AUTOMATIC SPELLING ERROR DETECTION AND CORRECTION IN UZBEK TEXTS: A MORPHOLOGY- AND CONTEXT-AWARE HYBRID FRAMEWORK

Sobirova Nazira
Department of Computational Linguistics and Digital Technologies, doctoral researcher (PhD student)
Tashkent State University of Uzbek Language and Literature
Email: nazirasobirova073@gmail.com

DOI: 10.67620/sai.510c0004

ABSTRACT

Automatic spelling correction in Uzbek must address productive agglutination, Latin–Cyrillic variation, apostrophe-like signs, proper names, borrowings, and context-dependent real-word errors. This conceptual and technical study proposes a morphology- and context-aware hybrid framework that separates orthographic normalization, lexical and morphological analysis, error detection, candidate generation, contextual ranking, and confidence-based acceptance or abstention. A selective narrative synthesis of 17 works compares approaches by error coverage, morphological suitability, context dependence, resource requirements, interpretability, and overcorrection risk. The proposed candidate score combines normalized contextual likelihood, weighted edit cost, morphological plausibility, and error-channel evidence. Parameter selection uses validation-only tuning, morphological ambiguity aggregation, calibrated confidence, and an explicit no-change candidate. Uzbek-specific policies include reversible apostrophe normalization, script-aware handling, suffix-sequence constraints, contextual screening of valid words, and conservative protection of unknown names and technical terms. A prospective evaluation protocol distinguishes natural from synthetic errors, prevents document-level data leakage, and measures candidate Recall@k, detection precision, recall and F1, correction accuracy, overcorrection, calibration, and runtime. Proposed ablations isolate morphology, context, error-channel modeling, and abstention. No corpus construction, implementation results, performance gains, or statistically significant superiority is claimed. The contribution is a testable technical specification for subsequent Uzbek-language implementation and empirical validation.

ACCEPTED

07.11.26

Keywords: *Uzbek NLP; spelling correction; spell checker; morphology; N-gram; edit distance; contextual ranking; agglutinative language.*

CC BY

Introduction

Spell checking is one of the earliest practical applications of language technology. Classical systems first identify an unfamiliar form and then attempt to propose a plausible replacement. Ralph Gorin's (1974) SPELL program is an important historical example: it reads text, queries words it does not recognize, supports multiple auxiliary dictionaries, strips prefixes and suffixes, records exceptions with context, and can automatically reuse corrections previously supplied by the user. These design ideas remain

ISSN: 2181-3337 | SCIENTISTS.UZ

relevant because modern systems still need a lexicon, a mechanism for generating alternatives, and a policy for deciding when a replacement is justified.

The central difficulty is that an unknown string is not necessarily an error and a dictionary word is not necessarily correct in context. The spell-checking literature therefore distinguishes non-word errors from real-word errors (Pirinen & Lindén, 2010). A non-word such as a form created by an omitted or substituted character can often be detected by lexicon lookup. A real-word error is harder: the observed form may itself be a valid word, but it is inappropriate in its sentence. Contextual systems address this limitation by combining an error model with language-model evidence (Richter et al., 2012; Whitelaw et al., 2009).

Uzbek makes this distinction especially important. It is an agglutinative language in which productive suffixation generates many valid surface forms, so a finite dictionary cannot safely represent all grammatical words. Uzbek text also exhibits Latin/Cyrillic variation and several apostrophe-like characters. Consequently, a practical spelling corrector should coordinate orthographic preprocessing, morphological analysis, candidate generation, and contextual selection rather than treating every out-of-vocabulary form as misspelled. Work on Uzbek morphological analysis, stemming, POS tagging, and root/stem parsing provides relevant linguistic infrastructure for this purpose (Elov, Adalı, et al., 2023; Elov, Hamroyeva, & Axmedova, 2023; Elov et al., 2024). Related Uzbek-language work also describes spelling-assessment software (Murodov & Xazratqulov, 2022) and hidden Markov model approaches to part-of-speech tagging (Elov, Sirojiddinov, et al., 2023).

