UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Ingeniería

https://revistas.udistrital.edu.co/index.php/reving/issue/view/1241

DOI: https://doi.org/10.14483/23448393.21135

Research

Advanced Neural Model for Spanish Spell-Checking

Modelo neuronal avanzado para corrección ortográfica en español

Eduard Gilberto Puerto Cuadros[1]*

[1]Universidad Francisco de Paula Santander, Cúcuta, Colombia.

Abstract

Context: Correcting spelling errors in written content, particularly in Spanish texts, remains a critical challenge in natural language processing (NLP) due to the complexity of word structures and the inefficiency of existing methods when applied to large datasets.

Method: This paper introduces a novel neural model inspired by the brain's cognitive mechanisms for recognizing and correcting misspelled words. Through a deep hierarchical framework with specialized recognition neurons and advanced activation functions, the model is designed to enhance the accuracy and scalability of spelling correction systems. Our approach not only improves error detection but also provides context-aware corrections.

Results: The results show that the model achieves an F-measure of 83 %, significantly surpassing the 73 % accuracy of traditional spell-checkers, marking a substantial advancement in automated spelling correction for the Spanish language.

Conclusions: The features of the neural model facilitate spelling correction by emulating the cognitive mechanisms of the human mind. Our model detects more orthographic error types and reports less false positives. As for its limitations, this proposal requires the supervised definition of the weights assigned to the variables used for recognition.

Keywords: neocortex, deep neural model, spell-checker, pattern recognition

Article history

Received: 12th/Aug/2023

Modified: 20th/Aug/2023

Accepted: 7th/Oct/2024

Ing, vol. 29, no. 3, 2024, e21135

*Correspondence: eduardpuerto@ufps.edu.co

Resumen

Contexto: La corrección de errores ortográficos en textos escritos, especialmente en textos en español, sigue siendo un desafío crucial en el procesamiento del lenguaje natural (PLN) debido a la complejidad de las estructuras de las palabras y la ineficacia de los métodos existentes cuando se aplican a grandes conjuntos de datos.
Método: Este artículo presenta un novedoso modelo neuronal inspirado en los mecanismos cognitivos del cerebro para reconocer y corregir palabras mal escritas. A través de un marco jerárquico profundo con neuronas de reconocimiento especializadas y funciones de activación avanzadas, el modelo está diseñado para mejorar la precisión y la escalabilidad de los sistemas de corrección ortográfica. Nuestro enfoque no solo mejora la detección de errores, sino que también proporciona correcciones conscientes del contexto.
Resultados: Los resultados muestran que el modelo alcanza una medida F del 83 %, superando significativamente el 73 % de precisión de los correctores ortográficos tradicionales, lo que representa un avance sustancial en la corrección automática de ortografía para el idioma español.
Conclusiones: Las funcionalidades del modelo neuronal computacional facilitan la corrección ortográfica al emular los mecanismos cognitivos de la mente humana. Nuestro modelo detecta más tipos de errores ortográficos y presenta menos falsos positivos. En cuanto a las limitaciones, la propuesta requiere una definición supervisada de los pesos asignados a las variables que se utilizan para el reconocimiento.

Palabras clave: corrector ortográfico, neocórtex, modelo neuronal profundo, reconocimiento de patrones

Table of contents

Page

1. **Introduction** 3
2. **Research trends and related work** 5
3. **Methodology** 7
   - 3.1. Formal definition of a neuron within the neuron module 7
   - 3.2. Activation functions within the neural network model 8
   - 3.3. Architecture of the neural model . 9
   - 3.4. Optimized computational neural model 10
4. **Experiments** 11
   - 4.1. Comparison with an n-gram language model 13
   - 4.2. A final discussion about the characteristics of the neural network model 13
5. **Conclusions and future works** 14
6. **CRediT author statement** 14

References 14

1. Introduction

Spelling correction is a fundamental task in natural language processing (NLP), with significant implications for various applications, including text composition, writing assistance, and automatic editing. In the literature, (1) suggests four classic misspelled word categories, and (2) analyze around 76K misspellings found in real-life texts produced by humans (Table I). Many errors tend to be insertions, deletions, substitutions, and transpositions of letters. (2) also found that many misspelling errors in Spanish are due to

1. Omissions, mainly of accents or of one character
2. The use of lowercase instead of uppercase at the beginning of a proper noun
3. The addition of a letter
4. The substitution of one character
5. The transposition of a letter

Table I. Common spelling errors by humans (2)

Type of error	Percentage
Insertion or addition of one character (*e.g.*, aereopuerto→aeropuerto)	4,7 %
Omission of diacritics (*e.g.*, dia→día)	51,5 %
Omission of one character (*e.g.*, mostar→mostrar)	6,8 %
Substitution of one character	4,1 %
Transposition or repetition of the same letter (*e.g.*, interpetración→interpretación, movimineto→movimiento, dirrección→dirección)	2,8 %
Cognitive errors (biene→viene)	5,9 %

Table I shows that a full 51 % of the misspellings found are omissions of a diacritic on a vowel. A notable example of this is the prevalence of word errors in millions of tweets and other massive datasets. In the context of Spanish, a language characterized by rich regional and orthographic variability, achieving accurate and efficient spelling correction poses unique challenges.

Various research efforts have focused on improving the accuracy and correctness of written content, using methods such as linguistic analysis, extraction, annotation, and correction based on dictionaries or statistical analysis. These methods have been applied to a range of tasks, including lemmatization, morphosyntactic labeling, syntactic analysis, sentiment analysis, and conceptual annotation (3–6).

