaptive corrector called *Babel* [30]. It contained new rules addressing the permutation of letters as well as a user model of the writer's spelling errors to adapt the detection and correction of typical errors to a specific user. Spooner [30] evaluated the spellchecker using text written by people with dyslexia and found out that it was more effective for some users.

Other Approaches. Li *et al.* [16] developed a model, *PoliSpell*, that aims at typical errors made by people with dyslexia, such as boundary errors and real-word errors. The authors plan to evaluate it in future work. Similarly, Gustafsson [10] developed a user model-based spellchecker for Swedish users with dyslexia, although it is yet to be evaluated.

2.2 Real-Word Error Correction

2.2.1 Semantic Information

Hirst and Budanitsky [12] developed a method using WordNet [20] together the semantic distance formula by Jiang and Conrath [15] to detect words that are potentially anomalous in context. The idea behind this approach in that words appearing together are semantically close. If the word was semantically distant from nearby words it was considered a potential error. The authors tested the method on an artificial corpus of English errors achieving from 23 to 50% recall and from 18 to 25% precision.[2]

2.2.2 Probability Information

Second, there are methods that use probability information using n-grams [14, 18, 33]. In these methods a word becomes an error candidate if the probability of the ngram within its context is lower than the one obtained by replacing one of the words with a spelling variation. The methods by Mays *et al.* [18], Wilcox-O'Hearn *et al.* [33] and Islam and Inkpen [14] used word-trigram probabilities for detecting and correcting real-word errors and they were all evaluated on the same corpus: 500 articles from the 1987-89 Wall Street Journal corpus where real-word errors were randomly inserted every 200 words. To the best of our knowlegde, the method by Islam and Inkpen [14] achieved the best results: 0.890 (R)[3] 0.445 (P) 0.593 (F1) for error detection, and 0.763 (R) 0.381 (P) 0.508 (F1) for error correction. Similarly, Verberne [32] proposed a trigram-based method where any word trigram occurring in the British National Corpus [5] is correct, and the rest are a likely error. The system was evaluation with part of the Wall Street Journal corpus, 7,100 words with 31 errors inserted every 200 words approximately. The results yield a 0.33 (R), 0.05 (P) and 0.086 (F1) for error correction.

2.2.3 Confusion Sets

Third, there are a number of approaches that rely on confusions sets. Confusion sets are pre-defined sets of commonly confounded words, such as $cruse|crews|cruise|curse$ or $from|form$. Once a word belonging to confusion set appears in a context, these methods perform different types of word sense disambiguation to predict which member of that confusion sets is the most appropriate for that context. There are rule-based methods [24, 25] as well as machine learning approaches [7, 9]. Using a corpus of texts written by people with dyslexia as development dataset, Pedler [25] designed three rule-based methods: a frequency-only based system, a syntax-based (using bigram probabilities), and semantic method (using WordNet). The author evaluated the best performing method (the semantic one) using 6,000 confusions sets and two corpora written by students, containing 199 and 1,049 real-word errors, respectively. For each of the corpus results were 31.1% and 23.4% (R); 83.3% and 77.2% (P) for error detection, and 70.3% and 60.3% (P) for error correction. To the best of our knowledge the machine learning approaches achieves the highest results –but using automatic corpora [7, 9]. These methods were tested running confusion sets on Wall Street Journal corpus and results are presented for each of the confusion sets. Golding and Roth [9] method detects about 96% of context-sensitive spelling errors, in addition to ordinary non-word spelling errors. Carlson *et al.* [7] method improved the scalability of the previous method as well as its performance reaching to 99% of accuracy for some confusion sets.

Commercial tools. We found a commercial spellcheckers that aim at English real-word errors either for people with dyslexia[4] or general population.[5] We could not find any documentation about how these commercial tools were developed.

2.3 What is Missing?

The method we present in this paper differs from previous work in the following aspects: First, it is for Spanish while the previous methods are for English. Second, it advances previous NLP approaches combining the probabilistic information from Google Books Ngram Corpus –using also 4- and 5-grams– with Spanish confusion sets together with a language model and rules over dependency parse trees. Third, it is evaluated by 34 people with and without dyslexia using –not an artificial corpus– real texts written by people with dyslexia.

3. METHOD

First, we collected texts written by native Spanish speakers with dyslexia to be able to develop and evaluate our method. Second, we developed an algorithm that works in 3 steps: *(i)* confusion set extraction, *(ii)* n-gram matching, and *(iii)* two filters, a language model and a dependency parser. For the development of the system we used a small subset of 22 sentences held-out from the final evaluation (Section 4.1).

