Evaluating Data Augmentation Strategies for Training Spanish Misspelling Detection Models

Manuel Castillo-Sancho[a], **Jordi Porta-Zamorano**[b], **Asunción Gómez-Pérez**[c]
[a,b]Centro de Estudios de la Real Academia Española, [c]Real Academia Española
{manuel.castillo, porta, agp}@rae.es

Abstract

This paper evaluates three data augmentation strategies for training misspelling detection models in Spanish. Using the Spanish CORRSIC corpus of naturally occurring misspellings, we compare three misspelling generation methods: random perturbations, keyboard-based errors, and a statistical model derived from empirical edit patterns encoded as weighted finite-state transducers. We also analyze two word selection strategies (random and length-based) and two augmentation configurations designed to balance data diversity and reduce spurious correlations. This study shows that the statistical model produces misspellings most similar to real data, showing the lowest Jensen–Shannon divergence (0.148 nats) with the empirical distribution. In downstream detection experiments, performance improves with training size, and differences between word selection strategies remain minimal. Overall, the results highlight the value of statistically grounded misspelling generation for realistic and effective data augmentation in spell-checking tasks in Spanish.

Keywords: misspelling detection, data augmentation, spelling correction, error generation, Spanish NLP

1. Introduction

Spell checking remains a fundamental component of natural language processing (NLP) systems, supporting a wide range of applications from text editing tools to large-scale information retrieval and language learning platforms. Despite its long history, the task continues to present challenges, particularly in adapting to diverse domains and handling the variability of real-world misspellings. A central difficulty is the scarcity of annotated corpora that capture authentic spelling errors with sufficient coverage to train modern discriminative models.

Data augmentation offers a promising avenue to address this limitation. By generating synthetic misspellings that approximate naturally occurring ones, it is possible to expand training data, improve model robustness, and reduce overfitting. However, the effectiveness of different augmentation strategies for spell checking has not been systematically assessed. Existing approaches often rely on ad hoc misspelling generation heuristics, such as random substitutions or keyboard-based noise, without a clear understanding of their impact on downstream misspelling detection performance.

This paper assesses the role of data augmentation in spell checking by evaluating multiple misspelling generation strategies in Spanish under controlled conditions. We compare random misspelling injection, keyboard-proximity perturbations, and statistical models designed to approximate empirical misspelling distributions observed in real-world corpora. Our experiments focus on misspelling detection with discriminative models, analyzing how different augmentation strategies and sampling methods influence precision, recall, and overall performance.

2. Related Work

Recent research has explored various approaches to modeling and generating spelling errors for data augmentation in spelling correction and detection tasks. The survey on grammatical error correction by Bryant et al. (2023) includes a section with references devoted to the generation of synthetic data to produce grammatical errors in text and classifies the methods as noise injection, back-translation, and round-trip translation. A particularly relevant subsequent contribution is that of Martynov et al. (2023), who investigate augmentation methods for emulating human misspellings in Russian. They introduce a multi-domain corpus of genuine spelling errors and compare two complementary generation strategies: Augmentex, a heuristic method based on random and keyboard-level perturbations, and Statistic-based Spelling Corruption (SSC), which derives probabilistic error distributions from aligned correct-incorrect word pairs. Their evaluation, using distributional similarity measures such as the Kolmogorov-Smirnov test, demonstrates that the statistical model produces synthetic errors that more closely resemble real ones, particularly in substitution-dominated domains.

Our work follows a similar line of inquiry but focuses on Spanish and emphasizes empirical evaluation of data augmentation strategies for training misspelling detection models rather than correction. Using the CORRSIC corpus of naturally occurring Spanish misspellings as reference, we compare random, keyboard-based, and statistical generation methods, along with alternative word selection

Proceedings of the Third Workshop on Computation and Written Language (CAWL 2026) @LREC 2026, pages 71–78
12 May, 2026.

and augmentation configurations. These studies converge on a key insight: statistically grounded models that reproduce empirical edit distributions generate more realistic and effective training data than purely heuristic approaches. However, unlike Martynov et al. (2023), our experiments directly quantify the downstream impact of each augmentation strategy on discriminative detection performance, thereby bridging the gap between distributional realism and task-specific utility. A more comprehensive account of the methodology, results, and implications of this work is presented in (Castillo-Sancho, 2025), a Master's thesis developed within the framework of the LEIA project[1].