This paper presents a technical specification rather than a validated production system. Its specific contributions are: (1) a script- and morphology-aware taxonomy of Uzbek spelling errors; (2) an explicitly delimited, criteria-based synthesis of prior techniques; (3) an implementable candidate-scoring and abstention rule with validation-based parameter selection; (4) Uzbek-specific policies for reversible sign normalization, ambiguous affix analyses, real-word errors, and protected rare words; and (5) a leakage-aware evaluation and ablation protocol. These are design contributions, not claims of demonstrated accuracy gains.

Review Scope and Analytical Method

This study uses a purposive, source-grounded narrative technical review, not a systematic review or meta-analysis. The 17 works cited in the manuscript form the reviewed source set; their inclusion is based on direct relevance to one or more components of spelling correction or Uzbek morphological processing. The source set was not assembled through a documented exhaustive database search, and no PRISMA-style completeness claim is made. To make the analytical method reproducible, the following selection and comparison rules are stated as a protocol for an expanded future review.

Comparative assessment rubric. Each cited approach is assessed qualitatively along six axes: (i) non-word versus real-word detection coverage; (ii) compatibility with productive stem–suffix combinations; (iii) use of sentence context; (iv) required annotated data and lexical resources; (v) interpretability and computational cost; and (vi) susceptibility to overcorrection. Table 2 summarizes functional strengths and limitations rather than empirical rankings. Evidence from English or multilingual query correction is treated as a design analogy, not as measured evidence for Uzbek full-sentence text.

Source identification and eligibility protocol. A subsequent systematic update should search ACL Anthology, IEEE Xplore, SpringerLink, Google Scholar, and relevant Uzbek-language journal repositories with combinations of “spell checking”, “spelling correction”, “real-word error”, “weighted edit distance”, “finite-state morphology”, “Uzbek”, “agglutinative”, and “contextual ranking”. Eligible items must describe an identifiable algorithm, linguistic resource, evaluation method, or Uzbek-specific language-processing constraint. Duplicates, purely promotional software descriptions, and items without sufficient technical detail should be excluded. Search dates, exact query strings, screening counts, and reasons for

ISSN: 2181-3337 | SCIENTISTS.UZ

exclusion must be recorded when this protocol is actually executed; no such counts are asserted for the present manuscript.

Jurafsky and Martin (2024) define an N-gram as a sequence of n words and describe N-gram language models as probabilistic models that estimate a word from its preceding context and assign probabilities to sequences. This is directly relevant to spelling correction because competing alternatives can be compared by contextual probability. Whitelaw et al. (2009) similarly combine a statistical error model with an N-gram language model and use token N-grams to make context-appropriate corrections. Korektor combines multiple language models with an error model and is designed to detect errors that simple dictionary-based spell checkers may miss (Richter et al., 2012).

Finite-state spell checking provides another transferable design. Pirinen and Lindén (2010) describe spell checking with a lexical automaton, weighted language/error models, and Internet corpora, while explicitly separating isolated non-word errors from context-based real-word errors. This distinction motivates a two-level Uzbek design: character/morphology-sensitive candidate generation followed by contextual ranking.

Recent multilingual work also emphasizes that spell correction commonly consists of candidate generation followed by candidate ranking. Sharma et al. (2023) combine contextual, behavioral, phonetic, and neural ranking signals in a fast multilingual search-query setting. The present framework adopts the same functional separation but does not assume access to proprietary behavioral signals.

Uzbek-Specific Spelling Error Taxonomy

Character-Level and Non-Word Errors

A widely used typology of typographic errors includes single-character insertion, deletion, substitution, and transposition of adjacent characters (Kumar et al., 2018). These operations are also consistent with the classical SPELL correction perspective. For Uzbek, examples can be constructed around ordinary letters and orthographically meaningful signs. Candidate generation should therefore support weighted versions of all four edit operations rather than a single undifferentiated similarity score.