3.1 Crowdsourcing Dyslexic Errors

Since written errors by people with dyslexia differ from regular spelling errors [25, 31], we collected text written by people with dyslexia to develop and evaluate our spellchecker. To do so, we made a public call to look for volunteers via the main associations of dyslexia of a Spanish native speaking country. We recruited two groups of people, one in Madrid

[2]Definitions of precision and recall are given in Section 4.2.
[3]R stands for recall, P for precision and F1 for F-measure.

[4]Ghotit Dyslexia Software: `http://www.ghotit.com/`
[5]Ginger Software: `www.gingersoftware.com/`

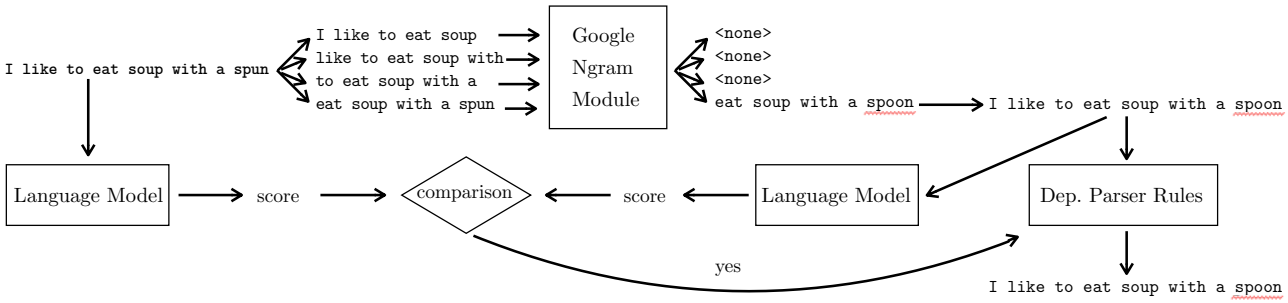

Figure 1: Workflow of the algorithm.

and one Barcelona, with whom we met to collect real-word errors. We asked them to bring with them texts written by people with diagnosed dyslexia and a laptop to introduce the errors using an on-line form. The volunteers were eight adults with diagnosed dyslexia, seven mothers and one father with children with dyslexia -four of them came with their children (eight children)-, and two secondary school teachers. We collected real-word errors with-in the sentence they appear from old notebooks, school essays, and texts written with a computer but without using spellcheckers. We gathered almost 600 sentences candidates but only keep 366; we discarded (a) the sentences which did not have real-word errors (such as typos or possible words but do not appear in the dictionary); (b) ungrammatical sentences; and (c) sentences that contained repeated errors. The selected sentences had at least one real-word error (1.12 errors per sentence) and their length ranged from 5 to 15 words.

3.2 Creating Confusion Sets

We analysed the errors linguistically [27], and found a simple set of rules would cover the great majority of real-word errors. Words that tended to be confounded differ from one to another in only one or two characters, or in the order of the characters, such as *casa ('house')* and *saca ('take out')*. Since Spanish has a shallow orthography, that is, the orthography of a word represents its pronunciation in a transparent and regular way, we could compile a list of confusion sets automatically using a Spanish dictionary[6] without requiring phonetic transcriptions. To this end, we used the Levenshtein Automaton dymamic algorithm [29].[7] We extract for each word all possible candidate words at Levenshtein distance 1 if the original word has 4 characters or less, and at Levenshtein distance 1 and 2 if it has more than 4 characters. We also calculate all possible anagrams for a word that occurs in the dictionary. As a result we had a resource of 1,250,781 different confusion sets in Spanish.

3.3 N-gram Matching

Once we have a confusion set for each word in the sentence, our system performs n-gram matching on the Google Books Ngram Corpus (2012 edition). This corpus consists of n-grams and their usage frequency over time,[8] and is derived from 8,116,746 books, over 6% of all books ever published. The corpus has 854,649 volumes and 83,967,471,303 tokens in Spanish [17].

Our algorithm works as follows (Figure1), given a sentence we generate all possible sentences in which each word has been changed for every word in its confusion set, we call them candidate sentences. The system extracts all possible 5-grams of the original sentence and all the candidate sentences. For each 5-gram the system extracts all possible 3-grams within the 5-gram and all possible 4-grams within the 5-gram. Finally, it checks for the existence of the n-grams[9] for the original sentence and for each candidate sentence. Candidate sentences that are more frequent due to 4-grams count twice as candidates found for 3-grams. In other words, if the candidate sentence is more frequent according to the n-gram corpus, we mark the candidate word to be a "suggested" correction and the original word to be a detected real-word error.

The system selects at most 7 suggestions for a single word. If there are more than 7, it filters out the less frequent ones, based on frequency in the Google Books Ngram Corpus, so it basically selects the top most frequent suggestions.

3.4 Filter-1: Language Model Filter

The third step of the algorithm is a language model,[10] we use the BerkeleyLM [22] with the pretrained Spanish n-gram model[11] trained on the WEB-1T corpus [3]. We use the language model to filter out ungrammatical suggestions.