On the other hand, large language models (LLMs) have revolutionized various NLP tasks, including spelling correction. Models such as GPT-4 and BERT have shown significant advances in identifying and correcting errors using deep neural networks. These models leverage large volumes of data to learn complex and contextual patterns, enhancing their ability to correct spelling errors in various languages,

including Spanish. However, despite these advancements, the current approaches still face significant challenges, particularly in adapting to different contexts and reducing errors in more complex situations within Spanish texts.

Recent studies have indicated that, although modern neural models achieve competitive results, there are still limitations in adapting to regional variations and specific contexts in Spanish. Less common spelling errors and linguistic peculiarities can reduce the effectiveness of current systems. For example, research has shown that LLMs can improve about precision and coverage if trained with more diverse and specific data (7–11).

One innovative approach to addressing these challenges is to computationally simulate the brain's process for recognizing and correcting misspelled words. This paper introduces a novel neural model designed to enhance the accuracy and efficiency of spelling correction systems for Spanish texts. The proposed model employs a deep hierarchical framework with specialized recognition neurons and formal strategies to identify and correct misspelled words. This approach not only enhances the accuracy of error detection and correction but also provides a more nuanced understanding of language processing. The new neural model for detecting misspelled words simulates the process that the human brain, specifically the neocortex, follows by reusing information. This efficient approach is based on the lexical and syntactic analysis of words in Spanish.

Like the brain's neocortex, a neural network uses multiple layers, with each layer handling progressively more complex aspects, starting from character-level analysis to higher-order syntactic structures. Additionally, specialized modules, akin to cortical columns in the brain, are employed to detect and process various spelling errors, from simple typos to more contextually inaccurate words. The model also leverages contextual cues, analyzing surrounding words and the overall context to enhance the accuracy of its corrections, thereby surpassing basic pattern recognition. Moreover, it integrates a mechanism for reusing previously acquired knowledge, analogous to the brain's ability to apply past experiences to novel situations. This adaptability allows the model to accommodate different writing styles and error patterns, making it more versatile and robust.

This new deep neural model emulates certain aspects of human brain function (12–14): memory is organized as a hierarchy of patterns, and, if only part of a pattern is perceived (through sight, hearing, or smell), it can still be recognized. The model also assumes several hypotheses regarding the structure of the biological neocortex, such as the uniformity of the basic neocortical structure, known as the *cortical column*. In addition, pattern recognition neurons are constantly interconnected.

Our model improves upon the state of the art by recognizing input patterns through a process of self-association in a hierarchy of patterns. It enables the decomposition of the pattern recognition problem into simpler patterns, allowing to analyze input patterns regardless of their level of complexity or their nature (a line, a word, a sentence, a paragraph, *etc.*). In addition, the neural model is easily parallelizable, as its calculations, defined in the theorems, are simpler and distributed across a hierarchy. Moreover, computational cost can be improved with respect to other approaches, with more efficient

use of memory due to a single abstract data structure that can be instanced by various text patterns. Finally, the model is adaptable since it can learn both the possible changes in pattern descriptors (such as their importance weights) and new neurons (components) if the atomic patterns are known, which is very useful in the context of a language for self-learning of words and idiomatic sentences.

In synthesis, the deep neural architecture is characterized by recognition neuron hierarchies, which increase the levels of complexity, *i.e.*, the pattern recognition neurons that constitute the lowest level levels (or X_{j-1}), will always be less complex than the neurons of the upper level (or X_j, for $j = 1, \ldots, m$). This is an innovative approach with respect to classical recognition models.

The primary contribution of this study lies in an innovative system for recognizing and correcting misspelled words in Spanish texts. This system

- demonstrates high levels of recursion and uniformity;
- operates on a self-associative principle within a hierarchical pattern framework;
- exhibits adaptability, given its ability to assimilate new patterns (words); and
- efficiently analyzes extensive Spanish texts containing words of varying structural complexity.

This method draws inspiration from cognitive neuroscience, particularly from the functioning of the neocortex, which plays a crucial role in complex language processing tasks. The pattern recognition theory of mind (PRTM) describes the basic algorithm of the neocortex, which is characterized by a hierarchy of patterns, uniformity in its basic structure, and continuous connectivity between its pattern recognition modules (14).

This work is organized as follows. Section 2 presents the research trends and related work, providing a comprehensive overview of the existing approaches to spelling correction systems for Spanish texts. Section 3 details the methodology employed, including the formal definition of a neuron within the neuron module and a description of the activation functions used within the neural network model. This section outlines the architecture of the neural model and its optimization, explaining how the model has been tailored for efficient performance in spelling correction. Section 4 presents the experiments conducted, describing the experimental setup, datasets, and performance evaluations for the proposed model in comparison with other existing methods. Finally, Section 5 states the conclusions and proposes directions for future work, highlighting the contributions of this study and discussing potential improvements and extensions of the model.

2. Research trends and related work

The field of NLP has witnessed remarkable advancements in recent years, particularly in the domains of spell-checking, grammatical error correction (GEC), and overall text improvement. These developments have been largely driven by the advent of sophisticated neural network architectures and the increasing availability of large-scale datasets. Transformer-based models have emerged as the dominant paradigm, demonstrating unprecedented performance across various NLP tasks. (15)

introduced GECToR, a transformer-based approach for GEC that achieves state-of-the-art results in several benchmarks. Building upon this work, in (16), the GECToR architecture was adapted for the Russian language (RuGECToR). Furthermore, (17) proposed a unified pre-training approach for monolingual and multilingual GEC. Their method leverages massive amounts of synthetic data and multi-task learning, achieving new state-of-the-art results in standard GEC benchmarks in English and extending well to other languages. (18) expanded the evaluation of GEC systems beyond essays from non-native learners by introducing CWEB, a new benchmark for GEC that comprises website texts written by English speakers of varying proficiency levels. This work highlights how the lower error density in these domains poses significant challenges for current GEC systems, demonstrating the need for models that better generalize across different topics and genres.

