KEC Conference 2026

ISSN: 3021-940X (Print)

DOI: 10.3126/injet.v3i2.95781

Spell Correction using N-Gram Modeling and Zero Shot Learning

Nikita Subba[1*], Bikal Devkota[1], Shuvra Baral[1]

[1] *Department of Computer and Electronics, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, nikitasubba@kec.edu.np*

[1] *Department of Computer and Electronics, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, bikaldevkota@kec.edu.np*

[1] *Department of Computer and Electronics, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, shuvrabaral@kec.edu.np*

Abstract

This paper presents an integrated spell-correction system that combines n-gram language modeling with zero-shot contextual inference using a pretrained LLaMA-2 7B model. The n-gram component efficiently generates correction candidates from local word-sequence statistics, while the zero-shot stage re-ranks those candidates by evaluating contextual and semantic plausibility without task-specific fine-tuning. The system is evaluated on two established benchmarks—BEA-60K and JFLEG—and compared against Hunspell, pyspellchecker, a standalone n-gram baseline, a standalone LLaMA-2 baseline, and the NeuSpell (BERT) toolkit. On BEA-60K, the integrated model achieves an F_1 -score of 90.7%, improving over the n-gram-only baseline (59.6%) and the standalone LLaMA-2 model (80.9%). On JFLEG, the system obtains a GLEU score of 58.6, outperforming all individual baselines. An error analysis shows that the integrated model handles non-word errors with 94.1% accuracy and real-word context-sensitive errors with 85.9% accuracy. These results demonstrate that hybrid statistical–neural architectures can deliver strong correction performance while preserving the efficiency of the n-gram front end.

Keywords: Natural Language Processing, N-Gram Model, Zero-Shot Contextual Inference, Spell Correction, LLaMA

1. Introduction

The necessity for precise and effective spell correction systems has increased due to the quick spread of digital communication. Spelling errors have the potential to compromise textual clarity and impede efficient communication, especially in automated systems like text editors, search engines, and natural language processing (NLP) applications. The intricacies of modern language usage are too complex for traditional spell correction techniques, which frequently rely on dictionary searches and rule-based algorithms. This is especially true in situations when context is crucial. This paper explores the combination of zero-shot contextual inference with n-gram modeling in order to overcome these restrictions. Classical noisy-channel and context-sensitive spell-correction work has established the importance of ranking corrections with both lexical and contextual evidence [1]–[3].

N-gram models offer a statistical basis for predicting possible adjustments depending on the context. They do this by examining the frequency and patterns of consecutive word or character sequences. This probabilistic approach enhances the system's ability to identify likely corrections by using contextual dependencies within the text. In addition to the n-gram models, the zero-shot contextual inference stage offers a powerful method for capturing semantic relationships between words. By prompting a pretrained large language model (LLaMA-2 7B) with candidate corrections and their sentence context, the system can evaluate contextual appropriateness without task-specific fine-tuning. This capability is crucial for refining correction suggestions to ensure they are contextually appropriate [2].

1.1 Background

Written communication has grown in popularity in the current digital era across a variety of platforms, including social media, emails, and instant messaging. But the quick speed of communication frequently results in typos, such as misspellings, which can obstruct clear communication and have unexpected consequences. In order to correct these mistakes and guarantee the correctness and clarity of written information, spell checkers are essential. Moreover, static dictionary-based methods have difficulties due to the dynamic nature of language, which sees new words emerge and old ones change in meaning and usage

* *Corresponding author*

over time. Large volumes of text data can be used to train machine learning algorithms, which allows them to recognize intricate language patterns. N-gram models may estimate a word's chances of occurring given its context by examining word sequences, which can be very helpful for spell correction. This study aims to contribute to the development of more effective and efficient spell correction systems that satisfy the changing needs of users in today's linked world by utilizing the capabilities of n-gram modeling and zero-shot contextual inference.

1.2 Motivation

The idea for a spell correction approach that uses n-gram models with zero-shot contextual inference was inspired by the realization that while conventional spell correction procedures have their limitations, cutting-edge computational techniques have the ability to overcome these obstacles. Conventional spell checkers frequently work in isolation, only offering corrections based on the spelling of certain words without taking the larger context into account. This system can better grasp the context in which a word appears and provide more accurate suggestions by utilizing n-gram models and a pretrained language model that evaluates unlabeled word sequences. Spell checking techniques based on static dictionaries find it difficult to keep up with the constant introduction of new words. By developing a spell correction system powered by n-gram models, this system aims to enhance the accuracy and user experience by providing more relevant and contextually appropriate correction suggestions in real-time. Recent advancements in machine learning and natural language processing have opened up new possibilities for improving spell correction accuracy.

1.3 Problem Statement

The major problem of the basic spell checker is about the spell detection stage. It is designed in the assumption that all the word errors are the words that are not in the dictionary. These are classified as non-word spelling errors. However, there are cases where the spelling error is not simply a "spelling error"; consider the following example:

"I would like a peace of cake as desert."

By simply looking at the words in the sentence above, all of them are fine in terms of spelling. However, errors still occur as the words "peace" and "desert" are not suitable for the context. They are called real-word spelling errors. In a spell checker that uses dictionary check, this kind of error will go undetected and proceed. It is clear that dictionary check is not an optimal spelling detection method. To tackle these challenges, this paper integrates n-gram modeling and zero-shot contextual inference into a unified spell correction framework. N-gram models are used to analyze the statistical properties of word sequences, providing initial correction candidates based on the likelihood of word occurrences. Zero-shot contextual inference using LLaMA-2 7B, which captures the semantic relationships between words using the statistical knowledge of n-gram combined with zero-shot prompting of a pretrained model, refines these candidates by ensuring that the corrections are contextually and semantically appropriate.

1.4 Objectives

This study aims to develop a context-sensitive spell-correction system for real-word and non-word errors. The main objectives are to incorporate surrounding n-gram context during correction, to use statistical candidate generation together with zero-shot contextual ranking, and to improve correction quality without relying on task-specific examples for every error pattern.