The filter works as follows: once the system has found a candidate word, we run the language model to get the score of the candidate sentence that has the original word replaced by the candidate one. If the score is lower than the score of the original sentence, then the candidate is filtered out, otherwise it keeps going (Figure1). This filtering approach is similar to the one implemented in the grammatical error correction system by Felice *et al.* [8]. In our case only sentences in which stop words[12] have been changed are evaluated with the language model, since we found that, in

[6]We use the GNU aspell dictionaries which are freely available via `http://aspell.net/`

[7]`https://github.com/klawson88/LevenshteinAutomaton`

[8]`http://books.google.com/ngrams`

[9]Being n, either 3 or 4. We found that the 5-gram approach did not find any of the 5-grams included in the development set, so we decided to only explore 3-grams and 4-grams.

[10]A language model is a system that assigns a score to a sentence according to a probability distribution. In other words, it predicts whether a sentence belongs to a language or not, so it gives higher scores to sentences that truly belong to that particular language, in our case, Spanish.

[11]It can be found at `http://tomato.banatao.berkeley.edu:8080/berkeleylm_binaries/`

[12]We use the list of stop words that is freely available at `https://code.google.com/p/stop-words/`

preliminary experiments on the development set (see Section 4.2), the language model did not help for other kind of real-word errors.

3.5 Filter-2: Dependency Parsing

As a fourth step and another way of filtering erroneous candidates, we use the joint dependency parser, lemmatizer, part-of-speech tagger and morphological tagger of Bohnet and Nivre [2] trained on the Spanish CoNLL-2009 data [11].[13]

A dependency parser outputs a syntactic dependency tree of a sentence, and this one, in particular, provides lemmas, part-of-speech tags and more fine grained morphosyntactic tags. By using the dependency tree, the part-of-speech tags, the lemmas and the morphology, we came up with a set of rules that filters the suggestions found in the Google Ngram Corpus and that are better ranked according to the language model. The rules are the following: (1) if the suggested word is a verb and is not in participle/gerund form but it has as syntactic head a token with lemma *haber,'to have'/estar, 'to be'*, then it is filtered out; (2) if the suggested word has been tagged as singular/plural, but it is the head of a determiner which is plural/singular, then it is filtered out; and (3) if the suggested word has been tagged as singular/plural, but it is the head of a plural/singular determiner, then it is filtered out.

4. EVALUATION

We evaluated our method in three ways: (i) an evaluation of the system using dataset that consists of sentences written by people with dyslexia; (ii) an evaluation of the system in comparison with widely used spellcheckers; and (iii) a user evaluation with 34 participants.

4.1 Evaluation Datasets

We compiled 3 different datasets: *(i)* a development set consisting of 22 sentences with errors that is held-out from the rest of the experiments; *(ii)* a evaluation dataset consisting of 344 sentences used for the system evaluation and the comparison with other spellcheckers; and *(iii)* a subset of *(ii)* of 37 sentences used for the user evaluation.

4.2 System Evaluation

The development set was only used to the development of the system: to tune the language model parameters and threshold described in Section 3.4 and to come up with the rules depicted in Section 3.5. During the development of the system, we applied both filters and a 3-4-5-gram approach. We managed to achieve good results for both the correction and detection of real-word errors on the development set (Detection: F1=0.67. Correction: F1=0.83), by manually enriching the confusion sets, setting up thresholds for the language model,[14] implementing the rules over dependency trees, and by deciding to remove the string matching for 5-grams since it did not yield to any improvements.

Once we had the system tuned to perform well in the 22 sentences of the development set, we carried out the final experiment on the test set for evaluation.

[13]The results –including punctuation symbols– of the parser are 98.82 for POS accuracy, 98.02 for morphology tagging, 92.70 for lemma prediction, 88.04 LAS and 91.22 UAS [11].

[14]The best threshold ended up being 0, and this is why we did not describe it in detail in Section 3.4.

Real-word Error Detection							
Real Check	TP	TN	FP	FN	P	R	F1
3	**256**	1404	317	**105**	44.68	70.91	54.82
4	158	1623	98	203	61.72	43.77	51.22
3-4	**256**	1409	312	**105**	45.07	**70.91**	55.11
3-4-LM-DP	238	1487	234	238	50.42	65.93	**57.14**
TextEdit	17	1719	2	344	89.47	4.71	8.94
Pages	26	1719	2	335	92.86	7.20	13.37
OpenOffice	17	1719	2	344	89.47	4.71	8.95
MS Word	21	**1721**	**0**	340	**100.0**	5.82	10.99
Google Docs	135	1718	3	226	97.83	37.40	54.11
Real-word Error Correction							
Real Check	TP	TN	FP	FN	P	R	F1
3	136	1404	437	**105**	23.73	56.43	33.42
4	120	1623	136	203	46.88	37.15	41.45
3-4	**137**	1409	431	**105**	24.12	**56.61**	33.83
3-4-LM-DP	121	1487	351	123	25.64	49.59	33.80
TextEdit	10	1719	133	344	6.99	2.82	4.02
Pages	14	1719	220	335	5.98	4.01	4.80
OpenOffice	13	1719	195	344	6.25	3.64	4.60
MS Word	11	**1721**	237	340	4.44	3.13	3.67
Google Docs	127	1718	**11**	226	**92.03**	35.98	**51.73**