The field of spell-checking and GEC has therefore seen substantial advancements in recent years. (19) provided an extensive overview of the field of GEC, which addresses the automatic detection and correction of various errors in texts, including grammatical, orthographic, and semantic discrepancies. Over the past decade, significant advancements have been driven by five major shared tasks, catalyzing the evolution from rule-based methods and statistical classifiers to advanced neural machine translation systems. The integration of deep learning techniques, particularly transformer-based models, has led to significant improvements regarding accuracy and capability. Furthermore, the focus has expanded beyond simple error correction to include context-aware corrections, fluency enhancement, and multilingual support. The emphasis on efficiency and real-time performance demonstrates the field's maturation and its readiness for widespread practical application. As these technologies continue to evolve, we can expect to see more sophisticated, efficient, and widely applicable text improvement systems that not only correct errors but also enhance the overall quality and fluency of written communication across multiple languages and domains.

Some particularly interesting works related to the proposed model include STILUS (20), which distinguishes four types of errors: grammatical, orthographic, semantic, and style. In the case of orthographic revision, STILUS corrects words in three stages: the generation of alternatives to the incorrect word, the weighting of alternatives, and the arrangement of alternatives. Another system is ArText, a prototype automatic help system for writing texts in Spanish in specialized domains (21). The system has three modules: the first module handles aspects of structure, content, and phraseology; the second focuses on format and linguistic revision; and the last allows users to linguistically revise their texts. XUXEN (22) is a spell checker/corrector defined based on two morphological formalisms. It uses a highly inflected standardized language with a broad relationship between nouns and verbs as well as a lexicon that contains approximately 50 000 items across different grammatical categories.

On the other hand, (23) proposed a spelling and grammar checker algorithm for texts where mistakes are not detected through tagging and parsing, but rather through statistical analysis, comparing combinations of two words used in the text against a corpus of one hundred million words. In (24), the JHU FLuency-Extended GUG corpus (JFLEG) aimed at developing and evaluating GEC was presented. It represents a broad range of language proficiency levels and uses holistic fluency edits to not only correct grammatical errors but also to make the original text sound more natural.

Additionally, (3) presented a general approach to various NLP applications, such as translation and recognition, using modern techniques like deep learning. Finally, to date, there are no works based on a hierarchical approach to pattern recognition (in our case, words) as a fundamental mechanism for reusing text patterns, which is an efficient way to recognize many words, some of which may be complex. The next sections detail our proposal.

3. Methodology

This section presents the mathematical formalization of the neural model, including the neuron or recognition module, its activation functions, and its recursive architecture, as well as the computational algorithm that integrates the entire model.

3.1. Formal definition of a neuron within the neuron module

A pattern recognition module (or neuron) is formally defined as a 3-tuple. The $\Gamma\rho$ notation is used to represent the module that recognizes the ρ pattern (ρ: shapes, letters, words, *etc.*). $\Gamma\rho =< E, U, S_o >$, where E is an array composed of the 2-tuple E =<S, C> (Table I), S =<Signal, State>is an array that represents the set of signals that make up the pattern recognized by Γ and their corresponding states, and C is an array that encodes information about the pattern, as defined by the 3-tuple C =<D, V, W>, where D represents the descriptors of Γ, V is the domain vector for each D (*i.e.*, the possible values of each descriptor), and W is the importance weight of the descriptor for the ρ pattern. Additionally, U denotes the thresholds vector used by the module (Γ) to recognize its respective pattern.

Table II shows one artificial neuron, *i.e.*, a neocortical pattern recognition module according to the PRTM theory.

Table II. Neuron: matrix E = <S, C>

E				
S		**C**		
Signal	**State**	**Descriptor (D)**	**Domain (V)**	**Weight (W)**
1	False	Descriptor1	<possible values of the descriptor >	[0,1]
2	False	Descriptor2	<possible values of the descriptor >	[0,1]
3	False	Descriptor3	<possible values of the descriptor >	[0,1]
...	...	...	...	...
n	False	Descriptor	<possible values of the descriptor >	[0,1]
U: <ΔU_1, ΔU_2>				

In the neural model, each neuron/module or pattern recognition module can recognize and observe every aspect of the input pattern $s()$, as well as the way in which the different parts of the data in the input pattern may or may not relate to each other.

There are two types of thresholds: ΔU_1 is the threshold for recognition by key signals, and ΔU_2 is the threshold for recognition by partial or total mapping. ΔU_1 should be stricter than ΔU_2, given that the process of recognition by key signals utilizes only a few signals. Finally, each module produces a recognition or petition signal (S_o) towards lower levels. As petition, S_o becomes the input signal $s()$ for the pattern-matching neurons of the lower levels. When there is a recognition signal, it is diffused to its higher attainable levels, in order to modify the state of the signal to "trueïn the patterns of said levels.