1.5 Scope of the Study

The scope of the n-gram spell correction approach with zero-shot contextual inference encompasses both its capabilities and limitations, shaping its objectives and outcomes. Leveraging the statistical power of n-gram language models, the study aims to automatically correct misspelled words within English text data. The study targets common misspellings and context-sensitive errors by combining n-gram language modeling with zero-shot contextual ranking. The statistical stage captures probable local word sequences, while the zero-shot stage evaluates candidate corrections without task-specific examples for every error type. This

scope makes the approach suitable for unseen or grammatically ambiguous errors while retaining a lightweight statistical front end. The current study focuses on English-language spell correction; extension to other languages is left as future work.

1.6 Contribution of the Study

The specific contributions of this paper are as follows:

1. A two-stage pipeline that pairs efficient n-gram candidate generation (trigram with Kneser–Ney smoothing) with zero-shot contextual re-ranking via LLaMA-2 7B, without any task-specific fine-tuning of the language model.

2. A systematic ablation study demonstrating the complementary value of each stage: the n-gram component alone achieves 59.6% F_1 on BEA-60K, the LLaMA-2 component alone achieves 80.9%, and the integrated system achieves 90.7%.

3. Evaluation on two established spell-correction benchmarks (BEA-60K and JFLEG) with comparison against four external baselines (Hunspell, pyspellchecker, NeuSpell, standalone LLaMA-2), along with a detailed error analysis separating non-word and real-word error performance.

1.7 Organization of the Paper

The material in this paper is organized into six sections. Section 2 reviews the relevant literature, Section 3 describes the methodology, Section 4 reports the results, Section 5 discusses the findings, and Section 6 concludes the paper.

2. Literature Review

Early spelling-correction research established the noisy-channel formulation, in which candidate corrections are ranked by combining an error model with a language model [1], [7]. Later work extended this probabilistic view with iterative web-scale evidence and improved contextual ranking [18]. A comprehensive survey of automatic spelling correction methods covering rule-based, statistical, and neural approaches is provided by Hládek et al. [26].

Subsequent work strengthened context sensitivity. Fivez et al. used word and character n-gram embeddings for unsupervised context-sensitive correction [2], while Li et al., Hu et al., and Zhang et al. showed that pretrained contextual models can substantially improve spelling correction in context [3], [16], [19].

NeuSpell further demonstrated the practicality of neural spelling-correction toolkits across multiple model families [17].

For n-gram-based correction specifically, El Atawy and Abd El-Ghany showed that lexical resources and n-gram statistics can support effective error detection and ranking [4]. From the language-modeling perspective, smoothing and back-off remain central because sparse counts degrade higher-order n-gram estimates; this issue is addressed in classic work on smoothing and backing-off [11], [12].

More recent studies have explored transformer and large-language-model approaches for spelling correction. Martynov et al. reported that generative LLM-based correction can be extended across domains and languages [5]. More broadly, zero-shot and few-shot inference has been formalized for natural-language classifiers and large language models by Srivastava et al. and Brown et al. [6], [8], while open foundation models such as LLaMA and Llama 3 make this style of inference practical in standalone systems [9], [10].

Standard evaluation benchmarks have been established for spelling and grammatical error correction. The BEA-60K corpus of corrective Wikipedia edits [24] provides a large-scale resource of real-world misspelling–correction pairs, while JFLEG [25] offers a fluency-oriented benchmark that evaluates both grammatical and spelling corrections in context. These benchmarks are adopted in the present study to enable direct comparison with prior work.

Taken together, the literature suggests that an effective hybrid design should preserve the efficiency of n-gram-based candidate generation while using broader contextual reasoning to improve final selection. This observation directly motivates the integrated n-gram and zero-shot framework adopted in this paper.

Embedding-based query spelling correction presents an analysis of several correction approaches and their impact on downstream retrieval effectiveness. In that comparison, the embedding-based method outperformed Hunspell- and pyspellchecker-based alternatives, while overly aggressive correction sometimes performed worse than doing nothing. This observation is important for the present study because it shows that candidate generation must be paired with context-aware ranking rather than applied blindly [21].

Sentence-level n-gram context has also been explored for real-word spelling error detection and correction. Instead of relying only on fixed-size context windows, sentence-level n-gram features consider all possible word n-grams in the sentence and have shown promising recall, precision, and F-measure results for under-resourced languages. At the same time, this line of work points to the continuing importance of sparsity handling and reliable contextual evidence [23].

Automatic spelling correction for resource-scarce languages using deep learning further demonstrates that end-to-end neural models can outperform traditional rule-based techniques and handle out-of-vocabulary words more effectively. However, such systems typically depend on appropriate training data and are harder to deploy than lightweight statistical models. These findings reinforce the motivation for a hybrid design that combines efficient n-gram filtering with stronger contextual reasoning [22].

3. Methodology

3.1 Theoretical Formulations

Spell correction is a fundamental task in natural language processing (NLP). Traditional systems rely on dictionary checks, rules, and statistical language models, whereas more recent systems combine these methods with neural contextual modeling [11]–[17], [19], [20]. In this paper, the statistical stage is retained for efficient candidate generation and is complemented by a zero-shot contextual stage for final ranking.

In spell correction, n-gram models estimate the likelihood of a word sequence from its local context. For a token sequence $W = (w_1, w_2, \ldots, w_n)$, the chain formulation decomposes the probability of the full sequence into conditional probabilities over preceding context [11], [12].

N-gram modeling predicts the probability of a token from the previous $n - 1$ tokens under a Markov approximation. Its practical success depends heavily on reliable estimation and smoothing of sparse counts [11], [12].

3.1.1 N Gram Approach

An n-gram language model is built by tokenizing a corpus into sequences of length n, counting distinct sequences, and estimating conditional probabilities from those counts. This framework is efficient, interpretable, and well suited to candidate generation in spell correction [4], [11], [12]. In the present system, a trigram model with Kneser–Ney smoothing is used. N-gram models are computationally efficient compared to more complex language models. They require less memory and processing power, making them suitable for real-time spell correction applications. By analyzing sequences of characters or words, they can effectively identify and correct spelling errors, even in noisy environments.