3. Description of CORRSIC

CORRSIC is a manually annotated Spanish corpus of natural misspellings derived from the *Corpus de Referencia del Español Actual* (CREA)[2], carefully annotated by qualified linguists to ensure high-quality labeling. Compiled several decades ago, it consists of text segments, typically full sentences, containing misspellings together with their corresponding corrections. The spelling of the corpus has been updated to reflect the latest reform of Spanish orthography. The size of the CORRSIC corpus is summarized in Table 1.

Element	Count
Text Segments	49676
Words	1696344
Annotated Misspellings	49958
Misspelling Density	2.95%
Misspelling Types	30008

Table 1: Summary of the CORRSIC corpus.

In the annotation scheme used in CORRSIC misspellings are marked with the `<sic>` element and the attribute `@corr` providing the corrected form. An illustrative example is shown below:

```
<sic corr="Qué">Que</sic>
bueno contar con una página
de información como la suya.
A mi hijo le interesa mucho
este deporte.  Él aún es pe-
queño y me gustaría saber
<sic corr="como">cómo</sic>
lo puedo encauzar para que
no lo tome temporalmente nada
<sic corr="más">mas</sic>,
sino que le ayude en su de-
sarrollo personal más <sic
corr="adelante">adlante</sic>.
```

where *que*/*qué*, *como*/*cómo* and *mas*/*más* are pairs of words distinguished by a diacritic accent, i.e., a special use of the accent mark that distinguishes words that are spelled the same but have different meanings or grammatical functions, and *adlante* is a misspelling of *adelante*.

A 60–40% split of this corpus was used to evaluate different misspelling detection models under natural conditions. The 60% portion will be used to obtain statistics for the statistical model, while the remaining 40% will serve as a test set.

4. Misspelling Modeling

To assess whether synthetic misspelling generation strategies for data augmentation influence the performance of the misspelling detection model, and to what extent, three models were developed in the LEIA project. Specifically, we compare: (i) a model that generates misspellings at random, (ii) a model that introduces misspellings based on keyboard proximity, and (iii) a model that produces misspellings following a distribution similar to that observed in CORRSIC.

4.1. Random and Keyboard Models

4.1.1. Random Model

This model generates misspellings through random perturbations, applying a fixed number of modifications to the characters of a word. The possible operations include the insertion of a random character, the substitution of one character with another, the deletion of an existing character, and the transposition of two consecutive characters.

4.1.2. Keyboard Model

This model simulates misspellings derived from the spatial organization of the QWERTY keyboard. Although some implementations exist for simulating keyboard errors in English (Ma, 2019), they do not fully support a Spanish keyboard layout; for this reason, we developed our own method.

The starting point is a complete representation of the Spanish layout, divided into the base layer, the *Shift* layer, the *AltGr* layer, and the extended layers with accented and diacritic characters. Each printable character is assigned a position in this map, which enables the definition of its neighbors in two dimensions.

Candidate substitutions for a given character are determined by computing their topological distance on the keyboard grid. The substitution probability

[1] *Lengua Española e Inteligencia Artificial* (Spanish Language and Artificial Intelligence): https://www.rae.es/leia-lengua-espanola-e-inteligencia-artificial

[2] https://www.rae.es/banco-de-datos/crea

decreases as this distance increases, following an exponential decay function. As a result, characters located in close proximity to the original key are the most likely candidates for replacement.

In addition to this distance-based weighting, the model integrates factors associated with the nature of the keyboard layout. Substitutions within the same layer are assigned the highest weight, while transitions across related layers (e.g., base to *Shift*) receive reduced weight, and substitutions involving more distant layers (e.g., base to *AltGr* or diacritic variants) are penalized more strongly. The final probability of substitution is obtained by combining the spatial and layer-based factors, followed by normalization of the resulting distribution. A replacement character is then sampled according to this distribution.