Thus, a pattern is represented as a set of lower-level sub-patterns that conform to it (n descriptors), and it also serves as the sub-pattern of a higher-level pattern. The value of n depends on the pattern to be recognized (the descriptors of the pattern). The values of W are normalized [0,1], and δU_1 or ΔU_2 are thresholds that must be overcome to recognize the pattern. These values are defined in a supervised manner, according to the domain of application. In the context of text analysis, the main patterns to be recognized (ρ) are letters, words, special signs, and numbers.

3.2. Activation functions within the neural network model

Our neural network model uses two strategies for the checking/correction/recognition process, one based on key signals and the other based on partial signals. Both use a threshold of satisfaction and the importance of the signal weights. Thus, the recursive model allows decomposing the pattern recognition problem into simpler patterns, which makes it possible to analyze very complex words.

Particularly, the first strategy defined to recognize and correct text patterns using the aforementioned structures is called *Activation function 1 by key signals*, and the other is the *Activation function 2 by partial pattern matching* (25). The first uses the importance weights of the input signals identified as key, and the second uses the partial or total presence of the signals. A signal is key when it represents information that allows quickly recognizing a pattern. For example, the final letter rïn infinitive verbs could be taken as a key.

Key signal. A s_i signal in the Γ module is key if its importance weight has a value greater than or equal to the average weight of all the signals in Γ.

$$\forall s_i \in S(\Gamma), \text{ if } [w(s_i) \geq w_{average} S(\Gamma)] \rightarrow Key_\Gamma \tag{1}$$

Activation function by key signals. A ρ pattern is recognized by key signals if the average of the weights of the key signals recognized exceeds the ΔU_1 threshold. This type of recognition uses the descriptors (signals or sub-patterns) with greater weight of importance.

$$\frac{\sum_{i=1 \cap \text{ state}(s_i=\text{true}) \cap s_i \in Key_\Gamma}^{n} w(s_i)}{|Key_\Gamma|} \geq \Delta U1 \rightarrow S_0 \tag{2}$$

Activation function by partial mapping. This strategy consists of validating whether a signal minimum's number in Γ exceeds the δU_2 threshold.

$$\frac{\sum_{i=1 \cap \text{ state}(s_i=\text{true})}^{n} w(s_i)}{n} \geq \Delta U2 \rightarrow S_0 \tag{3}$$

This process of calculation is carried out for each module and each level of recognition X_i (from X_1 until X_m).

3.3. Architecture of the neural model

This section describes the neural network model instanced for the specific case of text analysis. Particularly, the hierarchical system in Fig. 1 represents the recursive and iterative process for the recognition and correction of words. Each layer in the hierarchy is an interpretation space X_i from $i = 1$ to m. X_1 is the level of recognition for atomic patterns (*e.g.*, letters or letterforms), and X_m is the level of recognition for complex patterns (*e.g.*, words and compound words). Each level is composed of γ_{ji} recognition modules (for $j = 1, 2, 3 \ldots$ neurons at level i). X_{ij} is the pattern recognized by module j at level i. The function of each recognition module is to recognize its corresponding pattern.

Fig. 1 shows the architecture of the neural network (*i.e.*, the hierarchical pattern recognition system). The multiple hidden layers are the recognition spaces of level i or the levels of complex pattern recognition (X_i). This is how the neural model can find extremely complex patterns via bottom-up or top-down approaches.

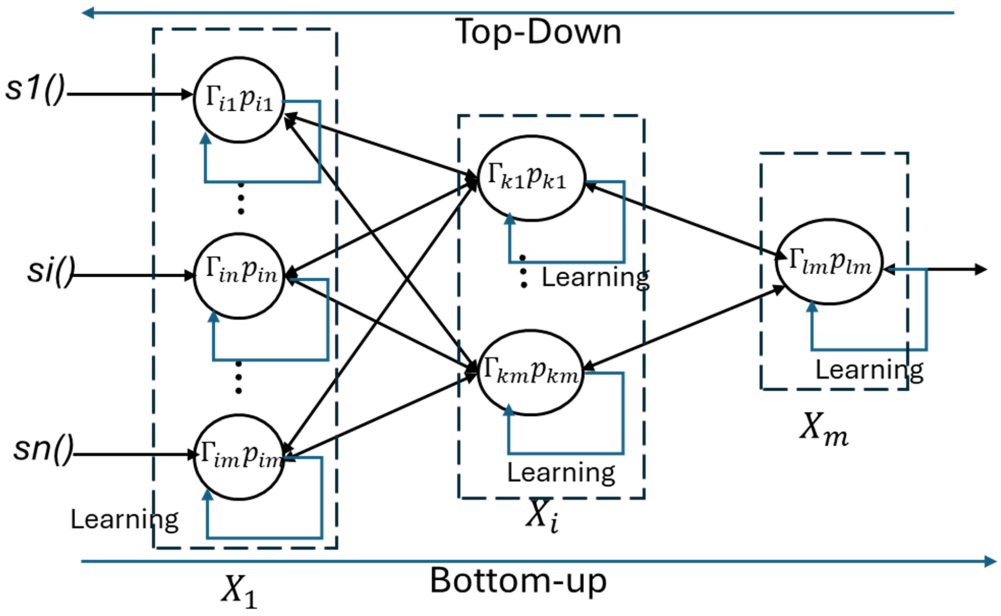

Figure 1. Recursive architecture of the neural model

Here, X_i is the recognition space of the 1-level, Γ_{i1}: is the j-neuron of the 1-level, which recognizes p_{i1}, *i.e.*, the pattern recognized by the neuron j at level 1.