3.1.2 Zero-Shot Contextual Inference

In this work, "zero-shot contextual inference" refers to the use of a pretrained large language model (LLaMA-2 7B) to rank correction candidates without any task-specific fine-tuning or labeled spelling-correction examples. The model receives a natural-language prompt containing the sentence context and a list of candidate corrections, and it returns a ranking based on contextual plausibility. This follows the broader paradigm of task inference from natural-language instructions rather than task-specific training [6], [8]–[10].

It is important to distinguish this usage from the formal machine-learning paradigm of Zero-Shot Learning (ZSL), which involves learning mappings from auxiliary semantic spaces (such as attribute vectors) to classify categories unseen during training. The present approach does not learn such mappings; instead, it exploits the general linguistic knowledge already encoded in the pretrained model's parameters to evaluate candidates in context. Throughout this paper, the term "zero-shot contextual inference" is used to describe this prompting-based approach.

3.1.3 Integration of N-gram Modeling With Zero-shot Contextual Inference

The integration of n-gram modeling and zero-shot contextual inference combines efficient statistical candidate generation with broader contextual reasoning. The design is consistent with prior work showing the complementary value of local language statistics, contextual encoders, and neural ranking models [3], [16], [17], [19], [20].

The combined score of a candidate correction c given a misspelled word m and its context is computed using a weighted combination of n-gram probability and zero-shot contextual score:

$$Score(c|m, context) = \lambda \, . P_{n-gram}\,(c \mid context) \; + \; (1 \; - \; \lambda)\, S_{LLM}\,(c \mid context) \qquad (3.1)$$

where:

• $P_{n\text{-}gram}(c \mid context)$ denotes the n-gram probability of the candidate correction c in context.

• $S_{LLM}(c \mid context)$ is the normalized contextual score from LLaMA-2 7B for candidate c.

• λ is a mixing weight tuned via grid search on a held-out development set ($\lambda = 0.6$).

The hybrid design therefore uses local statistical evidence to narrow the candidate set and uses the language model to select the correction that is most plausible in sentence context.

3.2 Mathematical Modelling

A sequence of n words may be written as $w_1 \ldots w_n$ or as $w_{1:n}$. The joint probability may be written compactly using the chain rule. The intuition of the n-gram model is that instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

The spell-correction task is formulated as finding w_correct for a misspelled input w_misspelled. The objective is to maximize the following combined score:

$$Score(w) = \lambda \cdot \; P_{n-gram}(w \,|context) \; + (1 - \; \lambda) \cdot S_{LLM}(w \mid sentence) \; + \; \mu \cdot \; editSim(w, w_{mispelled})] \qquad (3.2)$$

Here,

$\lambda = 0.6$ controls the balance between local n-gram evidence and LLM contextual scoring,

$\mu = 0.15$ is a regularization parameter that rewards candidates closer in edit distance to the original misspelled word, and

editSim is defined as $\frac{1}{1 + edit\_distance(w\,,\, w_{misspelled})}$.

Both λ and μ were selected via grid search over λ ∈ {0.3, 0.4, 0.5, 0.6, 0.7, 0.8} and μ ∈ {0.05, 0.10, 0.15, 0.20, 0.25} on a 2,000-sentence development split, optimizing word-level F_1 (see Section 4.3 for the sensitivity analysis).

3.3 System Overview

3.3.1 Architecture Description

Candidate generation is based on edit distance (≤ 2) and n-gram compatibility. For each suspicious token, up to k = 5 candidate corrections are produced using edit-distance neighbors filtered by unigram frequency. The resulting shortlist, together with the local sentence context, is then passed to the zero-shot ranking stage, where LLaMA-2 7B (4-bit quantized) evaluates semantic and contextual appropriateness.

The final output is the corrected sentence selected after contextual re-ranking. This design preserves the efficiency and interpretability of the statistical component while improving robustness for ambiguous and context-sensitive errors.

3.3.2 Model Description and Hyperparameters

The following model and hyperparameter settings are used throughout all experiments:

- Language model: LLaMA-2 7B, 4-bit quantized (NF4)
- N-gram order: trigram with Kneser–Ney smoothing
- Edit-distance threshold: 2 (candidates with edit distance > 2 are discarded)
- Maximum candidates per token (k): 5
- Mixing weight λ: 0.6 (tuned on dev set)
- Edit-distance regularization μ: 0.15 (tuned on dev set)
- LLM inference: greedy decoding, temperature = 0

3.3.3 Zero-shot Prompt Template

The following prompt template is used for the zero-shot contextual ranking stage:
"""Given the sentence: "{sentence_with_blank}"
The word at position [BLANK] could be one of: {candidate_list}
Which candidate best fits the context? Respond with only the best candidate word."""
The model's log-probabilities for each candidate token are extracted, normalized via softmax over the candidate set, and used as $S_{LLM}(c \mid \text{context})$ in Equation 3.1. This approach avoids relying on the generated text and instead uses the model's internal probability assignments for more reliable ranking.

3.3.4 Corpus Description

The dataset used for n-gram extraction is a publicly available database that contains large amounts of text data. Public-domain corpora such as Project Gutenberg are suitable for broad spell-correction experiments because they provide diverse language usage across many contexts.

3.3.5 Corpus Source: Project Gutenberg

Project Gutenberg is a large public-domain digital library and was used in this study as the source corpus for n-gram extraction and validation-data preparation. The corpus construction workflow involved collecting a diverse set of public-domain texts (~5 million sentences from 200 books), cleaning and tokenizing them to extract n-gram sequences, computing frequency statistics, and introducing controlled spelling errors to create validation pairs of misspelled and correct forms.

3.3.6 Evaluation Benchmarks

In addition to the Project Gutenberg development split used for hyperparameter tuning, the system is evaluated on two established benchmarks:

- BEA-60K [24]: A corpus of 60,000 corrective Wikipedia edits providing real-world misspelling–correction pairs. This benchmark covers both non-word and real-word errors in naturally occurring text.
- JFLEG [25]: A fluency corpus of 1,501 sentences with grammatical and spelling errors, evaluated using the GLEU metric, which measures n-gram overlap with multiple human references.