Table 1

Illustrative Uzbek Spelling-Error Categories and Processing Requirements

Error type	Illustrative input	Possible target	Processing need
Deletion	Nma	nima	insert missing character; check context
Insertion	Kitobb	kitob	remove extra character
Substitution	Maktap	maktab	substitute likely confused character
Transposition	Maktba	maktab	swap adjacent characters
Apostrophe/sign	Ozbek	o‘zbek	orthographic sign candidate + context
Real-word	valid word in wrong context	context-dependent	language-model ranking required
Morphological	valid stem + malformed suffix	correct inflected form	stem/affix analysis
Proper name / borrowing	unseen form	often preserve	avoid forced correction

Morphology and Lexical Validity

Dictionary lookup remains useful, but it is insufficient for Uzbek. General spell-checker descriptions note that morphology requires language-dependent processing and becomes more complicated for languages with agglutination and rich inflection (Bhaire et al., 2015). Uzbek-specific research on morphological analyzers and root/stem parsing is therefore relevant to deciding whether an unfamiliar

ISSN: 2181-3337 | SCIENTISTS.UZ

surface form can be decomposed into a plausible stem and suffix sequence (Elov, Hamroyeva, & Axmedova, 2023; Elov et al., 2024). A failed morphological analysis should increase suspicion but should not by itself prove that a token is erroneous, because names, borrowings, abbreviations, and new words may be absent from the analyzer.

Proper Names and Dictionary Overcoverage

Dictionary design creates a second problem: adding more entries can reduce false alarms while simultaneously allowing some errors to pass undetected. Lavoshnikova's (2015, 2017) analyses of Microsoft Word illustrate how low-frequency vocabulary and proper names can cause misspellings or unintended segmentations to be accepted because the resulting forms happen to exist in the system dictionary. For Uzbek, the same risk argues for treating lexical membership as one feature rather than the final decision rule.

Critical Synthesis of Spell-Checking Approaches

Table 2

Functional Comparison of Spelling-Correction Approaches

Approach	Strength	Main limitation	Role in proposed Uzbek system
Dictionary lookup	transparent and fast	unknown valid forms; real-word errors	initial lexical evidence
Affix stripping / morphology	handles productive word forms	analyzer coverage and ambiguity	stem–suffix plausibility
Edit distance	simple candidate generation	short distance ≠ contextual correctness	weighted candidate generation
N-gram LM	uses local context	sparsity; limited long context	candidate reranking
Finite-state models	efficient lexical/error constraints	resource construction needed	Constrained candidate space
Statistical error model	models likely misspellings	needs representative counts	error likelihood
Neural contextual model	captures broader context	latency/data/calibration	optional reranker
Hybrid	combines complementary evidence	engineering complexity	full proposed framework

Survey literature identifies dictionary lookup and N-gram analysis as common error-detection techniques, while correction methods include edit distance, similarity keys, rule-based, N-gram, probabilistic, and neural approaches (Kumar et al., 2018). The important design implication is that these methods solve different subproblems. Edit distance is suitable for generating orthographically nearby candidates; N-gram probability is suitable for comparing candidates in context; morphology helps reject structurally implausible analyses; and a learned error model can encode frequent confusions.

Web-scale statistical spell checking demonstrates the value of combining these signals. Whitelaw et al. (2009) derive misspelling information and candidate terms from large noisy web data and use token N-grams for contextual correction. Their approach also illustrates a practical advantage of data-driven models: they need not rely exclusively on a fixed handcrafted lexicon. However, direct transfer to Uzbek would require Uzbek-specific corpus collection, normalization policy, and evaluation.

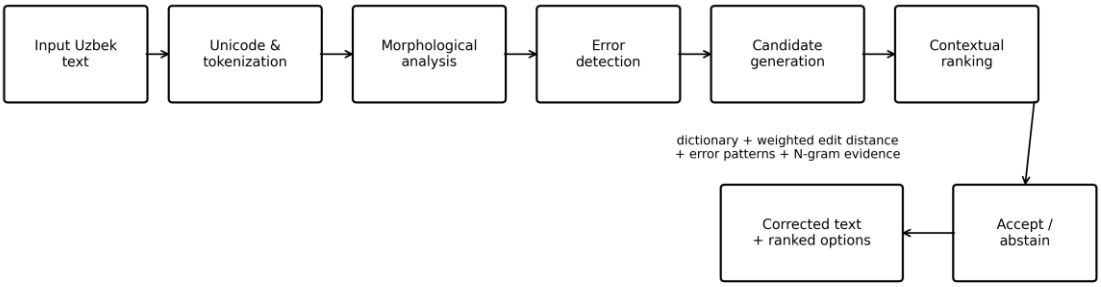
Korektor shows why context matters even when every token is individually legal: language and error models can rank replacements and detect errors that coincide with existing word forms (Richter et

ISSN: 2181-3337 | SCIENTISTS.UZ

al., 2012). This principle is central to the proposed Uzbek system. The framework therefore never treats edit distance alone as the correction decision.