Additionally, the model incorporates a small probability of introducing diacritic errors in vowels. Due to the complexity and variability of diacritic input mechanisms, these substitutions are not constrained by the keyboard topology and are instead applied independently of spatial considerations.

Finally, it is important to note that this model is limited to character substitutions. It does not simulate insertion or deletion errors, focusing exclusively on replacement phenomena derived from typing interaction.

4.2. Statistical Model

The statistical model seeks to approximate the empirical distribution of misspellings observed in the CORRSIC test partition. To achieve this, misspelling patterns are first extracted from the CORRSIC training partition and then used to construct a weighted finite-state transducer (WFST) (Mohri, 2009) capable of generating misspellings in an input string with a distribution similar to that observed in CORRSIC.

4.2.1. Misspelling Patterns Extraction

The extraction of misspelling patterns starts from aligned pairs of correct-incorrect words in the CORRSIC train partition. Each pair is processed using the Damerau-Levenshtein algorithm to find the minimal edit script (Damerau, 1964; Levenshtein, 1966), i.e., the sequence of atomic operations (insertion, deletion, substitution) that transforms the correct form into the erroneous one. For example, the pair (*amigo* / *amgo*) yields the single operation $i \rightarrow \epsilon$, while (*soleado* / *ssokeado*) results in the insertion $\epsilon \rightarrow s$ plus the substitution $l \rightarrow k$.

Three refinements are introduced to capture misspelling patterns more contextually and realistically. First, explicit boundary symbols are added to model edits occurring at the beginning or end of words. Second, consecutive edits are merged when they co-occur. Third, the representation is extended from unigrams to bigrams whenever an insertion or deletion takes place.

4.2.2. Misspelling Transducer Construction

Every edit operation is stored in the form of a tuple (α, β), where α is a string to be replaced in the input, and β is its replacement in the output string. After extracting the operations for each word pair in the corpus, they are aggregated over the entire dataset to obtain their empirical frequencies.

The probabilistic model of spelling errors is then derived directly from these extracted patterns. Each edit operation $\alpha \rightarrow \beta$, together with its empirical probability, is encoded as a transition in a WFST. In this framework, states represent alignment positions, and transitions correspond to possible edits (substitutions, insertions, deletions, or identity mappings when no error occurs).

The WFST is defined over the tropical semiring, where the weight of a path is calculated as the sum of the weights of its transitions. Each transition in the transducer corresponds to a possible edit operation and is weighted by the negative logarithm of its empirical probability.

As a result, a complete misspelling sequence transforming a word into a misspelled variant corresponds to a path through the transducer. The path weight is the sum of the individual edit costs, which in the tropical semiring framework is equivalent to the negative logarithm of the product of their probabilities.

By applying the inverse of the logarithmic transformation, the weight of a path can be mapped back into a probability value.

This formulation allows us to take advantage of the shortest-path algorithms available for WFSTs. Given an input word w, it is composed with the WFST, and the operation $\mathrm{ShortestPath}(w \circ \mathrm{WFST}, k)$ returns the k lowest-cost paths, i.e., the k most probable misspellings (Mohri, 2002). Because each path weight can be mapped back into a probability through the inverse log transformation, we can not only rank candidates but also sample from them according to their true probabilities. In practice, we set $k = 1000$ to balance computational time and coverage.

4.2.3. Misspelling Model Evaluation

When comparing discrete probability distributions, such as the misspelling distributions generated by the misspelling models and the empirical distribution extracted from the CORRSIC test partition, a common choice is the Kullback–Leibler (KL) divergence (D_{KL}) (Kullback and Leibler, 1951), as it directly quantifies the information loss incurred when one distribution is used to approximate another. It

is defined as follows:

$$D_{\mathrm{KL}}(Q \,||\, P) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