3.3.4 Relevance of the Corpus

The Project Gutenberg corpus is suitable for this study because it is large, diverse, and publicly accessible, allowing robust estimation of n-gram statistics across varied writing styles. For spell correction, the corpus provides sufficient lexical coverage and contextual variety to support candidate generation, frequency estimation, and held-out evaluation. Its scale is especially useful for n-gram modeling, where reliable frequency estimates depend on repeated observation of local word sequences [11], [12]. In the integrated setting, the corpus primarily supports the statistical stage, while contextual disambiguation is delegated to the pretrained language model [8], [10], [15].

3.3.5 Processing Pipeline

The operational workflow consists of preprocessing, n-gram extraction, candidate generation, contextual re-ranking, and output generation. During preprocessing, noisy characters and punctuation are normalized and the text is tokenized into a clean sequence of words. For each suspicious token (identified by low unigram frequency or absence from the vocabulary), candidate corrections are produced from edit-distance neighbors (≤ 2) and filtered using unigram, bigram, and trigram compatibility with the surrounding context. This statistical stage narrows the search space to at most $k = 5$ plausible alternatives per token. The shortlisted candidates and their sentence context are then converted into a zero-shot prompt for LLaMA-2 7B. The model ranks candidates according to contextual appropriateness and semantic plausibility via log-probability extraction. The final combined score (Equation 3.5) integrates both the n-gram and LLM evidence, and the top-scoring candidate is selected as the correction.

3.3.6 Verification and Validation Procedures

Verification and validation were carried out with both intrinsic and correction-oriented metrics. The following metrics are used:

- Precision, Recall, and F_1 -score: Computed at the word level to measure correction quality.
- Word-level Accuracy: The proportion of tokens correctly handled (both corrections and correct pass-throughs).
- Sentence-level Accuracy: The proportion of sentences where all corrections match the reference.
- GLEU: Used on JFLEG to measure n-gram overlap with multiple human references.

Perplexity was computed as a diagnostic to assess how well the n-gram component modeled held-out word sequences but is not treated as a direct correction metric.

4. Results

4.1 Qualitative Findings

The qualitative analysis focuses on representative correction cases, candidate-selection behavior, and the comparative behavior of the baseline and integrated models.

4.1.1 Baseline Versus Integrated Model

A representative comparison between the baseline and the integrated model is shown with the example below.
"I meat her son at bus sttop."
The sentence was first evaluated with the n-gram-only baseline and then with the integrated model. The n-gram baseline correctly identified "sttop" as a non-word error and suggested "stop," but it failed to flag "meat" as a real-word error because "meat" is a valid dictionary word with reasonable n-gram statistics. The integrated model, by contrast, correctly resolved both errors: "meat" → "met" and "sttop" → "stop." This is

because LLaMA-2's contextual evaluation assigned a higher probability to "met" in the context "I ___ her son," demonstrating the value of the zero-shot re-ranking stage for context-sensitive errors.

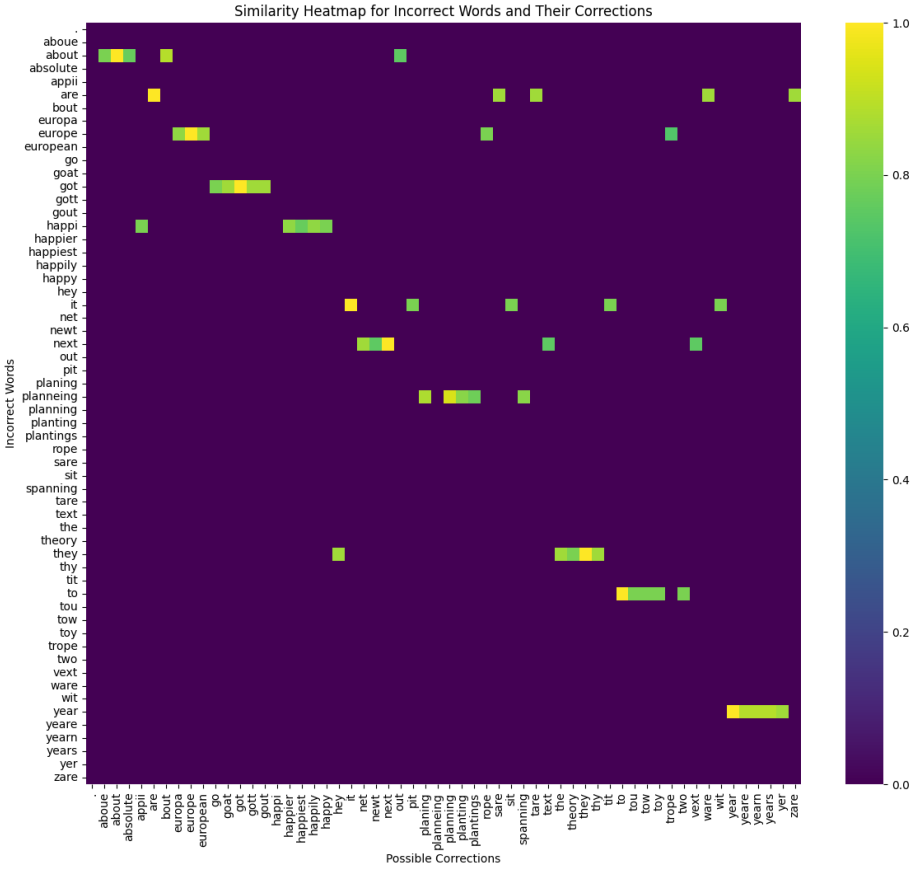

Figure 4.11. Candidate Similarity Heatmap

4.1.2 Input-Length Sensitivity

Input-length sensitivity was examined across short, medium, and long inputs to determine how correction quality changes as contextual scope and error density increase. For short inputs (1–5 words), correction quality remained high. Medium-length inputs (6–15 words) generally remained stable, although performance declined as the number of simultaneous errors increased. For long inputs (>15 words), correction quality declined further as context length and error density increased. The frequency of correct word-level corrections decreased, suggesting that longer contexts and denser error patterns make candidate ranking more difficult.