Figure 1

Proposed Morphology- and Context-Aware Hybrid Architecture for Uzbek Spelling Correction



Proposed Hybrid Framework

Orthographic Preprocessing and Tokenization

The system first preserves the raw input and creates a working representation. Unicode-equivalent or policy-approved apostrophe-like characters can be normalized for processing, while URLs, e-mail addresses, code fragments, identifiers, and other protected spans should be excluded from ordinary lexical correction. Tokenization must preserve enough positional information to reconstruct the original text after edits. Separating preprocessing from subsequent analysis follows the modular NLP pipeline perspective described by Elov, Khamroeva, and Xusainova (2023).

Morphological Analysis

Each eligible token is checked for lexical and morphological plausibility. A known dictionary form can be accepted provisionally, while an unknown form is passed to stem–affix analysis. Uzbek morphological work provides a basis for this stage (Elov, Adalı, et al., 2023; Elov, Hamroyeva, & Axmedova, 2023; Elov et al., 2024). The output should contain possible stems, suffix sequences, and grammatical features when available. Importantly, an unknown analysis is represented as uncertainty rather than as an automatic error label.

Error Detection

Detection combines lexical status, morphological plausibility, character-pattern evidence, and contextual anomaly. Non-word errors may be detected primarily from lexical/morphological evidence. Real-word errors require contextual evidence because the observed token may itself be well formed. A thresholded anomaly score can be used to decide whether candidate generation is warranted, but thresholds must be tuned on validation data rather than selected arbitrarily.

Candidate Generation

For a suspicious token u, the candidate set C(u) is constructed from dictionary neighbors, weighted edit operations, known error–correction pairs, and morphologically plausible reconstructions. The unchanged token u must remain in the candidate set so that the system can abstain from an unnecessary edit. Character operations include insertion, deletion, substitution, and adjacent transposition. Costs can be made Uzbek-sensitive, for example by assigning different weights to empirically frequent keyboard or apostrophe confusions, but such weights require corpus evidence.

Contextual Ranking

N-gram language models provide a transparent baseline for contextual ranking. A bigram or trigram model estimates how probable a candidate is given neighboring words (Jurafsky & Martin, 2024). This is

ISSN: 2181-3337 | SCIENTISTS.UZ

useful when several candidates have similar edit distance. A more advanced implementation may add a compact neural reranker, as recent multilingual systems combine contextual and neural signals (Sharma et al., 2023). The framework keeps these components modular so that an N-gram baseline and a neural model can be evaluated under the same candidate set.

For observed token u, sentence context x, and candidate $c \in C(u)$, the conceptual score is $S(c|u,x) = \alpha L(c,x) - \beta D(u,c) + \gamma M(c) + \eta E(u \rightarrow c)$. Here L is contextual log-likelihood, D is weighted edit cost, M is morphological plausibility, and E is error-channel log-likelihood. The four terms cannot be combined directly without scaling; the implementation protocol below specifies their transformations and fitting. The unchanged input u is always included in C(u).

Parameter estimation. Fit α, β, γ, and η with nonnegative constraints and $\alpha+\beta+\gamma+\eta=1$, using a fixed validation grid or nested validation optimization. Candidate-generation costs and feature scalers are learned on training data; ranking weights, confidence calibration, and decision thresholds are chosen on validation data. The primary selection objective is correction F1 subject to a prespecified upper bound on overcorrection of originally correct tokens; ties favor lower overcorrection and lower latency. No numerical weights are presented because annotated training and validation data have not yet been collected for this study.

Morphological ambiguity. Let A(c) contain all admissible analyses a=(stem, suffix sequence, features) of c. For each analysis, compute a nonnegative compatibility weight q(a|c) from lexical coverage, allowed suffix ordering, and optional contextual feature compatibility. Define $M(c)=\max_{a \in A(c)} q(a|c)$ as a transparent baseline, and compare it in ablation with a normalized log-sum-exp aggregation of analysis scores. If A(c) is empty, assign an explicit “unanalysed” state rather than categorically rejecting c. This prevents productive inflections, proper names, and borrowings from being automatically classified as mistakes. The choice between max and marginal aggregation is to be fixed using validation data, not test outcomes.