Table 1: Real-word error detection (above) and correction (below) results. 3 is a system that works with 3-grams. 4 works with 4-grams. 3-4 works with 3-grams and 4-grams. LM-DP means that the system has the filters implemented (dep.parser and language model). Best results are presented in bold.

In Table 1 we depict the results of each of the components of the system standing alone and in combination, showing the results of a simple 3-gram approach, a 4-gram approach, a 3-gram-4-gram approach, and the latter with the parsing and the language model filters. The table shows the number of true positives (TP), number of true negatives (TN), the number of false positives (FP), the number of false negatives (FN), precision (P) which is calculated as P=TP/TP+FP, recall (R) which is calculated as R=TP/TP+FN and the F1 score (F1) which is calculated as F1=2TP/(2TP+FP+FN). Note that, following [14], in the error correction task, if any of the suggestions given for a word is correct then it is considered as a TP.

4.3 Comparison with Spellcheckers

We compare our model with spellcheckers that are implemented in widely used word processing software: Microsoft Word,[15] Google Docs,[16] OpenOffice,[17] Pages,[18] and TextEdit.[19] We could not include commercial spellcheckers that aimed at real-word errors (Section 2) because they are only available for English.

As shown in Table 1, our system(s) achieved competitive results for both the detection and correction of real-word errors. The inclusion of a language model leads to a more precision oriented system, which can be useful depending on the task. The only spellchecker that provides competitive

[15]`https://products.office.com/en-us/word`

[16]`http://docs.google.com`

[17]`https://www.openoffice.org/`

[18]`https://www.apple.com/mac/pages/`

[19]`https://developer.apple.com/library/mac/samplecode/TextEdit`

results compared to us is the one implemented under the Google Docs on-line tool that it is better in the correction of errors. With an implementation towards a high precision and lower recall, it is capable of detecting half of the errors and it provides high precision results. However, it is worth noting that our implementation outperforms its results for error detection, and we therefore correct more errors.

4.4 User Evaluation

To evaluate the effect of the spellchecker on the correction of texts we conducted an experiment with 34 participants divided into two groups: participants with dyslexia and strong readers. Objective measures were collected using an on-line test in which each participant had to correct 37 randomly selected sentences containing real-word errors. Subjective measures were gathered via questionnaires.

4.4.1 Design

In our experimental design, *Correction Type* served as an *independent measure* with three levels: *None* denotes the condition where the text was presented without using any spellchecker method; *Error Detection Only* denotes the condition where the text was presented to the participant enabling only the option that highlights the errors detected by our spellchecker (Figure 2, up); and *Error Suggestions* denotes the condition where the text was presented to the participant enabling all the options of our system, highlighting the error candidates and showing suggestions for the correction of the error candidates (Figure 2, down).

We used a within-subject design, that is, each participant contributed to the three conditions. To cancel out order effects, the sentences and the conditions were presented to the participant shorted randomly.

For quantifying the efficiency of the participants using our corrector we defined four *dependent measures*: two objective and two subjective. We collected the objective measures from the interaction with our on-line test, while subjective measures were collected by self-report questionnaires using 5-point Likert scales. The dependent measures are:

Writing Accuracy: the correctness degree of the sentence. We measured correctness of the sentences in a scale from 1 to 100. If the sentence was left with no modifications we raked the accuracy as 0. If the sentence was perfectly correct matching the intended sentence *Writing Accuracy* equals to 100. For the cases that did not fall into these categories (140 out of 1,258 data points) we manually analysed the input of the participants to identify the following categories from the input responses:

- *Punctuation and capital letters:* where the participant changed the punctuation and capitalization of the sentences. Since the sentences were correct, they had a score of 100.
- *Semantic errors:* where the participants detect the error and wrote another word correctly but is was not the intended word. Since these sentences were correct we also gave them 100 score. *e.g. un café tostado ('a roasted coffee')*, instead of *un café cortado ('one coffee machiatto')*.
- *Accentuation:* where the participant detected the erroneous word but made an accentuation mistake when writing the intended word. *e.g. raiz* instead of *raíz ('root')*. The score of these errors were 75.
- *Typo:* where the participant detected the erroneous word but made a typo mistake when writing the intended word. These cases had a score of 25.
- *Wrong:* where the participant included more erroneous words in the sentence than in the input. These sentences had a score of 0.