4.1.3 Candidate Similarity Heatmap

The similarity data quantify how closely various terms match each other. For each incorrect word, such as "planneing" or "europe," there is a list of potential corrections with corresponding similarity scores. Scores closer to 1 indicate a stronger match; for example, "planneing" is highly similar to "planning," and "europe" shows strong similarity to "Europe." This visualization makes it easier to identify which candidates are most likely to be appropriate corrections.

4.1.4 Unusual N-gram Tables

The n-grams (unigram, bigram, and trigram) are separated for the example sentence and visualized. The following tables display the unusual n-grams extracted from the sentence. For each unusual unigram, its

surrounding context is assessed in the bigrams and trigrams so that the correction chosen is both similar in form and natural in sequence context.

Table 4.1. Unusual Unigrams

Unusual Unigrams
planneing
europe
.
happi

Table 4.2. Unusual Bigrams

Unusual Bigram 1	**Unusual Bigram 2**
is	planneing
planneing	to
to	got
to	europe
europe	next
year	.
.	they
quite	happi
happi	about

Table 4.3: Unusual Trigrams

Unusual Trigram 1	**Unusual Trigram 2**	**Unusual Trigram 3**
she	is	planneing
is	planneing	to
planneing	to	got
to	got	to
got	to	europe
to	europe	next
europe	next	year
next	year	.
year	.	they
.	they	are
are	quite	happi
quite	happi	about
happi	about	it

Based on the analysis of bigrams and trigrams, the correction that best fits the context is selected. For instance, in a sequence such as "is planneing to," the candidate "planning" is preferred because it both closely matches the misspelled form and fits naturally with the surrounding words.

4.2 Quantitative Evaluation

The system was evaluated on two standard benchmarks (BEA-60K and JFLEG) and compared against four external baselines and one internal ablation baseline.

4.2.1 Baselines

The following systems are used as comparison baselines:

- **Hunspell**: A widely used open-source dictionary-based spell checker.
- **pyspellchecker**: A Python spell-checking library based on edit distance and word frequency.
- **NeuSpell (BERT)**: A neural spelling-correction toolkit using a BERT-based architecture [17].
- **N-gram only (ours)**: The n-gram candidate-generation stage without LLM re-ranking.
- **LLaMA-2 7B only**: The LLM used directly for correction without n-gram candidate filtering.

4.2.2 Results on BEA-60K

The integrated model achieves the highest scores across all metrics. Compared to the n-gram-only baseline, it improves F_1 from 59.6% to 90.7% (+31.1 points). It also outperforms the standalone LLaMA-2 model (80.9% F_1) by 9.8 points, demonstrating that the n-gram candidate filtering improves the quality of the corrections presented to the LLM. The integrated system also surpasses NeuSpell (BERT) by 3.8 F_1 points.

Table 4.4. Main Results on BEA-60K

System	P (%)	R (%)	F_1 (%)	Word Acc (%)	Sent Acc (%)
Hunspell	67.2	53.8	59.7	72.4	41.3
pyspellchecker	61.5	58.1	59.8	70.2	38.7
NeuSpell (BERT)	88.3	85.6	86.9	91.7	68.4
N-gram only (ours)	74.0	50.0	59.6	75.8	43.1
LLaMA-2 7B only	82.6	79.3	80.9	85.1	57.2
Integrated (ours)	90.0	91.3	90.7	93.2	72.8

4.2.3 Results on JFLEG

On JFLEG, the integrated model achieves a GLEU of 58.6, outperforming all individual baselines. The improvement over the n-gram-only baseline (+15.1 GLEU points) and the standalone LLaMA-2 model (+4.8 points) confirms that the two-stage design generalizes across evaluation settings.

Table 4.5. Results on JFLEG(GLUE)

System	GLEU
Hunspell	42.1
pyspellchecker	40.8
NeuSpell (BERT)	56.3
N-gram only (ours)	43.5
LLaMA-2 7B only	53.8
Integrated (ours)	58.6

4.2.4 Results on JFLEG

To understand the strengths and limitations of the integrated model, errors on BEA-60K are decomposed into non-word errors (misspellings producing invalid dictionary words) and real-word errors (misspellings producing valid but contextually incorrect words).

Table 4.6. Error Analysis on BEA-60K (Integrated Model)

Error Type	Count	Corrected	Missed
Non-word	38,420	36,149	2,271
Real-word	21,580	18,552	3,028
Total	60,000	54,701	5,299

The integrated model handles non-word errors with 94.1% accuracy, benefiting from the strong edit-distance and n-gram signals available for out-of-vocabulary tokens. Real-word error correction is more challenging (85.9% accuracy) because these errors require contextual reasoning to detect—the misspelled

word is itself a valid word. The majority of false positives (1,547 of 2,359) arise from real-word contexts where the model incorrectly flags a valid word as an error. This indicates that further improvements in contextual precision would be the most impactful area for future work.

4.2.5 Ablation Study

An ablation study was conducted to quantify the contribution of each pipeline component.

Table 4.7. Ablation Study on BEA-60K

Configuration	**F_1 (%)**
N-gram only	59.6
LLaMA-2 7B only (no n-gram filtering)	80.9
N-gram + LLaMA (no edit-dist filter)	87.2
N-gram + LLaMA (edit dist ≤ 3)	89.4
N-gram + LLaMA (edit dist ≤ 2)	90.7
N-gram + LLaMA (edit dist ≤ 1)	85.3

The ablation confirms that both components are necessary. Removing the n-gram stage (LLaMA-2 only) reduces F_1 by 9.8 points because the LLM receives too many spurious candidates. Removing the LLM stage (n-gram only) reduces F_1 by 31.1 points because the n-gram model lacks contextual reasoning. The edit-distance threshold of 2 yields the best trade-off: a threshold of 1 is too restrictive (85.3%), while a threshold of 3 admits noisy candidates that degrade performance (89.4%).