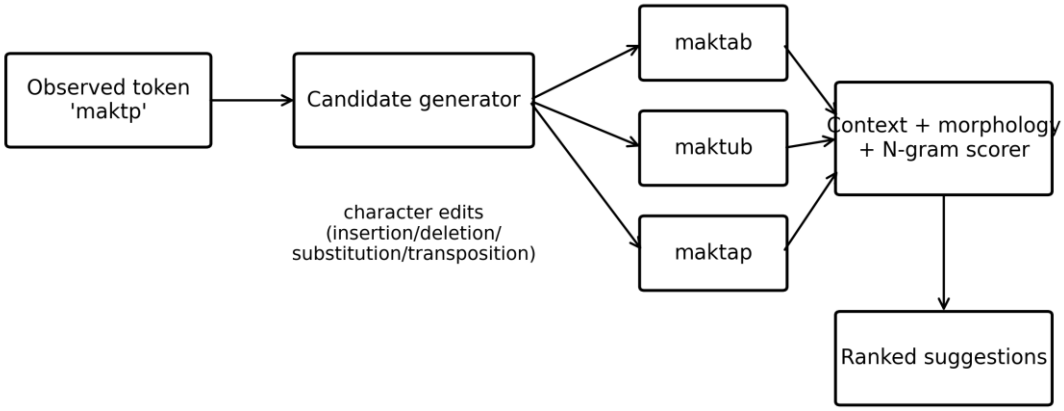
Score normalization and training protocol. Estimate L(c,x) as the smoothed sentence-level change in log probability relative to the unchanged sentence, using the same left/right context window for all candidates. Estimate D(u,c) as the minimum nonnegative weighted Damerau-style edit path divided by $\max(1, \max(|u|,|c|))$; edit costs are nonnegative and may be learned from aligned training errors with additive smoothing. Estimate $E(u \rightarrow c)$ from a smoothed character-confusion model, including an explicit identity transition. On the training partition only, fit a robust location and scale for each continuous feature (median and interquartile range, with a fixed fallback when the range is zero); clip transformed values to a prespecified interval such as $[-5,5]$. Store these training-derived transformations and apply them unchanged to validation and test. For M, map morphological evidence to [0,1], reserving a separate unknown-analysis indicator rather than assigning a zero probability to unseen names.

A generic candidate score can be written as: $S(c|u,x) = \alpha \cdot LM(c,x) - \beta \cdot Edit(u,c) + \gamma \cdot Morph(c) + \eta \cdot Err(u \rightarrow c)$, where LM(c,x) is contextual language-model evidence, Edit is a weighted character-edit cost, Morph is morphological plausibility, and Err is the estimated likelihood of the observed error transformation. The coefficients are trainable or validation-tuned parameters; no numerical values are claimed in this paper.

ISSN: 2181-3337 | SCIENTISTS.UZ

Figure 2

Candidate Generation and Contextual Ranking



Note. The example is illustrative, not an experimental output.

Uzbek-specific safeguards and novelty boundary. Apostrophe-like Unicode characters are mapped to a canonical internal representation while preserving raw offsets and an invertible mapping to the original text; orthographic normalization is never silently counted as lexical correction. Latin and Cyrillic inputs are tagged by script, and cross-script candidate generation is disabled by default unless explicitly requested. Candidate generation respects attested stem–suffix order and tests suffix-local edits independently from stem edits. Valid dictionary words are not exempt from contextual anomaly screening, whereas capitalized names, identifiers, URLs, and domain-specific terms receive a higher replacement threshold or are protected. The novelty asserted here is this jointly specified Uzbek-oriented decision policy and its evaluation plan, not the invention of edit distance, morphology, N-grams, or hybrid spelling correction themselves.