However, D_{KL} presents well-known limitations: it is asymmetric ($D_{\mathrm{KL}}(P \,\|\, Q) \neq D_{\mathrm{KL}}(Q \,\|\, P)$), which complicates its interpretation as a measure of similarity, and it becomes infinite whenever the supports of the compared distributions, those produced by the misspelling model and those observed in the CORRSIC test partition, do not fully overlap. To address these issues, we adopt the Jensen–Shannon (JS) divergence (Lin, 1991), a symmetrized and smoothed variant of KL that is always finite and thus better suited for comparing empirical discrete distributions. JS divergence is defined as follows:

$$D_{\mathrm{JS}}(P \,||\, Q) = \tfrac{1}{2} D_{\mathrm{KL}}(P \,||\, M) + \tfrac{1}{2} D_{\mathrm{KL}}(Q \,||\, M)$$

where $M = \frac{1}{2}(P + Q)$.

The JS divergence was applied to compare the edit distributions of words misspelled by different models. For reference, the JS divergence between the CORRSIC train and test partitions is 0.006 nats. We used the same misspelled words from the corpus test partition and applied the other noise-injection methods to them. From Table 2, we observe that the statistical model (Statistic) yields the lowest JS divergence with respect to the empirical CORRSIC test partition distribution (0.148 nats), indicating that it produces misspelling patterns most similar to those found in real data. By contrast, the keyboard-based misspelling model shows the highest divergence (0.548 nats), while the random model lies in between (0.374 nats). This higher divergence in the keyboard model can be attributed, in part, to its restriction to character substitutions only, as it does not simulate insertion or deletion errors, which are present in the real data.

P	$D_{\mathrm{JS}}(P, \mathrm{CORRSIC}_{\mathrm{Test}})$
Random	0.374
Keyboard	0.548
Statistic	0.148

Table 2: Jensen-Shannon divergence between the distributions of editions in words produced by different misspelling models and the empirical distribution from CORRSIC test partition. Units are given in nats.[1]

The ability of the statistical error model to emulate other misspelling distributions was also evaluated. The distribution of keyboard-induced errors is accurately reproduced by the WFST model, with a Jensen–Shannon divergence of 0.041 nats. In contrast, the distribution of random errors is not captured as precisely, showing divergences of 0.245 nats with k=1000 and 0.222 nats with k=5000. This difference can be explained by the greater variability of the randomly generated misspellings, which makes it more difficult to model a stable statistical pattern.

4.3. Length-Based Word Selection

As previous studies have shown for English (Özbey et al., 2022; Flor et al., 2015), word length and word frequency strongly influence the production of misspellings. When words are randomly sampled from running text, the resulting word-length distribution becomes heavily biased toward shorter words, in accordance with Zipf's law, since high-frequency words tend to be shorter. As illustrated in Figure 1, this distribution differs markedly from that observed in the Spanish CORRSIC corpus. The same figure also presents the word-length distribution of words sampled according to this empirical distribution. These two word selection strategies for error injection, the Random and Length-based approaches, are used in the experiments for training the misspelling detection models.

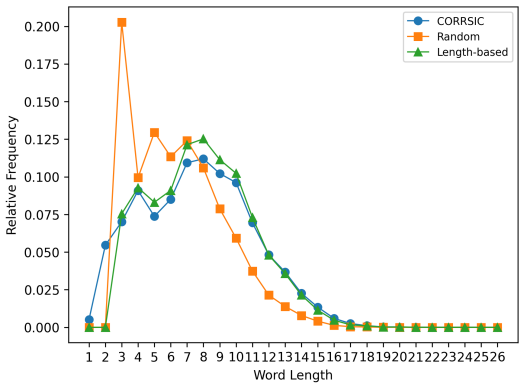

Figure 1: Comparison of word-length distributions for randomly sampled words, words from CORRSIC, and length-based samples replicating CORRSIC's distribution.

5. Experiments on Misspelling Detection

Misspelling detection is framed as a token classification problem, implemented through a discriminative model that predicts a misspelling/non-misspelling label for each token in the input sequence. We use `roberta-base-bne`[3], a monolingual Spanish variant of RoBERTa and the BIO

[1] Nats are the unit of information when logarithms are taken in base e (the natural logarithm).