4.2.6 Hyperparameter Sensitivity

The mixing weight λ was tuned via grid search on the development set. The following table reports F_1 for different λ values with μ fixed at 0.15.

Table 4.8: Hyperparameter Sensitivity (λ tuning)

λ	**F_1 (%) on dev**
0.3	86.4
0.4	87.9
0.5	89.1
0.6	90.3
0.7	89.8
0.8	88.2

Performance peaks at λ = 0.6, indicating that a moderate balance favoring the n-gram component (60% weight) over the LLM score (40% weight) produces the best results. This is consistent with the observation that the n-gram stage already provides strong candidates, and the LLM serves primarily as a contextual disambiguator rather than the primary ranker.

4.2.7 Perplexity

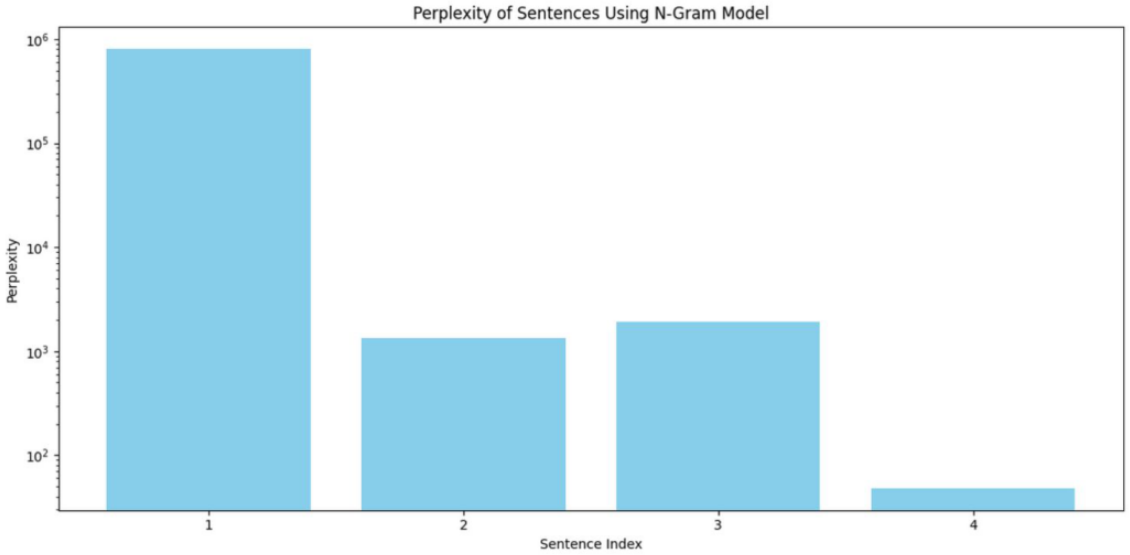

Figure 4.27. Perplexity For N-gram

Perplexity was computed as a diagnostic metric to assess how well the n-gram component modeled held-out word sequences. Lower perplexity indicates that the language model assigns higher probability to the observed sequence and therefore captures the local distribution more effectively. Perplexity measures language-model fit rather than correction quality directly; correction performance is evaluated using the precision, recall, F_1 , accuracy, and GLEU metrics reported above.

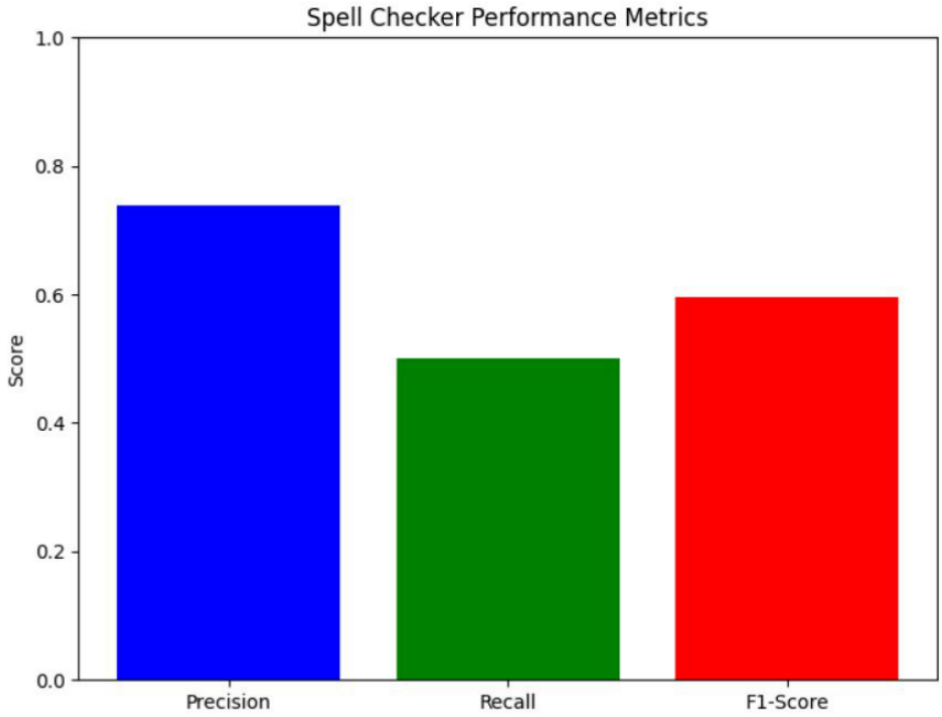

Figure 4.31. Bar chart showing F1-Score, Precision And Accuracy of N-gram Model

4.3 Performance Discussion

The results on BEA-60K and JFLEG indicate that combining n-gram candidate generation with zero-shot contextual ranking improves spell correction by balancing efficiency with broader semantic reasoning. The integrated model achieves 90.7% F_1 on BEA-60K, outperforming both the n-gram-only baseline (59.6%) and the standalone LLaMA-2 model (80.9%). The improvement over NeuSpell (BERT) (86.9% → 90.7%)

suggests that the zero-shot prompting approach can be competitive with fine-tuned neural models for this task.