Acceptance, Abstention, and Output

Explicit correction decision. Let c^* be the highest-scoring non-identity candidate, and let $\Delta = S(c^*|u,x) - S(u|u,x)$. Obtain calibrated $p(\text{change}|u,x)$ from validation predictions (e.g., logistic or isotonic calibration, selected on validation only). Automatically replace u with c^* only if $\Delta \geq \tau_\text{margin}$ AND $p(\text{change}|u,x) \geq \tau_\text{conf}$ AND the token is not protected; otherwise return u unchanged, optionally with suggestions. The two thresholds are selected on validation data under the overcorrection constraint. The identity candidate and protected-span logic ensure that a plausible but unsupported replacement is not forced.

The highest-ranked correction should not automatically replace the input. The system should compare the best non-identity candidate against the unchanged form and apply a replacement only when confidence and margin thresholds are satisfied. Otherwise it should preserve the original token and optionally expose ranked suggestions. This mechanism is intended to reduce overcorrection of names, borrowings, rare terminology, and valid morphological forms.

Prospective Evaluation Protocol

Corpus Design

Annotation and partition protocol. Collect naturally occurring text with consent or appropriate reuse rights across edited prose, educational writing, news comments, and informal digital communication. Preserve source genre, script, raw token, intended correction(s), error type, and annotator decisions. Use two independent annotators for a subset and adjudicate disagreements; report agreement and ambiguous cases. Partition at document or author level where feasible (a provisional 80/10/10 train/validation/test

ISSN: 2181-3337 | SCIENTISTS.UZ

split), keeping near duplicates together. Synthetic character errors may supplement training only; natural held-out test data must remain untouched. Report corpus counts and proportions only after collection, not as projected achievements.

Controlled Comparisons and Ablations

Controlled comparisons. All systems must use the same held-out documents, tokenization, normalization policy, and allowed lexicon resources. Compare leave-unchanged, dictionary plus unit edit distance, dictionary plus morphology, edit candidates plus N-gram, and the complete model. Remove suffix constraints, contextual likelihood, learned confusion costs, and abstention separately. Add script-specific and real-word/non-word strata, and compare the max versus aggregated morphology variant. Fix tuning rules before test evaluation; confidence intervals should be computed by document-level bootstrap when sufficient test data are available.

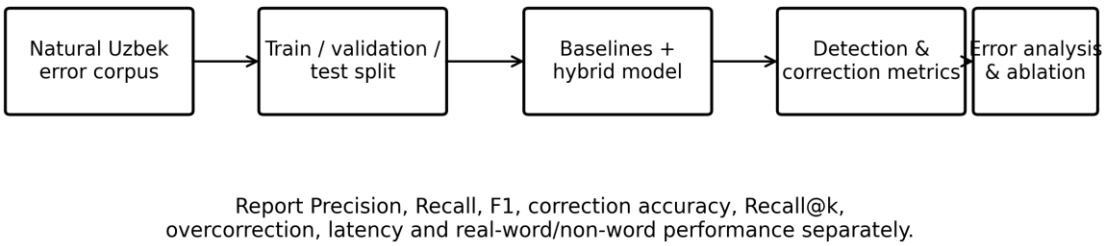
Metrics

Metric definitions. Candidate Recall@k is the proportion of erroneous tokens for which any adjudicated correct target occurs in the top k generated candidates. Detection precision=TP/(TP+FP), recall=TP/(TP+FN), and F1=2PR/(P+R), where a predicted erroneous span must match the annotation policy. Exact correction accuracy is the fraction of error instances replaced by an acceptable target; report it separately from unchanged-token accuracy. Overcorrection rate equals originally correct eligible tokens incorrectly changed divided by all originally correct eligible tokens. Report coverage (fraction auto-edited), selective accuracy at each coverage level, calibration error, mean latency per 1,000 tokens, and peak memory. Undefined ratios must be reported as not applicable rather than zero.

Detection should be evaluated with Precision, Recall, and F1 over erroneous spans. Correction should report exact correction accuracy and top-k candidate recall on error tokens rather than allowing the large number of already-correct tokens to dominate the score. Overcorrection—the proportion of originally correct tokens changed incorrectly—should be reported separately. Real-word and non-word errors should also be evaluated separately because they require different evidence. Runtime and memory should be recorded for practical deployment.