This architecture is marked by hierarchies of recognition neurons that increase in complexity. Lower-level neurons are less complex than those at higher levels, which makes this approach innovative compared to traditional recognition models (26, 27).

Two analytical processes characterize the algorithm. The first process, called *top-down*, recognizes the input pattern through decomposition (Fig. 1). The top-level module invokes the recognition neurons

of lower-level constituents, and these recursively perform the same function. The second process, called *bottom-up*, is used for recognizing atomic patterns. Here, the output signal of the recognized pattern is sent to the neurons that make up the top-level representation. These top-level neurons will be activated (or not) depending on whether they exceed a recognition threshold.

3.4. Optimized computational neural model

The algorithm works as follows. It receives the input text (y = s(): sentences, words (s)) (line 2). Then, it decomposes this input into its sub-patterns, which are stored in L (lines 4-5). On line 7, it determines the level of the hierarchy where the recognition of the input pattern (y) should start and end. Afterwards, L recognition requests for (y) are created (line 12). The algorithm enters a loop that iterates over each pattern in L (line 13). If the recognition is successful at the current level (line 14), the information is updated using the *updateSubpatterns* function (line 18). If the input pattern is composed of lower-level signals or has recognized atomic signals, a bottom-up process is conducted; otherwise, a top-down procedure is followed (line 16). If the recognition fails at the lowest level (level == 1) and no pattern has been recognized, the algorithm ends (lines 22-24). This process continues until all patterns in L have been processed or their recognition has failed at the lowest level. Finally, the function returns the result of the hierarchical recognition process (line 28).

Algorithm for the implementation of the recursive neural recognition model

```
1   function hierarchicalRecognition(y)
2       // y is the input pattern
3
4       // Step 1: Decompose y into hierarchical patterns
5       subPatterns = decomposePattern(y)
6
7       // Step 2: Determine the highest-level N of the hierarchy to start recognition
8
9       N = determineInitialLevel(subPatterns)
10      // Steps 3-5: Hierarchical recognition process
11      for level from N to 1 do
12          L = createPatternList(subPatterns, level)
13          for each pattern in L do
14              if recognizePattern(pattern) then
15                  if level == 1 then
16                      return success(pattern)
17                  else
18                      updateSubPatterns(subPatterns, pattern, level)
19                      break // Move to the next lower level
20                  end if
21              end if
22          end for
23
24          if level == 1 and no pattern was recognized then
25              return failure()
26          end if
27      end for
28  end function
```

Figure 2. Algorithm for the implementation of the recursive neural recognition model

4. Experiments

This section presents the results and an evaluation of the neural model in comparison with other works, for which three additional systems were selected: STILUS, SpanishChecker, and Microsoft word. These three systems were chosen because they are tools designed to analyze spelling and find basic grammar and style mistakes in Spanish texts. All of them show errors automatically. For these test scenarios, several paragraphs were artificially made, with misspelled words in Spanish. In these tests, the inputs, *i.e.*, the paragraphs, were introduced into the neural model as an array of words, while, in the other systems, they were entered as a plain text file. The standard precision (P) and coverage (C) metrics, as well as the F-measure (F) (28), were used during these experiments to compare the performance of the evaluated systems.

Table III. Results of applying different systems to various types of spelling errors

	Methods	Detected errors	False negatives	False positives	P	C	F
Added Letters	Neural Model	12	0	0	100 %	100 %	100 %
	SpanishChecker	15	0	3	83 %	100 %	90 %
	STILUS	14	0	2	88 %	100 %	93 %
	Microsoft Word	13	0	2	92 %	100 %	95 %
Omitted letters	Neural Model	10	0	0	100 %	100 %	100 %
	SpanishChecker	14	0	4	77 %	100 %	87 %
	STILUS	10	0	0	100 %	100 %	100 %
	Microsoft Word	10	0	0	100 %	100 %	100 %
Cognitive errors	Neural Model	14	0	0	100 %	100 %	100 %
	SpanishChecker	16	2	5	76 %	88 %	81 %
	STILUS	15	0	2	88 %	100 %	93 %
	Microsoft Word	14	1	0	100 %	92 %	95 %
Exchange of two letters	Neural Model	8	0	3	72 %	100 %	83 %
	SpanishChecker	11	0	6	64 %	100 %	78 %
	STILUS	13	0	8	61 %	100 %	75 %
	Microsoft Word	14	0	9	60 %	100 %	75 %
Digits or special characters	Neural Model	5	0	0	100 %	100 %	100 %
	SpanishChecker	1	4	0	100 %	20 %	33 %
	STILUS	1	3	1	50 %	25 %	33 %
	Microsoft Word	2	3	0	100 %	40 %	57 %
Full text in digital versions	Neural Model	23	0	9	71,5 %	100 %	83 %
	SpanishChecker	56	3	45	57,9 %	95 %	72 %
	STILUS	54	1	41	58,1 %	98 %	73 %
	Microsoft Word	51	2	39	58,6 %	96 %	73 %

The first row of Table III presents the results of applying the neural model to a paragraph containing 205 words, out of which more than 5 % contained errors due to added letters. The neural

model effectively detects all real errors caused by the addition of letters, as in 'entrdada', 'tellevisión', and 'teléfonoo'. In contrast, SpanishChecker generates three false positives corresponding to the names 'Pantoja', 'Goya', and 'Chabelita'. STILUS produces similar results, with two false positives ('Pantoja' and 'Goya'), while Microsoft Word also flags 'Chabelita' and 'Goya' as errors. The neural model recognizes 'Chabelita', 'Goya', and 'Pantoja' as well-written words.