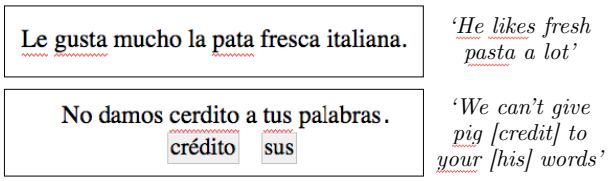

Figure 2: Screenshots of two sentences used in the experiment presented under the conditions of *Detection Only* (up) and *Suggestions* (down).

Correcting Time: Time in seconds that the participant spent correcting the sentence. It was calculated in milliseconds starting from the time that the participant is exposed to the sentence until they click submit. Once the sentence was modified by the participant, there was no indication to show whether the sentence was correctly corrected or not.

Subjective Writing Accuracy For each of the conditions the participants had to rate how accurate they found their performance correcting the sentences. For instance, *Using the word suggestions I corrected the sentences... 1* = Very bad *to 5* = Very good.

Subjective Writing Time For each of the conditions the participants had to rate how fast they found their performance correcting the sentences. The scores range from 1 = *Very slow* to 5 = *Very fast*. For instance, *Using the highlight option I corrected the sentences... [scale].*

4.4.2 Participants

We recruited 34 participants divided into two groups: 17 with dyslexia and 17 'strong readers' —people without dyslexia who were familiar with Spanish orthographic rules and able to evaluate the corrector.

The average age of the participants with dyslexia was 38.65 with a standard deviation of 11.19, and their ages ranged from 20 to 50 years old. The ages of the strong readers ranged from 25 to 58 (M= 38.29, SD = 12.92). All the participants were Spanish native speakers.

Except from one participant, all of them used spellcheckers frequently when: 3 participants used the iOS corrector installed in Pages word processor, 2 did not know because they used LaTeX and the corrector depends on the packages, 1 participant used the spellchecker from Google Docs and the rest (27 participants) used Microsoft Word spellchecker.

With the exception of two participants with dyslexia who dropped high school, 4 participants had finished high school, 2 and 4 participants had college and university degrees, respectively; and 4 had professional education. On the other hand, all the strong readers except from 2, who finished college, have finished their university degrees.

4.4.3 Materials and Procedure

Evaluation Dataset. We randomly extracted 37 sentences, that is 11% from the text set (344 of sentences written by people with dyslexia, Section 3.1) and run *Real Check* over the sentences. The corrector output served as an evaluation dataset.

Test. The evaluation dataset was integrated in an on-line test implemented using json. Each of the sentences were presented randomly within a *Correction Type* condition.

Once the participants agreed to take part in the study, we gave them specific instructions. Then, they first took the on-line test were they had to fix the sentences, and second the completed an on-line questionnaire.

4.4.4 Results

First, we use the Shapiro-Wilk test for checking if data fits a normal distribution and the Bartlett's test to check for homogeneity. For both groups, Shapiro-Wilk tests showed that the six datasets (one per condition per group) were not normally distributed for the *Correcting Time*. Similarly, Shapiro-Wilk tests showed that no datasets were normally distributed for the *Subjective Correcting Time* of both groups. Also, Bartlett's tests showed that none of the datasets had an homogeneous variance for all the measures and both groups.

As our data was not normal nor homogeneous, we include the median for all our measures in addition to the average and the standard deviation. For the same reason, to study the effects of the dependent variables we used the two-way Friedman's non-parametric test for repeated measures plus a complete pairwise Wilcoxon rank sum post-hoc comparison test with a Bonferroni correction that includes the adjustment of the significance level. We used the same procedure to show effects of the conditions within groups, dividing the data for each group. For the Likert scales we also used non-parametric tests [6].

Table 2 summarizes the main statistical measures for each of the conditions per group.

Writing Accuracy. There was a significant effect of *Correction Type* on *Writing Accuracy* ($\chi^2(2) = 10.015,\ p = 0.007$). The results of the post-hoc tests show that:

- **Between Groups:** Participants with dyslexia had significantly lower writing accuracy under all conditions ($M = 86.72, Mdn = 100.00,\ SD = 32.43$) than the stronger readers ($M = 93.48, Mdn = 100.00,\ SD = 23.72,\ p < 0.001$).
- **Participants with Dyslexia:** There was a significant effect of *Correction Type* on *Writing Accuracy* ($\chi^2(2) = 19.452,\ p < 0.001$) (Table 2).
  - *Suggestions* had the highest *Writing Accuracy* mean. Participants with dyslexia wrote more correctly text using *Suggestions* than *None* ($p < 0.001$).
  - *Detection Only* had the second highest *Writing Accuracy* mean. Participants with dyslexia wrote more correctly text using *Detection Only* than *None* ($p = 0.004$).
- **Strong Readers:** We could not find effect of *Correction Type* on *Writing Accuracy* for the strong readers group ($\chi^2(2) = 1.047,\ p = 0.596$).