[3] https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne

(*Beginning–Inside–Outside*) scheme (Ramshaw and Marcus, 1995), where tokens at the beginning of a misspelling span are tagged as `B-ERR`, tokens inside the span as `I-ERR`, and all other tokens as `O`.

Table 3 reports the results of the misspelling detection task on CORRSIC, comparing different word selection strategies for noise induction, misspelling models, and alternative sampling methods for constructing the training set for fine-tuning the model. The word selection strategies are based on random sampling (RND) and on word length (LEN), designed as described in 4.3 to approximate the length distribution observed in CORRSIC. Regarding the selection of training examples, two approaches were considered: one that augments each noisy example by generating multiple misspelling combinations at varying density levels (from 25% downward) (AUG1), and another that performs random sampling over the augmented set generated by AUG1 (AUG2). AUG1 includes fewer base sentences but with a larger variety of misspelling combinations, whereas AUG2 contains a greater number of distinct base sentences but fewer combinations. This design aims to mitigate potential spurious correlations (Ye et al., 2025; Liu et al., 2024) arising from the model's memorization of specific errors or contextual patterns.

With respect to the size of the training data, we observe a general improvement in misspelling detection metrics on the CORRSIC test partition as the training set increases, suggesting that model performance scales with data availability. The full dataset contains 10 million sentences, and experiments with fewer samples were performed on subsets derived from this complete corpus.

In spell-checking tasks, however, precision is typically favored over recall, and performance is therefore often reported in terms of $F_{0.5}$ rather than F_1. As shown in Table 3, the statistical model (c) achieves higher F and recall scores than the random (a) and keyboard-based (b) models, indicating a stronger ability to capture real misspelling patterns. Regarding data-augmentation configurations, AUG2 yields higher precision but lower recall than AUG1, consistent across all models. This can be explained by the fact that AUG2 includes a wider range of distinct sentences but fewer misspelling variations per sentence, allowing the model to observe more diverse contexts and become more precise in identifying actual errors. However, the limited number of variations per context reduces its exposure to certain error structures, leading to lower recall. In contrast, AUG1 exposes the model to fewer unique contexts but with a larger set of misspelling combinations, enabling it to better generalize across diverse error patterns and achieve higher recall, albeit at the cost of precision. The mixed model (d) reaches an F score comparable to that of the statistical model, yet with slightly higher precision and somewhat lower recall, suggesting that combining generation strategies can balance complementary strengths while maintaining overall performance.

The results indicate only minimal differences between random and length-based word selection strategies in Spanish. Since the reported scores are aggregate values, no valid p-value can be computed from them alone. However, the descriptive comparison suggests that the choice of word selection method has little impact on downstream performance when using discriminative models, which calls into question the benefit of more complex selection heuristics. A possible explanation is that the models operate at the level of the tokens defined by the underlying model's tokenizer rather than whole words, which may reduce the impact of word-level selection strategies. Supporting this, the JS divergence between the token distributions of datasets with different word selection strategies is 0.00787 nats, indicating that their token distributions are nearly identical.

To evaluate how accurately a detection model trained on a dataset with misspellings produced by different generation methods can identify the source of each error, we computed the corresponding confusion matrix (Figure 4). The figure shows these results, revealing a certain degree of distinguishability among the different methods. In general, the detection model successfully recognizes most errors generated by their respective methods, with accuracies of approximately 77%, 85%, and 79% for the Random, Keyboard, and Statistical models, respectively. Confusions tend to occur mainly between the Random and Statistical categories, which present more overlapping noise patterns. By contrast, errors generated through keyboard-based perturbations are more easily identifiable, suggesting that this type of noise exhibits more distinctive and consistent features.