The second row corresponds to a paragraph containing 203 words, with 5 % of the words exhibiting omitted letters. According to these results, the neural model effectively detects all errors related to omitted letters, just like Microsoft Word and STILUS. In contrast, SpanishChecker generates four false positives corresponding to the words 'expresidente', 'Aznar', 'populismo', and 'patrióticamente' (with the latter being two false positives) while omitting 'Congres' (resulting in one false negative).

The third row uses a paragraph that contains 203 words, with 20 % of the words exhibiting cognitive errors. The neural model detects 14 errors, and Microsoft Word identifies a false negative: 'Confexionar'. Although SpanishChecker identifies the errors, many of its correction recommendations are syntactically and semantically far from the correct word. For example, for the word 'Confexionar', SpanishChecker suggests: {conexionar, confinar, confinara, confinare, confinará, confinaré, confinaría, confina}; for 'vioseguridad', it suggests: {vio seguridad, vio-seguridad, seguridad, esguardad, resguardad, seguridades, *etc*.}; and, for 'varato', it suggests: {va rato, va-rato, verato, grato, arto, parto, urato, verte, varío, barato, *etc*.}. Among its false positives are 'Cofeccionistas' {confusionistas, confesionistas, confusionista, cancionista, *etc*.} and 'Cucuteños' {cicutinas, cacereños, cicateros, etc.}. The two false negatives are 'Crusada' and 'Biernes'. Finally, STILUS generates the false positives 'Oquendo' and 'Cucuteña'.

The fourth row corresponds a paragraph that contains 5 % of words with two exchanged letters. In this case, all systems incorrectly detect more errors. The neural model incorrectly detects only three errors, while the other systems detect significantly more.

In the fifth row, the results obtained by the different systems are shown for a paragraph with 5 % of words containing errors related to digits or special characters. In this case, only the neural model correctly detects the errors. The other systems either fail to detect the errors or incorrectly identify more (as is the case with STILUS).

Finally, the last row of the dataset used in the experiment consists of 32 texts from the digital version of El País, dated May 17^{th}, 2001, which were copied into a Word document. Approximately 9000 words were analyzed, with 14 spelling errors. SpanishChecker identifies the fewest errors (11), while Microsoft Word and STILUS are more effective, detecting 13 and 12 errors, respectively. Our approach detects all errors but incorrectly identifies nine additional ones. The other approaches incorrectly detect many more. The main issue with these systems is that they flag errors arising from the inclusion of digits or special characters, which is not applicable in this case, resulting in many false positives. Overall, their results are significantly lower than those achieved by our system.

4.1. Comparison with an n-gram language model

As a point of reference, we considered the standard n-gram language model algorithm. This spell-checking and auto-correction system uses the Web to infer misspellings, incorporating a term list, an error model, a language model (LM), and a confidence classifier algorithm. For each token in the input text, candidate suggestions are drawn from the term list and scored using the error model. These candidates are evaluated in context using the LM and then re-ranked. A classifier is used for each token to determine the confidence level regarding whether a word has been misspelled, and, if so, whether it should be autocorrected to the best-scoring suggestion available.

The main contribution of this work is that it does not require any manually annotated resources, inferring its linguistic knowledge entirely from the Web. In this sense, we propose a deep neural architecture based on both supervised and unsupervised mechanisms for the discovery and selection of features in classification problems.

Our approach consists of three phases. The first phase, called *feature analysis*, is supported by two feature-engineering approaches to discover or select atomic features/descriptors. This phase should include many well-spelled words as well as a substantial number of non- words or misspellings, which is equivalent to the term list. The second phase, called *aggregation*, creates a feature hierarchy by merging descriptors from the atomic features. Then, the last phase classifies the inputs. This phase employs a supervised learning approach, while the earlier phases combine both supervised and unsupervised learning methods (29). If a word is misspelled, the system utilizes clusters of well-written words for recommendations; if no suitable suggestion exists, it searches the Web (for example, in repositories such as WordReference) and recommends a suitable alternative, including it in the group of well-written words. Future work may involve a quantitative evaluation of the performance of these approaches.

4.2. A final discussion about the characteristics of the neural network model

Our neural network model is highly recursive and uniform. It recognizes input patterns through a process of self-association within a hierarchy of patterns, and it allows decomposing the pattern recognition problem into simpler components, enabling the analysis of input patterns regardless of their complexity or nature (*e.g.*, a line, a word, a sentence, a paragraph, *etc.*).

Moreover, our recognition model is adaptable; it can learn both the potential changes in the pattern descriptors (including their importance weights) and new neurons (patterns) if the atomic patterns are known. This adaptability is particularly useful for the self-learning of words and idiomatic expressions in a language context. Unlike other approaches, the neural model can recognize words with special characters in a manner similar to how the brain does (*e.g.*, = a, E = 3, S = 5, 9 = q), allowing for interpretations like m@ma, p3ra, *etc*.

Additionally, its novelty lies in the way it addresses this problem. While several NLP models have achieved unprecedented performance levels, they often come at high computational costs.

5. Conclusions and future works

The proposed model aims to advance the state of the art in spelling correction for Spanish – and potentially other languages – by offering a solution informed by biological processes and optimized for practical application. This innovative approach could lead to significant improvements in automated text correction systems, benefiting users across various linguistic contexts. The neural model detects a wider range of orthographic errors and generates fewer false positives, although it requires a supervised definition of the weights assigned to the variables used for recognition.