Figure 3

Proposed Experimental Workflow for Future Uzbek Spell-Checker Evaluation



Discussion and Limitations

This work is a conceptual and technical specification, not a completed empirical evaluation. Neither the 17 reviewed references nor the diagrams establish an accuracy advantage for the proposed Uzbek system. The literature synthesis is selective and lacks an executed systematic-search log; its reproducible expansion protocol is therefore prospective. Candidate normalization, morphological ambiguity treatment, constrained parameter fitting, and confidence thresholds are defined operationally but remain uncalibrated

ISSN: 2181-3337 | SCIENTISTS.UZ

until a labeled corpus exists. Consequently, any claim of improved accuracy, lower overcorrection, or cross-script robustness must await controlled experiments, held-out testing, and uncertainty reporting. The proposed novelty is limited to the integrated Uzbek-specific design choices and their falsifiable evaluation criteria.

The framework deliberately combines deterministic and statistical evidence because no single mechanism addresses all Uzbek spelling phenomena. Dictionary lookup is interpretable but brittle for productive morphology. Edit distance can recover nearby forms but cannot determine intended meaning. N-gram models add local context but suffer from sparse observations, especially for rare inflected forms. Morphological analysis reduces some sparsity but can itself be ambiguous. Neural contextual models may capture broader dependencies but introduce training-data, latency, and calibration requirements.

The literature on children's spell checking further shows that error distributions can be user-group specific and that phonetic information can improve candidate generation and ranking (Downs, 2020). Such evidence motivates a future Uzbek phonetic component, but this paper does not assume that findings from English-speaking children transfer directly to Uzbek writers. Likewise, multilingual search-query spell checking demonstrates scalable candidate/ranking architectures (Sharma et al., 2023), but query correction and full-sentence Uzbek writing are different evaluation settings.

No empirical accuracy values are reported here. The supplied sources support the component choices and known limitations, but they do not provide a directly comparable Uzbek benchmark on which the proposed complete architecture has already been tested. Any claim of superiority must therefore wait for implementation, corpus construction, controlled baselines, and held-out evaluation.

Conclusion

This paper specifies a morphology- and context-aware hybrid architecture for Uzbek spelling-error detection and correction. Beyond assembling known techniques, it defines reversible orthographic-sign normalization, script-sensitive candidate control, ambiguity-aware affix analysis, normalized multi-signal ranking, and confidence-based preservation of uncertain tokens. A transparent review rubric and prospective corpus, tuning, baseline, ablation, and metric protocols make the design independently testable. No implementation accuracy or superiority is claimed. The next step is to assemble and adjudicate a natural-error Uzbek corpus, implement the proposed variants, fit parameters exclusively on development data, and measure correction and overcorrection on a held-out test set.

This paper formulated an Uzbek-specific hybrid framework for automatic spelling-error detection and correction. The design connects the classical SPELL principles of dictionary lookup, affix handling, and reusable corrections with modern morphology-sensitive analysis, weighted edit-distance candidate generation, N-gram contextual ranking, statistical error evidence, and confidence-aware abstention. The framework distinguishes non-word and real-word errors and treats productive Uzbek morphology and proper names as sources of uncertainty rather than automatic error signals. Its staged architecture and evaluation protocol are intended to make future experiments interpretable: candidate generation, ranking, overcorrection, morphology, and context can be measured separately. The next research step is to build an annotated natural-error corpus for Uzbek, implement the baselines and hybrid variants, and evaluate them under identical data and resource conditions.

REFERENCES

1. Bhaire, V. V., Jadhav, A. A., Pashte, P. A., & Magdum, P. G. (2015). Spell checker. *International Journal of Scientific and Research Publications, 5*(4), 1–3. https://www.ijsrp.org/research-paper-0415/ijsrp-p4076.pdf
2. Downs, B. (2020). *Improving spellchecking for children: Correction and design* [Master's thesis, Boise State University]. Boise State University ScholarWorks. https://scholarworks.boisestate.edu/td/1721/