6. Limitations

This study presents several limitations that should be acknowledged. First, the evaluation is confined to the datasets and experimental conditions considered here, which may limit the external validity of the findings across domains and error distributions. Second, because the analysis is based on aggregate performance scores, it does not permit formal statistical significance testing, thereby constraining the strength of comparative claims. Third, the study does not differentiate between competence errors and performance errors, despite the fact that these error types may respond to distinct linguistic, cognitive, and contextual factors. Fourth, the keyboard-

Size	Word Selector	AUG1				AUG2			
		$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$	$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$
500k	RND	0.73	0.64	0.68	0.71	0.86	0.58	0.69	0.78
	LEN	0.71	0.67	0.69	0.70	0.83	0.61	0.70	0.77
1M	RND	0.77	0.62	0.68	0.73	0.84	0.60	0.70	0.78
	LEN	0.73	0.66	0.69	0.71	0.82	0.62	0.70	0.77
5M	RND	0.74	0.67	0.71	0.72	0.80	0.65	0.72	0.76
	LEN	0.75	0.68	0.71	0.73	0.77	0.66	0.71	0.75
10M	RND	0.78	0.68	0.72	0.76	0.76	0.66	0.71	0.74
	LEN	0.78	0.68	0.73	0.76	0.76	0.68	0.72	0.74

(a) Random Misspelling Model

Size	Word Selector	AUG1				AUG2			
		$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$	$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$
500k	RND	0.74	0.62	0.67	0.71	0.84	0.55	0.66	0.76
	LEN	0.76	0.60	0.67	0.72	0.85	0.56	0.67	0.77
1M	RND	0.73	0.63	0.68	0.71	0.82	0.58	0.68	0.76
	LEN	0.74	0.62	0.68	0.71	0.83	0.60	0.69	0.77
5M	RND	0.78	0.63	0.70	0.74	0.80	0.63	0.70	0.76
	LEN	0.79	0.63	0.70	0.75	0.76	0.65	0.70	0.74
10M	RND	0.78	0.63	0.70	0.74	0.75	0.67	0.71	0.73
	LEN	0.77	0.68	0.72	0.75	0.76	0.67	0.71	0.74

(b) Keyboard Misspelling Model

Size	Word Selector	AUG1				AUG2			
		$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$	$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$
500k	RND	0.67	0.79	0.73	0.69	0.83	0.74	0.78	0.81
	LEN	0.69	0.79	0.74	0.71	0.83	0.76	0.79	0.81
1M	RND	0.72	0.78	0.75	0.73	0.82	0.77	0.79	0.81
	LEN	0.72	0.80	0.76	0.73	0.83	0.77	0.80	0.82
5M	RND	0.75	0.81	0.78	0.76	0.79	0.81	0.80	0.79
	LEN	0.75	0.84	0.79	0.77	0.78	0.83	0.80	0.79
10M	RND	0.72	0.84	0.78	0.74	0.75	0.84	0.79	0.77
	LEN	0.75	0.85	0.80	0.77	0.76	0.84	0.80	0.77

(c) Statistical Misspelling Model

Size	Word Selector	AUG1				AUG2			
		$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$	$\mathbf{P}$	$\mathbf{R}$	$\mathbf{F_1}$	$\mathbf{F_{0.5}}$
500k	RND	0.75	0.69	0.72	0.74	0.84	0.69	0.75	0.81
	LEN	0.72	0.73	0.73	0.72	0.85	0.70	0.77	0.82
1M	RND	0.77	0.71	0.74	0.76	0.84	0.72	0.77	0.81
	LEN	0.77	0.72	0.75	0.76	0.85	0.72	0.78	0.82
5M	RND	0.79	0.75	0.77	0.78	0.83	0.72	0.77	0.81
	LEN	0.82	0.74	0.78	0.80	0.83	0.76	0.79	0.81
10M	RND	0.81	0.76	0.79	0.80	0.83	0.75	0.79	0.81
	LEN	0.83	0.75	0.79	0.81	0.84	0.73	0.78	0.82

(d) Mixture of Random, Keyboard and Statistical Misspelling Models

Table 3: Results of the downstream misspelling detection task on the CORRSIC test partition under different training set sizes, word selection strategies, and data augmentation methods for different misspelling generation.