This work presents a new neural model for detecting misspelled words, closely mirroring how the human brain (specifically the neocortex) addresses this problem. The model reuses information to propose an efficient approach based on the lexical analysis of word syntax in Spanish.

This system demonstrates high levels of recursion and uniformity, operating on a self- associative principle within a hierarchical pattern framework. It showcases adaptability by assimilating new patterns (words) and efficiently analyzes extensive Spanish texts containing words of varying structural complexity.

A comparative evaluation indicated that the precision and coverage of the neural model are competitive with those of other spell-checkers. In the experiments, the system outperformed three other tools; it achieved an F-measure of 83 %, surpassing the other spell-checkers' 73 %. Our neural model can detect a broader variety of orthographic errors while producing fewer false positives. As future work, the architecture of the neural model should be extended to incorporate unsupervised learning mechanisms, allowing it to improve its performance by learning new words. Additionally, it must be adapted to handle other languages. The model could also be developed to simultaneously correct texts written in both English and Spanish, which would be valuable in translation tasks. To this effect, the model needs to be expanded with more recognition neurons in different languages, enhancing its lexical basis.

Finally, a comparison with other approaches in the domain of NLP is not presented herein, as our primary focus is on the spell-checking (auto-correction) problem. While there are several NLP approaches for machine translation, cognitive dialogue systems, sentiment analysis, text classification, and text summarization – utilizing techniques from natural language understanding and generation present in state-of-the-art NLP – a future research direction could involve analyzing the application of our approach in these contexts for comparison with existing techniques. By leveraging the capabilities of LLMs like GPT-3.5, researchers can further refine and extend these systems, leading to more accurate, efficient, and sophisticated solutions for spelling correction in Spanish and beyond.

6. CRediT author statement

Puerto Cuadros, E.G: contributed to conceptualization, data collection, formal analysis, and writing – original draft preparation and editing.

References

[1] S. Almurashi, "Analysis of the most common spelling errors in English for Saudi students: A case study of foundation year students," *Getsempena English Edu. J.*, vol. 10, no. 1, pp. 73-89, 2023. https://doi.org/10.46244/geej.v10i1.2081 ↑3

[2] F. Bustamante and E. Díaz, "Spelling error pattern in Spanish for word processing applications," in *Proc. 5th Int. Conf. Lang. Res. Eval.*, 2006, pp. 93-98. http://www.lrec-conf.org/proceedings/lrec2006/pdf/119_pdf.pdf ↑3

[3] S. Singh and A. Mahmood, "The NLP cookbook: Modern recipes for transformer based deep learning architectures," *IEEE Access*, vol. 9, pp. 68675-68702, 2021. https://doi.org/10.1109/ACCESS.2021.3077350 ↑3, 7

[4] A. Ferreira and S. Hernández. "Diseño e implementación de un corrector ortográfico dinámico para el sistema tutorial inteligente", *Rev. Signos*, vol. 50, no. 95, pp. 385-407, 2017. http://dx.doi.org/10.4067/S0718-09342017000300385 ↑3

[5] A. San Mateo, "Un corpus de bigramas utilizado como corrector ortográfico y gramatical destinado a hablantes nativos de español," *Rev. Signos*, vol. 49, no. 90, pp. 94-118, 2016. http://dx.doi.org/10.4067/S0718-09342016000100005 ↑3

[6] P. Gamallo and M. Garcia, "LinguaKit: A multilingual tool for linguistic analysis and information extraction," *Linguamatica*, vol. 9, no. 1, pp.19-28, 2017. ↑3

[7] G. Zomer and A. Frankenberg-Garcia, "Beyond grammatical error correction: Improving L1-influenced research writing in English using pre-trained encoder-decoder models," in *Find. Assoc. Comp. Ling. EMNLP 2021* , 2021, pp. 2534-2540. https://doi.org/10.18653/v1/2021.findings-emnlp.216 ↑4

[8] B. Ünlütabak and O. Bal, "Theory of mind performance of large language models: A comparative analysis of Turkish and English," *Comp. Speech Lang.*, vol. 89, art. 101698, 2025. https://doi.org/10.1016/j.csl.2024.101698 ↑4

[9] M. Bijoy *et al.* "A transformer-based spelling error correction framework for Bangla and resource scarce Indic languages," Comp. Speech https://doi.org/10.1016/j.csl.2024.101703 ↑4

[10] E. Puerto, J. Aguilar, R. Vargas, and J. Reyes, "An Ar2p deep learning architecture for the discovery and the selection of features," *Neural Process. Letters*, vol. 50, no. 1, pp. 623-643, 2019. https://doi.org/10.1007/s11063-019-10062-4 ↑4

[11] E. Puerto, and J. Aguilar and A. Pinto, "Automatic spell-checking system for Spanish based on the Ar2p neural network model," *Computers*, vol. 13, no. 13, art. 76, 2024. https://doi.org/10.3390/computers13030076 ↑4, 12

[12] E. Puerto and B. R. Pérez, "Análisis de la teoría de la mente humana basada en el reconocimiento de patrones," 2014. [Online]. Available: http://hdl.handle.net/20.500.12749/12358 ↑4

[13] E. Puerto Cuadros, "Avances en el conocimiento y modelado computacional del cerebro autista: Una revisión de literatura," *Cuad. Activa*, vol. 9, no. 2017, pp. 109-125, 2017. https://doi.org/10.53995/20278101.425 ↑4

[14] R. Kurzweil, "How to make mind," *Futurist*, vol. 47, no. 2, pp. 14-17, 2013. ↑4, 5

