Overview of Spelling Correction algorithms: from simple to deep learning

Rasul Alakbarli, Karim Rochd, Ahmed Nazar

13/02/2025

Abstract

In this report, we present a comprehensive exploration of spelling correction methodologies, tracing their evolution from classical approaches to modern deep learning techniques. Our research is motivated by an interest in the development history of spelling correction algorithms and their practical applications. We began with foundational methods, implementing the Levenshtein distance and the Wagner-Fischer algorithm. These simple models were evaluated using basic datasets, including the Birkbeck and Wikipedia corpora. Building on this groundwork, we plan to investigate statistical models, such as n-grams, to better understand probability-based corrections, though dataset selection for this phase is still in progress. Finally, we aim to advance to deep learning approaches, including T5 and BERT models, to explore context-aware spelling corrections, with future work focusing on dataset identification and model training. This article provides a structured overview of our methodologies and insights gained from each phase, contributing to a deeper understanding of spelling correction techniques from traditional algorithms to modern neural models.

1 Introduction

Spelling correction is a fundamental task in natural language processing (NLP) with applications in search engines, text editors, and automated systems. The evolution of spelling correction methods spans from simple rule-based techniques to advanced deep learning models.

Our research explores this progression, starting with classical approaches such as the Levenshtein distance and the Wagner-Fischer algorithm. We further improved the Wagner-Fischer method by incorporating keyboard distance weighting to handle typographical errors more effectively. For these models, we used simple datasets, including Birkbeck and Wikipedia.

Next, we plan to investigate statistical models, such as n-grams, to understand how probabilistic methods predict and correct errors based on language patterns. Dataset selection for this phase is currently in progress.

Finally, we aim to explore deep learning models, focusing on transformers like T5 and BERT, which leverage context for more accurate corrections. Although

we have not yet selected datasets for this phase, it will be a key area of our future work.

This article aims to provide a comprehensive overview of spelling correction methodologies, highlighting our implementations, findings, and planned directions. By examining the evolution from simple algorithms to advanced deep learning techniques, we aim to contribute to a deeper understanding of spelling correction systems and their practical applications.

2 Classical approaches

Classical approaches to spelling correction rely on rule-based and distance-based algorithms to identify and correct errors. These methods are simple, efficient, and effective for isolated word corrections. In our research, we explored three key algorithms: **Levenshtein distance**, **Wagner-Fischer algorithm**, and an **improved Wagner-Fischer algorithm** with additional features.

2.1 Levenshtein Distance

The Levenshtein distance is a fundamental string similarity measure that calculates the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one word into another. It provides a basic but effective method for detecting and correcting spelling errors, especially for typographical mistakes. For example, the distance between `"hte"` and `"the"` is 1 (one substitution). We used this method as a baseline due to its simplicity and efficiency and it gave us close to 100% accuracy.

2.2 Wagner-Fischer Algorithm

The Wagner-Fischer algorithm is an optimized dynamic programming implementation of the Levenshtein distance. It constructs a matrix to compute edit distances between words efficiently, enabling faster corrections for longer inputs. This algorithm provided a robust improvement in performance for correcting multiple words or larger corpora. The accuracy was as high as previously while the execution time was better.

2.3 Improved Wagner-Fischer Algorithm

To enhance the accuracy of corrections, we developed an improved version of the Wagner-Fischer algorithm by integrating multiple scoring factors:

- **Keyboard Distance Weighting:** We assigned different costs to character substitutions based on their proximity on a QWERTY keyboard, improving corrections for typographical errors (e.g., `"grape"` vs. `"frape"`).

- **Context-Aware Scoring:** We introduced penalties based on neighboring word probabilities, allowing corrections to be influenced by sentence context.
- **Phonetic Similarity (Metaphone):** By incorporating the Metaphone algorithm, we prioritized corrections with similar phonetic structures, enhancing the handling of homophone errors (e.g., `"nite"` → `"night"`).
- **Word Frequency Consideration:** To improve accuracy, we weighted corrections based on word frequency from our datasets, favoring common words over rare ones (e.g., `"hte"` → `"the"` rather than `"hue"`).

2.4 Datasets and Evaluation

For evaluating these classical approaches, we used the **Birkbeck spelling error corpus** and a subset of **Wikipedia articles**, both containing real-world spelling mistakes. These datasets provided a suitable benchmark for assessing error detection rates, correction accuracy, and computational efficiency.

2.5 Outcomes

All methods showed great accuracy when comparing predicted words with their correct versions as the task was very basic. Our observations showed that improved Wagner-Fischer algorithm was the best compared to other two. However, it remained limited in addressing errors requiring broader sentence context.

In the next phase of our research, we aim to overcome these limitations by exploring **statistical models**, such as **n-grams**, which incorporate language modeling to further improve context-aware corrections.

3 Statistical models

References