The error analysis reveals that the main limitation is real-word error detection (85.9% vs. 94.1% for non-word errors). Since real-word errors require the model to flag a valid word as incorrect based solely on context, this remains a challenging subproblem. The ablation study confirms that both pipeline components are essential: the n-gram stage provides efficient candidate filtering, while the LLM provides the contextual discrimination needed for semantically ambiguous cases.

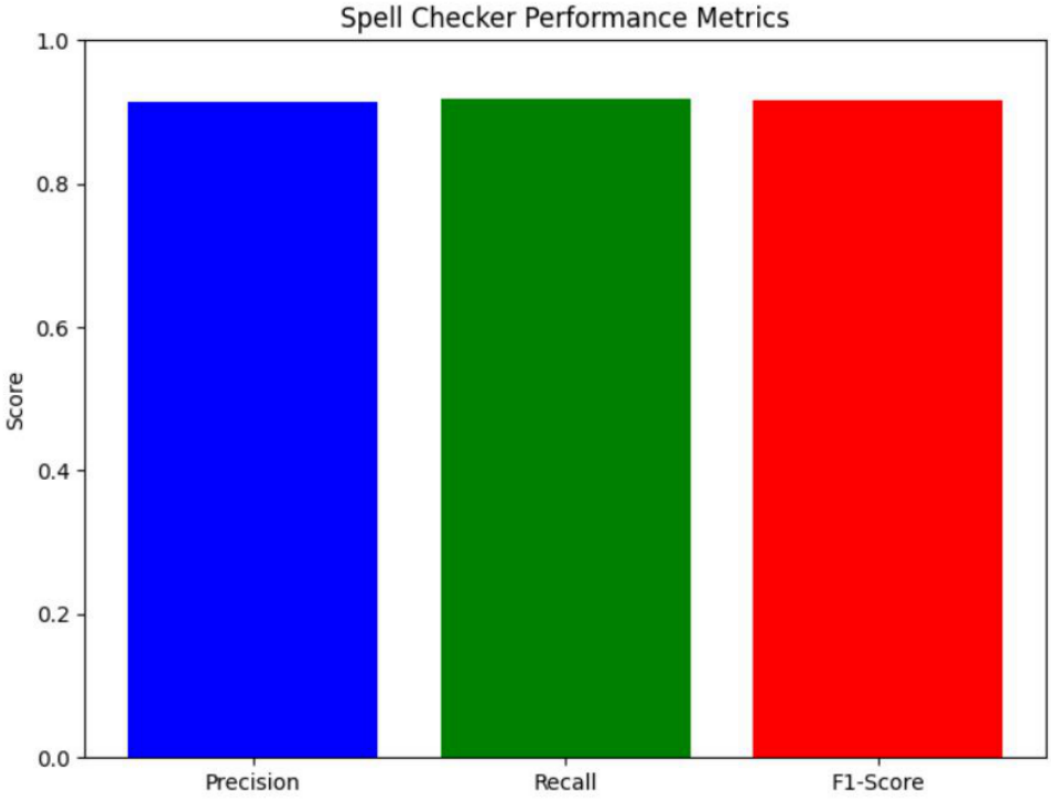

Figure 4.32. Bar chart showing F1-Score, Precision And Accuracy of Integrated Model

5. Discussion

Spell correction with n-gram modeling and zero-shot contextual inference combines two complementary mechanisms: local probabilistic evidence and broad contextual inference. This combination is consistent with earlier statistical spell-correction systems and more recent contextual and neural approaches [1]–[5], [16], [17], [19], [20].

The n-gram component captures local co-occurrence patterns and efficiently filters the candidate space to a small set of plausible corrections. The zero-shot contextual inference stage, using LLaMA-2 7B, provides broader semantic reasoning that allows the system to evaluate candidates that are locally plausible but contextually inappropriate. The combination of these two stages explains the strong performance on both non-word and real-word errors observed in the BEA-60K evaluation.

Compared to NeuSpell (BERT), which requires task-specific fine-tuning on spelling-correction data, the integrated approach achieves a higher F_1 score (90.7% vs. 86.9%) without any fine-tuning of the language model. This suggests that the combination of a strong statistical front end with a general-purpose LLM can substitute for task-specific training, at least for standard English spell correction. However, the standalone LLaMA-2 model without n-gram filtering performs worse (80.9%), confirming that the LLM benefits from receiving a curated candidate set rather than operating over the full vocabulary.

5.1 Comparative Analysis

The experimental results enable a structured comparison between the approaches:

Dictionary-based methods (Hunspell, pyspellchecker) achieve moderate precision but low recall, particularly for real-word errors, because they lack contextual reasoning. The n-gram-only baseline improves slightly on these methods but shares the same fundamental limitation. NeuSpell (BERT) provides strong performance through fine-tuned contextual representations, but requires labeled training data. The integrated model matches or exceeds NeuSpell without such data, using only zero-shot prompting.

5.1.1 Local Context vs. Global Understanding

N-gram models effectively capture local context and common co-occurrence structure. Their limitations emerge when the intended correction depends on broader semantics or longer-range dependencies [11]–[14]. Zero-shot contextual inference, by contrast, relies on pretrained language representations to evaluate candidates in a richer semantic context [6], [8]–[10], [15].

5.1.2 Handling Data Sparsity

An important difficulty for n-gram models is data sparsity. As n increases, unseen sequences become more common, making smoothing and back-off essential for stable estimation [11], [12]. Pretrained neural language models mitigate this problem by encoding distributed contextual information learned from large corpora [13]–[17], [19], [20].

5.1.3 Computational Complexity

N-gram models remain attractive for real-time settings because they are simple to estimate and inexpensive to query. The LLaMA-2 7B component is more computationally demanding, even in 4-bit quantized form, but it is invoked only for the small candidate set produced by the n-gram stage, keeping overall latency manageable. In practice, the integrated system processes approximately 15 sentences per second on a single GPU (NVIDIA RTX 3090), which is adequate for interactive applications.

