Automatic Spelling Correction

Rasul Alakbarli[1]
Karim Rochd[1]

[1]M1 AI

Décembre 2024

Contents

What is Spelling Correction?

- Identifying and correcting misspelled words.
- Applications: search engines, text editors, chatbots.
- Two key tasks:
  - **Error Detection**: Identifying incorrect words.
  - **Error Correction**: Finding the most likely correct word.

Historical Development

- **1950s-1980s: Rule-Based Systems**
  - Dictionary-based lookups.
  - Edit distance (Levenshtein, 1965).
- **1990s-2000s: Statistical Methods**
  - Noisy Channel Model (Bayesian inference).
  - N-gram models for contextual spelling correction.
- **2010s-Present: AI-Based Approaches**
  - Neural networks (RNNs, LSTMs).
  - Transformer-based models (BERT, T5).

Basic Approach: Rule-Based

- Uses predefined rules and dictionaries.
- Detects misspelled words that are not in the dictionary.
- Uses **edit distance** to find the closest correct word.
- Example:
  - Input: `hte`
  - Correction: `the` (Edit distance $= 1$)

Dataset: Webster's Dictionary, WordNet, Aspell Dictionary.
Model: Levenshtein Distance, Wagner–Fischer algorithm

Wagner-Fischer Implementation

- **Key Features:**
    - Computational efficiency

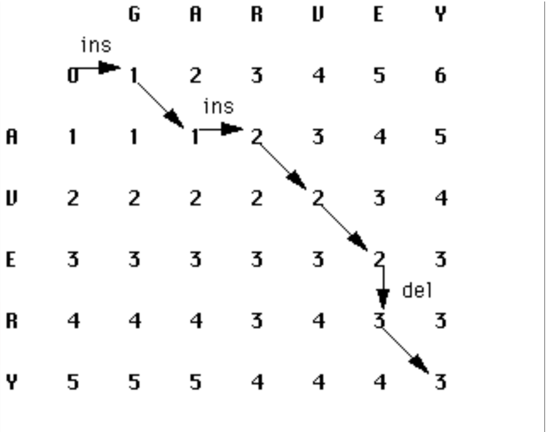
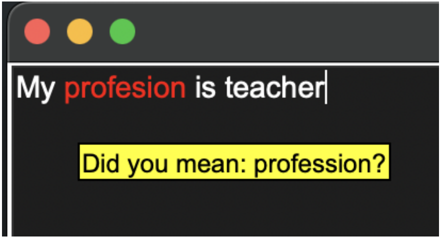
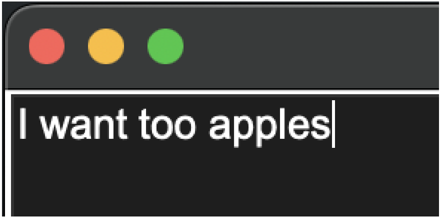


Figure 1. d(i,j) Matrix with Minimal Path Identified

- **Downsides:**
  - Lack of context awareness
  - No tolerance for new words





Enhanced Wagner-Fischer Implementation

- **Key Features:**
  - Keyboard distance weighting
  - Context-aware scoring
  - Phonetic similarity (Metaphone)
  - Word frequency consideration

Enhanced Wagner-Fischer Implementation

- **Scoring Formula:**

$$Score = 0.4 \cdot \frac{1}{1 + d_{edit}} + 0.3 \cdot s_{phonetic} + 0.2 \cdot f_{word} + 0.1 \cdot s_{context} \quad (1)$$

Where:

- d_{edit}: Enhanced edit distance
- $s_{phonetic}$: Metaphone similarity
- f_{word}: Word frequency
- $s_{context}$: Context score

Norvig's Statistical Approach

- **Key Components:**
  - Word frequency dictionary from training corpus
  - Edit distance generation (1 and 2 edits)
  - Probability-based candidate selection
- **Correction Algorithm:**
  1. Generate all possible edits:
     - Deletions: `hello` → `helo`
     - Transpositions: `hello` → `hlelo`
     - Replacements: `hello` → `hallo`
     - Insertions: `hello` → `helloo`
  2. Select most frequent candidate:

$$correction = \arg\max_{c \in candidates} P(c) \tag{2}$$

- **Advantages:**
  - Simple yet effective implementation
  - No complex language models needed
  - Training requires only text corpus

Advanced Statistical Implementation

- **Features:**
  - N-gram language model for context
  - Metaphone phonetic matching
  - Log-scaled frequency weighting
  - Confidence threshold ($score > 1.0$)
- **Implementation Details:**
  - Uses NLTK for n-gram modeling
  - Phonetic similarity via Metaphone algorithm
  - Contextual analysis with 2-word window
  - Preserves original capitalization
- **Scoring Formula:**

$$Score = 2.0 \cdot E_1 + 1.0 \cdot E_2 + 0.5 \cdot \log(f + 1) + 1.5 \cdot P + 0.3 \cdot C \quad (3)$$

  Where:
  - E_1: Edit distance 1 match
  - E_2: Edit distance 2 match
  - f: Word frequency
  - P: Phonetic match
  - C: Context score

AI-Based Approach: Deep Learning

- Context-aware spelling correction.
- Models:
  - **Seq2Seq** (LSTMs).
  - **Transformers** (BERT, T5).
- Example:
  - Input: `"I am going too the store"`
  - Output: `"I am going to the store"`
- Used in Google Search, Grammarly, and ChatGPT.

Dataset: TypoCorpus, Common Crawl Dataset, Lang-8 Learner Corpus.
Model: BERT, T5.

Evaluation Metric: BLEU Score

- Measures similarity between the corrected text and the reference text.
- Formula:

$$BLEU = \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4}$$

Comparison of Approaches

Method	Accuracy	Speed	Context Awareness
Rule-Based	Medium	Fast	Low
Statistical	Medium	Medium	Medium
AI-Based	Very High	Slow	High

Conclusion

- Spelling correction evolved from rule-based to AI-based methods.
- Statistical and ML approaches improve accuracy.
- AI-based models can handle complex, context-dependent errors.