[15] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, "GECToR – Grammatical error correction: Tag, not rewrite," in *15th Work. Innov. Use NLP Build. Edu. App.*, 2020, pp. 163- 170. https://doi.org/10.48550/arXiv.2005.12592 ↑5

[16] I. A. Khabutdinov, A. V. Chashchin, A. V. Grabovoy, A. S. Kildyakov, and U. V. Chekhovich, "RuGECToR: Rule-based neural network model for Russian language grammatical error correction," *Programm. Comp. Software*, vol. 50, no. 4, pp. 315-321, 2024. https://doi.org/10.1134/S0361768824700129 ↑6

[17] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, "A simple recipe for multilingual grammatical error correction," in *ACL-IJCNLP 2021*, 2021, pp. 702-707. https://doi.org/10.18653/v1/2021.acl-short.89 ↑6

[18] S. Flachs, O. Lacroix, H. Yannakoudakis, M. Rei, and A. Søgaard, "Grammatical error correction in low error density domains: A new benchmark and analyses," in *2020 Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 8467-8478. https://doi.org/10.48550/arXiv.2010.07574 ↑6

[19] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical error correction: A survey of the state of the art," *Comp. Ling.*, vol. 49, no. 3, pp. 643-701. https://doi.org/10.1162/coli_a_00478 ↑6

[20] V. González, B. González, and M. Muriel, "STILUS: sistema de revisión lingüística de textos en castellano," *Proc. Leng. Nat.*, vol. 29, pp. 305-306, 2002. ↑6

[21] I. da Cunha, M. Montané, and L. Hysa, "The arText prototype: An automatic system for writing specialized texts," in *Proc. Euro. Chapter Assoc. Comp. Ling.*, 2017, pp. 57-60. https://aclanthology.org/E17-3015 ↑6

[22] E. Agirre et al., "XUXEN: A spelling checker/corrector for Basque based on two-level morphology," in *3rd Conf. Applied Natural lang. Process.*, 1992, pp. 119-125. ↑6

[23] A. Valdehíta, "Un corpus de bigramas utilizado como corrector ortográfico y gramatical destinado a hablantes nativos de español," *Rev. Signos*, vol. 49, pp. 94-118, 2016. ↑6

[24] C. Napoles, K. Sakaguchi, and J. Tetreault, "A fluency corpus and benchmark for grammatical error correction", in *Proc. Euro. Chapter Assoc. Comp. Ling.*, 2017, pp. 229-234. https://doi.org/10.48550/arXiv.1702.04066 ↑6

[25] E. Puerto and J. Aguilar, "Formal description of a pattern for a recursive process of recognition," in *Proc. IEEE Latin American Conf. Comp. Intell.*, 2016, pp. 1-2. https://doi.org/10.1109/LA-CCI.2016.7885746 ↑8

[26] E. Puerto, J. Aguilar, and D. Chávez, "A new recursive patterns matching model inspired in systematic theory of human mind," *Int. J. Advance. Comp. Tech. (IJACT)*, vol. 28, no. 9, 2017. ↑9

[27] E. Puerto, J. Aguilar, R. Vargas, and J. Reyes, "An Ar2p deep learning architecture for the discovery and the selection of features," *Neural Process. Letters*, vol. 50, no. 1, pp. 623-643, 2019. https://doi.org/10.1007/s11063-019-10062-4 ↑9

[28] D. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation," *J. Machine Learn. Tech.*, vol. 2, pp. 37-63, 2011. ↑11

[29] E. Puerto and J. Aguilar. "Learning algorithm for the recursive pattern recognition model," *App. Artif. Intell.*, vol. 30, no. 7, pp. 662-678, 2016. https://doi.org/10.1080/08839514.2016.1213584 ↑13

Eduard Gilberto Puerto Cuadros

Systems engineer, MSc in Computer Science, and PhD in Applied Science and Engineering. Junior Researcher at MinCiencias, with experience as Director of the Artificial Intelligence Research Group (GIA), recognized by MinCiencias (https://gia.ufps.edu.co/index/), at Universidad Francisco de Paula Santander (UFPS); and as Director of the Eduardo Cote Lamus Library Division. Full professor at the Department of Systems and Informatics of UFPS, and professor of the Master's program in ICTs Applied to Education. He has supervised multiple Undergraduate and Master's projects and has served as co-researcher in various research groups, *i.e.*, the International Internship of the Department of Computer Science at the University of Miami, the Institute for Research in Applied Mathematics and Systems (IIMAS) in Mexico DF, the Center for Microcomputing and Distributed Systems (CEMISID) at Universidad de los Andes (Mérida, Venezuela), the Robotics and Intelligent Systems Laboratory of Escuela Politécnica Nacional in Quito (Ecuador), and the Computational Sciences Research Group (CICOM) of Universidad de Pamplona (Colombia). He has actively participated in several scientific events and authored numerous articles and book chapters on his areas of interest, which include artificial intelligence, machine learning, deep learning, logic programming, theory of computation, and complex systems.
Email: eduardpuerto@ufps.edu.co

Eduard Gilberto Puerto Cuadros

Advanced Neural Model for Spanish Spell-Checking
Modelo neuronal avanzado para corrección ortográfica en español

Ingeniería
vol. 29, no. 3, e21135, 2024
Universidad Distrital Francisco José de Caldas,
ISSN: 0121-750X
ISSN-E: 2344-8393

DOI: https://doi.org/10.14483/23448393.21135