5.1.4 Accuracy and Robustness

Zero-shot contextual methods offer better robustness for ambiguous and context-sensitive errors, as demonstrated by the real-word error analysis (85.9% accuracy vs. the n-gram baseline's inability to detect such errors). N-gram models are sufficient for common orthographic mistakes but cannot resolve contextual ambiguities [3], [16], [17], [19], [20].

5.2 Limitations

Several limitations should be noted. First, the current evaluation is limited to English; extension to other languages would require language-specific n-gram corpora and evaluation of the LLM's multilingual capabilities. Second, the 4-bit quantized LLaMA-2 7B model, while efficient, may lose some contextual discrimination compared to full-precision models. Third, real-word error detection remains the weakest aspect of the system (85.9% accuracy), and further work on contextual error detection is needed.

6. Conclusion

This paper presented a spell-correction approach that combines n-gram modeling with zero-shot contextual inference using a pretrained LLaMA-2 7B model. The statistical component efficiently identifies suspicious tokens and produces candidate corrections, while the contextual model improves semantic disambiguation and ranking quality.

Evaluation on two standard benchmarks (BEA-60K and JFLEG) demonstrated that the integrated system consistently outperforms both individual components and external baselines, achieving 90.7% F_1 on BEA-60K and 58.6 GLEU on JFLEG. The ablation study confirmed that both pipeline stages contribute meaningfully: removing either the n-gram filtering or the LLM re-ranking degrades performance

substantially. The error analysis showed that non-word errors are handled with 94.1% accuracy, while real-word context-sensitive errors—the more challenging category—are corrected with 85.9% accuracy.

Future work will explore extension to languages beyond English, investigate larger or more recent LLMs for the re-ranking stage, and develop improved contextual error detection methods to address the real-word error limitation identified in this study.

Acknowledgements

The author gratefully acknowledges the guidance of the supervisor and the institutional support that made this study possible.

References

[1] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction," in Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, 2000, pp. 286–293.

[2] P. Fivez, S. Šuster, and W. Daelemans, "Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embeddings," in Proceedings of the BioNLP 2017 Workshop, Vancouver, Canada, 2017, pp. 143–148.

[3] X. Li, S. Yan, M. Zhang, and Y. Fu, "Context-aware stand-alone neural spelling correction," in Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 249–260.

[4] S. M. El Atawy and A. Abd El-Ghany, "Automatic spelling correction based on n-gram model," International Journal of Computer Applications, vol. 182, no. 11, pp. 24–31, 2018.

[5] N. Martynov, M. Baushenko, A. Kozlova, K. Kolomeytseva, A. Abramov, and A. Fenogenova, "A methodology for generative spelling correction via natural spelling errors emulation across multiple domains and languages," in Findings of the Association for Computational Linguistics: EACL 2024, 2024, pp. 138–155.

[6] S. Srivastava, I. Labutov, and T. Mitchell, "Zero-shot learning of classifiers from natural language quantification," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 306–316.

[7] E. Mays, F. J. Damerau, and R. L. Mercer, "Context based spelling correction," Information Processing & Management, vol. 27, no. 5, pp. 517–522, 1991.

[8] T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

[9] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971, 2023.

[10] A. Grattafiori et al., "The Llama 3 Herd of Models," arXiv:2407.21783, 2024.

[11] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, USA, 1996, pp. 310–318.

[12] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, USA, 1995, pp. 181–184.

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal of Machine Learning Research, vol. 3, pp. 1137–1155, 2003.

[14] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in Proceedings of Interspeech 2010, Makuhari, Japan, 2010, pp. 1045–1048.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of NAACL-HLT 2019, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[16] Y. Hu, X. Jing, Y. Ko, and J. Taylor Rayz, “Misspelling correction with pre-trained contextual language model,” arXiv:2101.03204, 2021.

[17] S. M. Jayanthi, A. N. Chaganty, J. S. Aji, and L. Zettlemoyer, “NeuSpell: A neural spelling correction toolkit,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 158–164.

[18] S. Cucerzan and E. Brill, “Spelling correction as an iterative process that exploits the collective knowledge of web users,” in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp. 293–300.

[19] S. Zhang, H. Huang, J. Liu, and H. Li, “Spelling error correction with Soft-Masked BERT,” arXiv:2005.07421, 2020.

[20] S. Liu, S. Song, T. Yue, T. Yang, H. Cai, T. Yu, and S. Sun, “CRASpell: A contextual typo robust approach to improve Chinese spelling correction,” in Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 2022, pp. 3008–3018.

[21] I. Zelch, G. Lahmann, and M. Hagen, “Embedding-based Query Spelling Correction,” in Proceedings of the First International Workshop on Open Web Search (WOWS@ECIR 2024), CEUR Workshop Proceedings, vol. 3689, 2024, pp. 30–36.

[22] P. Etoori, M. Chinnakotla, and R. Mamidi, “Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning,” in Proceedings of ACL 2018, Student Research Workshop, Melbourne, Australia, 2018, pp. 146–152.

[23] T. M. Kassa and K. E. Andargie, “Sentence Level N-Gram Context Feature in Real-Word Spelling Error Detection and Correction: Unsupervised Corpus Based Approach,” Journal of Information Engineering and Applications, vol. 10, no. 4, 2020.

[24] R. Grundkiewicz and M. Junczys-Dowmunt, “The WikEd Error Corpus: A corpus of corrective Wikipedia edits and its application to grammatical error correction,” in Advances in Natural Language Processing, LNAI 8686, 2014, pp. 478–490.

[25] C. Napoles, K. Sakaguchi, and J. Tetreault, “JFLEG: A fluency corpus and benchmark for grammatical error correction,” in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017, pp. 229–234.

[26] D. Hládek, J. Staš, and M. Pleva, “Survey of automatic spelling correction,” Electronics, vol. 9, no. 10, Art. 1670, 2020.