Correcting Time. There was a significant effect of *Correction Type* on *Correcting Time* ($\chi^2(2) = 639.44,\ p < 0.001$). The results of the post-hoc tests show that:

- **Between Groups:** Participants with dyslexia had significantly shorter attempts to correct the sentences ($M = 12.47, Mdn = 10.23,\ SD = 12.80$) than the strong readers ($M = 12.62, Mdn = 7.96,\ SD = 16.41,\ p < 0.001$).
- **Participants with Dyslexia:** There was a significant effect of *Correction Type* on *Correcting Time* ($\chi^2(2) = 349.76,\ p < 0.001$) (Table 2).
  - *Suggestions* had the shortest correction time mean. Participants spent significantly less time using *Suggestions* than *Detection Only* ($p < 0.001$), and *None* ($p < 0.001$).
  - *Detection Only* had the second shortest correction time mean. Participants spent significantly less time using *Detection Only* than *None* ($p = 0.004$).
- **Strong Readers:** There was a significant effect of *Correction Type* on *Correcting Time* ($\chi^2(2) = 263.27,\ p < 0.001$) (Table 2).
  - Strong readers spent significantly less time correcting sentences using *Suggestions* than with *Detection Only* ($p < 0.001$) or the absence of the condition, *None* ($p = 0.002$).
  - We could not find effects between *Detection Only* and *None* conditions for the strong readers ($p = 0.722$).

Subjective Writing Accuracy. There was a significant effect of *Correction Type* on *Subjective Writing Accuracy* ($\chi^2(2) = 22.40,\ p < 0.001$). The results of the post-hoc tests show that:

- **Between Groups:** Participants with dyslexia had significantly lower subjective writing accuracy under all conditions than the stronger readers ($p = 0.046$), Table 2.
- **Participants with Dyslexia:** There was a significant effect of *Correction Type* on *Subjective Writing Accuracy* ($\chi^2(2) = 13.90,\ p = 0.001$) (Table 2).
  - Participants with dyslexia perceived that they wrote more accurate text using *Suggestions* than with the absence of the condition ($p = 0.005$).
- **Strong Readers:** There was a significant effect of *Correction Type* on *Subjective Writing Accuracy* ($\chi^2(2) = 8.86,\ p = 0.012$) (Table 2).
  - Strong readers reported that they wrote more accurate text using *Suggestions* than with the absence of the condition ($p = 0.080$).

Subjective Correcting Time. There was a significant effect of *Correction Type* on *Subjective Correcting Time* ($\chi^2(2) = 26.28,\ p < 0.001$), see Table 2. The results of the post-hoc tests show that:

- **Between Groups:** We could not find significant differences between groups on *Subjective Correcting Time* ($p = 0.073$).

Dependent Variable/Condition	*People with Dyslexia*			*Strong Readers*		
	Mdn	$M \pm SD$	%	Mdn	$M \pm SD$	%
Writing Accuracy						
None	100	78.05 ± 39.8	100	100	91.97 ± 25.79	100
Error Detection Only	100	89.83 ± 27.92	115	100	92.65 ± 25.08	101
Error Suggestions	100	93.01 ± 25	119	100	95.96 ± 19.51	104
Correcting Time						
None	10.26	11.97 ± 7.30	119	8.33	12.35 ± 14.06	111
Error Detection Only	11.93	15.44 ± 18.72	154	8.50	14.37 ± 19.73	129
Error Suggestions	8.375	10.03 ± 9.13	100	6.97	11.17 ± 14.96	100
Subjective Writing Accuracy						
None	4	3.76 ± 0.75	100	4	4.24 ± 0.75	100
Error Detection Only	4	4.24 ± 0.66	112	4	4.53 ± 0.80	107
Error Suggestions	5	4.65 ± 0.70	124	5	4.75 ± 0.58	112
Subjective Correcting Time						
None	3	3.41 ± 0.80	100	4	3.94 ± 0.83	100
Error Detection Only	4	4.24 ± 0.75	124	5	4.59 ± 0.62	116
Error Suggestions	5	4.59 ± 0.71	135	5	4.62 ± 0.81	117

Table 2: Median, mean and standard deviation of the dependent measures per condition and group. We include the relative percentage with respect to the smallest average value per condition.

- **Participants with Dyslexia:** There was a significant effect of *Correction Type* on *Subjective Correcting Time* ($\chi^2(2) = 19.62,\ p < 0.001$) (Table 2).
  - Participants with dyslexia perceived that they corrected the sentences significantly faster using the *Detection Only* and ($p = 0.020$) *Suggestions* ($p = 0.001$) than *None*.
- **Strong Readers:** There was a significant effect of *Correction Type* on *Subjective Correcting Time* ($\chi^2(2) = 8.14,\ p = 0.017$), see Table 2.
  - Strong readers reported that they spent significantly less time correcting sentences using *Suggestions* than with the absence of the condition ($p = 0.033$).