ISSN: 2181-3337 | SCIENTISTS.UZ

3. Elov, B. B., Adalı, E., Khamroeva, S. M., Abdullayeva, O. X., Xusainova, Z. Y., & Xudayberganov, N. U. (2023). The problem of POS tagging and stemming for agglutinative languages (Turkish, Uyghur, Uzbek languages). In *2023 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 57–62). IEEE. https://doi.org/10.1109/UBMK59864.2023.10286792
4. Elov, B. B., Hamroyeva, S. M., & Axmedova, X. I. (2023). Methods for creating a morphological analyzer. In H. Zaynidinov, M. Singh, U. S. Tiwary, & D. Singh (Eds.), *Intelligent human computer interaction* (Lecture Notes in Computer Science, Vol. 13741, pp. 27–38). Springer. https://doi.org/10.1007/978-3-031-27199-1_4
5. Elov, B. B., Khamroeva, S. M., & Xusainova, Z. Y. (2023). The pipeline processing of NLP. *E3S Web of Conferences, 413*, Article 03011. https://doi.org/10.1051/e3sconf/202341303011
6. Elov, B. B., Sirojiddinov, S. S., Khamroeva, S. M., Adalı, E., & Xusainova, Z. Y. (2023). Pos taging of Uzbek text using hidden Markov model. In *2023 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 63–68). IEEE. https://doi.org/10.1109/UBMK59864.2023.10286619
7. Elov, B. B., Xusainova, Z. Y., Umirova, S. M., Normamatov, S., Abdimurod kızı, A. S., & Mahmadiev, S. (2024). Algorithms for parsing roots and stems of words in Uzbek language. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*. IEEE. https://doi.org/10.1109/UBMK63289.2024.10773598
8. Gorin, R. E. (1974, December 14). *SPELL: Spelling check and correction program* [Program documentation]. Stanford Artificial Intelligence Laboratory. https://www.saildart.org/SPELL.REG%5BUP,DOC%5D3
9. Jurafsky, D., & Martin, J. H. (2024). *Speech and language processing* (3rd ed. draft, August 20, Chapter 3, “N-gram language models”). https://web.stanford.edu/~jurafsky/slp3/old_aug24/3.pdf
10. Kumar, R., Bala, M., & Sourabh, K. (2018). A study of spell checking techniques for Indian languages. *JK Research Journal in Mathematics and Computer Sciences, 1*(1), 105–113. https://jkhighereducation.nic.in/jkrjmcs/issue1/15.pdf
11. Lavoshnikova, E. K. (2015). Proverka orfografii v tekstovom redaktore Word 2013: Imena sobstvennye [Spell check in Word 2013: Proper names]. *Aktualnye voprosy obshchestvennykh nauk*, (8).
12. Lavoshnikova, E. K. (2017). “Problemnye” slova kak prichina propuska oshibok pri kompyuternoy proverke orfografii [“Problematic” words as a cause of missed errors in computer spell checking]. *Tekst. Kniga. Knigoizdanie*, (15), 113–129. https://doi.org/10.17223/23062061/15/8
13. Murodov, S. A., & Xazratqulov, S. Z. (2022). Spellchecker Windows ilovasi orqali o‘zbek tilidagi imloviy savodxonlikni tekshirish usullari [Methods for checking Uzbek spelling literacy through the Spellchecker Windows application]. *Academic Research in Educational Sciences, 3*(6), 364–369. https://cyberleninka.ru/article/n/spellchecker-windows-ilovasi-orqali-o-zbek-tilidagi-imloviy-savodxonlikni-tekshirish-usullari
14. Pirinen, T. A., & Lindén, K. (2010). Finite-state spell-checking with weighted language and error models: Building and evaluating spell-checkers with Wikipedia as corpus. In *Proceedings of LREC 2010: Workshop on creation and use of basic lexical resources for less-resourced languages*. https://helda.helsinki.fi/server/api/core/bitstreams/3d9eddbe-e4d7-4a39-9159-30d1566cafd3/content
15. Richter, M., Straňák, P., & Rosen, A. (2012). Korektor: A system for contextual spell-checking and diacritics completion. In *Proceedings of COLING 2012: Posters* (pp. 1019–1028). The COLING 2012 Organizing Committee. https://aclanthology.org/C12-2099/
16. Sharma, S., Valls-Vargas, J., King, T. H., Guerin, F., & Arora, C. (2023). Contextual multilingual spellchecker for user queries. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3395–3399). Association for Computing Machinery. https://doi.org/10.1145/3539618.3591861