based misspelling model is restricted to a single input-device setting and therefore does not capture the variability associated with other keyboards, mobile devices, or alternative input modalities. Finally, the corpus employed is not fully contemporary, which may reduce its representativeness with respect to present-day Spanish usage; diachronic change and distributional shift may affect both the prevalence and the form of misspellings, and thus limit the generalizability of the results to present-

Real/Pred	Random	Keyboard	Statistical
Random	13178 (**0.77**)	2250 (0.13)	1639 (0.10)
Keyboard	1064 (0.06)	14806 (**0.85**)	1490 (0.09)
Statistical	2093 (0.12)	1568 (0.09)	13603 (**0.79**)

Table 4: Confusion matrix for the identification of the source model that generated each misspelling. Values are reported as absolute counts, with percentages in parentheses.

day text. The results reported here remain valid for the relative comparison of models, even under the limitations described.

7. Conclusions

This work has presented a systematic evaluation, under the conditions and limitations described above, of data augmentation strategies for training misspelling detection models in Spanish. Using the CORRSIC corpus as a reference for natural errors, we compared three misspelling generation approaches—random, keyboard-based, and statistical—and analyzed their impact on downstream detection performance.

Our results show that the statistical model, which reproduces empirical misspelling distributions through a weighted finite-state transducer, most closely approximates real misspelling patterns, as confirmed by the lowest Jensen–Shannon divergence values. Although the keyboard and random models also improve robustness, their generated noise deviates more from the empirical distribution.

In the downstream detection task, performance increases consistently with training data size, while differences between random and length-based word selection remain marginal. Moreover, the two augmentation configurations (AUG1 and AUG2) exhibit complementary behaviors in terms of precision and recall, highlighting the importance of balancing data diversity and contextual variability.

Overall, the findings suggest that statistically grounded error generation constitutes an effective and realistic strategy for data augmentation in Spanish spell-checking systems. Future work will extend these experiments to multilingual settings, explore hybrid generative–statistical approaches, and assess their integration in large-scale language models for orthographic error correction.

8. Acknowledgements

The authors thank the reviewers for their valuable comments and suggestions, which contributed to improving the quality of the manuscript and to clarifying its scope and limitations. This publication results from work undertaken within the framework of the LEIA (*Lengua Española e Inteligencia Artificial*) project, promoted by the Spanish Ministry of Economic Affairs and Digital Transformation and by the Recovery, Transformation, and Resilience Plan, funded by the European Union - NextGenerationEU.

9. Bibliographical References

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701.

Manuel Castillo-Sancho. 2025. Modelado de erratas para el entrenamiento de un verificador ortográfico en español. Master's thesis, Universidad Politécnica de Madrid, Madrid, Spain.

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

Michael Flor, Yoko Futagi, Melissa Lopez, and Matthew Mulholland. 2015. Patterns of misspellings in L2 and L1 English: a view from the ETS Spelling Corpus. *Bergen Language and Linguistics Studies*, 6.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Jiahua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Jiawei Liu, Min Huang, and Qinghai Miao. 2024. Mitigating spurious correlations in named entity recognition models through counterfactual data augmentation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Nikita Martynov, Mark Baushenko, Alexander Abramov, and Alena Fenogenova. 2023. Augmentation methods for spelling corruptions. In *Proceedings of the International Conference "Dialogue 2023"*, volume 2023.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.*, 7(3):321–350.

Mehryar Mohri. 2009. *Weighted Automata Algorithms*, pages 213–254. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text Chunking Using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*, pages 82–94.

Wenqian Ye, Luyang Jiang, Eric Xie, Guangtao Zheng, Yunsheng Ma, Xu Cao, Dongliang Guo, Daiqing Qi, Zeyu He, Yijun Tian, Megan Coffee, Zhe Zeng, Sheng Li, Ting-hao, Huang, Ziran Wang, James M. Rehg, Henry Kautz, and Aidong Zhang. 2025. The clever hans mirage: A comprehensive survey on spurious correlations in machine learning.

Can Özbey, Hatice Altınok, and Mustafa Umut Demirezen. 2022. A Novel Probabilistic Framework for Modeling Spelling Errors by Term Length and Frequency. In *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, pages 1–6.