5. DISCUSSION

We discuss the different evaluation results as well as the limitations of our system based on the error analyses.

5.1 System Evaluation

Our system achieves competitive results compared with other similar approaches [14], nevertheless it is worth noting that they use an automatic generated test set for evaluation, while we use real world sentences written by people with dyslexia. This makes evaluation more challenging, since written errors by people with dyslexia are sometimes semantically similar [25], not only orthographically similar.

5.2 Comparison with Spellcheckers

The only spellchecker that provides competitive results compared to us is the one implemented under the Google Docs on-line tool that it is better in the correction of errors. However, the number of errors corrected by Google Docs is significantly smaller than in our system, since they provide a very precision oriented outcome. Nonetheless, since readers with dyslexia cannot consciously detect errors [26], we hypothesize that a more recall oriented system, like ours, would be useful for this target population. It is worth noting that our implementation outperforms Google Doc results for error detection, the number of true positives is higher in both tasks, and the number of false negatives is smaller in both tasks.

5.3 User Evaluation

Our evaluation with people with dyslexia and strong readers demonstrated that our spellchecker leads both populations to more efficient corrections. Using the suggestions proposed by our corrector, both populations wrote significantly more accurate text in less time. While no effects were found for strong readers in the error detection condition, the fact that the error candidates were highlighted was beneficial for participants with dyslexia, since the written accuracy improved significantly and the correction time decreases significantly. Readers with dyslexia perceived that they could write more accurately and in less time using our corrector.

5.4 Error Analyses and Limitations

Some real-word errors are difficult to detect because they are both grammatically and semantically correct. For example, *Real Check* does not detect the error **No ha gustado mucho tu propuesta ('Your proposal was not liked')* where the intended sentence was *Nos ha gustado mucho tu propuesta ('We do like your proposal')*. Future work may address this by considering document-level context, as it is done NLP tasks, such as anaphora resolution [21].

Another limitation comes from the coverage of Google n-grams. Some of the sentences that were written with dyslexia did not have a match in any of the n-grams at disposal. In order to overcome this issue we will try alternative approaches used in recent NLP tools, such as word vectors [19], word clusters [4], and synonym generation systems [1]. This would enhance the coverage of the method since it will allow to try with different string matching approaches, in

spite of the exact string matching approach that it is implemented in the current version.

Real Check does not detect some special kind of real-word errors that involve word boundary errors, for instance *más cara ('more expensive')* instead of *máscara ('mask')*, a system that detects this kind of errors would need to increase the processing load of the algorithm geometrically. Tokens should be checked by pairs and, for instance, the dependency parser filter would need to deal with subtrees instead of tokens. It is worth noting that none of the systems shown in Table 1 solves this problem.

6. CONCLUSIONS

The main contribution of this paper is a method that detects real-word errors in Spanish with 50.42 precision and 65.93 recall. Our *Real Check* system offers an improvement over widely-used spell checkers in both error detection and in error correction. An evaluation with 34 people shows that both people with dyslexia and strong readers correct sentences more accurately in less time with *Real Check*.

Acknowledgements

We thank the following associations that support people with dyslexia who helped us collect written errors: *Madrid con la Dislexia ('Madrid for Dyslexia')*[20], *Associació Catalana de Dislèxia ('Catalan Association of Dyslexia')*[21], and *Avecedari Associació i Glifing('Avecedari Association and Glifing')*[22].

We especially thank those who contributed significant amount of written errors, including Cristina Villanueva with Emma and Berta; Gloria Dotor Ferreira; Queti Porras with Miquel Romero Porras and Pedro Romero Porras; Luis Darriba Fernández; Montse Segarra Calvis with Gris Ventura-Gibert Segarra; Justina Pilar Velasco Ayala with Juan; Gloriana Hernanz Plaza with Ana Sánchez Hernanz; Joan Albà with Júlia Albà Coll; Úrsula M. Cózar; Ángeles Fernández Begines; Montserrat García with Mercè Gonzalez Calderón and Silvia Roca; Sergio Herrero; Fer del Real; Ruth Rozensztejn; Luciana Salerno; and Pepi Saravia.

The contents of this paper were developed under a grant from the National Science Foundation (#IIS-1149709), a Google Research Award, and a grant from the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR grant number 90DP0061-01-00). MB was supported by the European Commission under the contract numbers FP7-ICT-610411 (project MULTISENSOR) and H2020-RIA-645012 (project KRISTINA).

7. REFERENCES

[1] R. Baeza-Yates, L. Rello, and J. Dembowski. A simple context aware synonym simplification method. In *Proceedings of the NAACL-HLT 2015*. Association for Computational Linguistics, 2015.

[2] B. Bohnet and J. Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*, 2012.

[3] T. Brants and A. Franz. {Web 1T 5-gram Version 1}. 2006.

[4] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[5] L. Burnard. Reference guide for the british national corpus (world edition), 2000.

[6] J. Carifio and R. Perla. Resolving the 50-year debate around using and misusing Likert scales. *Medical education*, 42(12):1150–1152, 2008.

[7] A. J. Carlson, J. Rosen, and D. Roth. Scaling up context-sensitive text correction. pages 44–50, 2001.

[8] M. Felice, Z. Yuan, Ø. E. Andersen, H. Yannakoudakis, and E. Kochmar. Grammatical error correction using hybrid systems and type filtering. In *Proc. CoNLL: Shared Task*, pages 15–24, Baltimore, Maryland, June 2014. ACL.

[9] A. R. Golding and D. Roth. A winnow-based approach to context-sensitive spelling correction. *Machine learning*, 34(1-3):107–130, 1999.

[10] M. Gustafsson. A spell checker with a user model for Swedish dyslexics. *Projektarbeten 2003*, page 69, 2003.

[11] J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. pages 1–18, 2009.

[12] G. Hirst and A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(01):87–111, 2005.

[13] International Dyslexia Association. Frequently Asked Questions About Dyslexia, 2011. `http://www.interdys.org/`.

[14] A. Islam and D. Inkpen. Real-word spelling correction using google web it 3-grams. In *Proc. EMPLN'09*, pages 1241–1249. ACL, 2009.

[15] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, 1997.

[16] A. Q. Li, L. Sbattella, and R. Tedesco. Polispell: an adaptive spellchecker and predictor for people with dyslexia. In *User Modeling, Adaptation, and Personalization*, pages 302–309. Springer, Berlin, Heidelberg, 2013.

[17] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov. Syntactic annotations for the google books ngram corpus. In *Proc. ACL'12 (demonstrations)*, pages 169–174. Association for Computational Linguistics, 2012.

[18] E. Mays, F. J. Damerau, and R. L. Mercer. Context based spelling correction. *Information Processing & Management*, 27(5):517–522, 1991.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[20] `http://www.madridconladislexia.org/`

[21] `http://www.acd.cat/`

[22] `http://www.avesedari.com/ca/glifing/`

[21] R. Mitkov. *Anaphora resolution*. Longman, London, 2002.

[22] A. Pauls and D. Klein. Faster and smaller n-gram language models. In *Proc. ACL'11*, pages 258–267. Association for Computational Linguistics, 2011.

[23] J. Pedler. Computer spellcheckers and dyslexics - a performance survey. *British Journal of Educational Technology*, 32(1):23–37, 2001.

[24] J. Pedler. Using semantic associations for the detection of real-word spelling errors. In *Proceedings from The Corpus Linguistics Conference Series*, volume 1, 2005.

[25] J. Pedler. *Computer Correction of Real-word Spelling Errors in Dyslexic Text*. PhD thesis, Birkbeck College, London University, 2007.

[26] L. Rello. *DysWebxia. A Text Accessibility Model for People with Dyslexia*. PhD thesis, Universitat Pompeu Fabra, 2014.

[27] L. Rello, R. Baeza-Yates, and J. Llisterri. A resource of errors written in Spanish by people with dyslexia and its linguistic, phonetic and visual analysis. *Language Resources and Evaluation*, to appear.

[28] L. Rello, R. Baeza-Yates, H. Saggion, and J. Pedler. A first approach to the creation of a Spanish corpus of dyslexic texts. In *LREC Workshop NLP4ITA*, pages 22–27, Istanbul, Turkey, May 2012.

[29] K. U. Schulz and S. Mihov. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition*, 5(1):67–85, 2002.

[30] R. Spooner. *A spelling aid for dyslexic writers*. PhD thesis, University of York, 1998.

[31] C. Sterling, M. Farmer, B. Riddick, S. Morgan, and C. Matthews. Adult dyslexic writing. *Dyslexia*, 4(1):1–15, 1998.

[32] S. Verberne. Context-sensitive spell checking based on word trigram probabilities. *Unpublished master's thesis, University of Nijmegen*, 2002.

[33] A. Wilcox-O'Hearn, G. Hirst, and A. Budanitsky. Real-word spelling correction with trigrams: A reconsideration of the mays, damerau, and mercer model. In *Computational Linguistics and Intelligent Text Processing*, pages 605–616. Springer, 2008.

[34] World Health Organization. *International statistical classification of diseases, injuries and causes of death (ICD-10)*. World Health Organization, Geneva, 10th edition, 1993.