International Journal of Innovative
Computing, Information and Control
Volume **21**, Number **6**, December **2025**

 ISSN 1349-4198
pp. **1741–1759**

ORTHOPHONOMATCH: INTEGRATING ORTHOGRAPHIC AND PHONOLOGICAL FEATURES FOR ENHANCED SPELL CORRECTION IN TONAL LANGUAGE SEARCH ENGINES

Passakorn Phannachitta and Chartchai Doung sa-ard

College of Arts, Media and Technology
Chiang Mai University
239 Suthep, Muang, Chiang Mai 50200, Thailand
{ passakorn.p; chartchai.d }@cmu.ac.th

Received March 2025; revised June 2025

Abstract. *Accurate correction of misspellings in catalog search engines is crucial for improving retrieval performance, especially in specialized domains with complex terminology and tonal language characteristics. However, conventional approaches often struggle with these linguistic challenges, resulting in reduced search effectiveness. To overcome these limitations, this study introduces OrthoPhonoMatch, a hybrid spell correction algorithm designed for Thai-language industrial catalog search. OrthoPhonoMatch employs a two-stage tokenization process (word-to-syllable) and generates hybrid representations combining orthographic syllables and their phonological International Phonetic Alphabet (IPA) representations. Indexing and searching leverage these hybrid representations along with precomputed Levenshtein similarity within a TF-IDF weighting scheme to efficiently retrieve error-tolerant matches. The evaluation was conducted using 425,729 queries derived from Autopair's product database, a well-known Thai automotive aftermarket platform where query misspellings negatively impact auto-parts procurement and workshop management. Compared with eight baseline spell correction techniques, OrthoPhonoMatch consistently achieved superior performance across key retrieval metrics, including Precision@k, Recall@k, AP@k, and NDCG@k, with k values up to 20, confirmed through robust statistical validation. Notably, OrthoPhonoMatch demonstrated strong results in challenging multi-error queries, where its Precision@1, AP@1, and NDCG@1 scores (representing critical top-rank performance) surpassed the best scores of competing techniques. These findings highlight the effectiveness of integrating syllable-level orthographic and phonological features for robust spell correction in high-error industrial catalog searches.*

Keywords: Catalog search engines, Spell correction, Orthographic similarity, Phonological similarity, Tonal languages, Industrial applications

1. **Introduction.** Full-text search in catalog search engines is a fundamental tool for retrieving domain-specific information, such as retail product catalogs and healthcare records, by supporting flexible user queries [1]. However, specialized search environments present distinct challenges, especially when handling domain-specific terminology and spelling inconsistencies [2]. In online marketplaces, for example, proper nouns such as product brands and model names frequently appear across multiple records, complicating the effectiveness of conventional frequency-based relevance models in retrieval assessment. These challenges become even more significant in tonal languages such as Thai, where typographical errors and tonal variations increase query complexity, leading to a reduction in retrieval effectiveness [3, 4].

DOI: 10.24507/ijicic.21.06.1741

This study proposes OrthoPhonoMatch, a novel algorithm specifically designed for Thai-language catalog search queries in industrial applications. OrthoPhonoMatch addresses challenges caused by misspellings in tonal languages through a syllable-level orthographic and phonological matching strategy. The key hypothesis of OrthoPhonoMatch is that users in industrial search environments tend to retain partial recall of a word's pronunciation or individual characters when making typographical errors. By integrating orthographic and phonological representations, OrthoPhonoMatch enables partial query-document matching, improving robustness and retrieval performance in industrial catalog searches.

To evaluate its effectiveness, OrthoPhonoMatch was tested on 425,729 misspelled Thai-language queries derived from Autopair, a major Thai automotive aftermarket platform, where query misspellings negatively impact search efficiency in auto-parts ordering and workshop management systems [5]. The assessment framework consisted of six query scenarios, each structured to reflect distinct homophonic and omission error distributions, the two most frequent misspelling patterns identified by Autopair. The number of spelling errors per query ranged from one to three, ensuring representative error distributions and including challenging cases, particularly those with three errors per query. Performance was assessed across four standard retrieval performance metrics (Precision@k, Recall@k, AP@k, and NDCG@k) [6]. Statistical significance was evaluated through win/tie/loss analysis [7] and the robust percentile bootstrap test with a modified one-step M-estimator [8]. The results, which were statistically significant, confirmed that OrthoPhonoMatch consistently outperformed competing techniques across all evaluated metrics, particularly for queries with higher error rates. Further analysis showed that when decomposed into Ortho and Phono components, neither alone matched the performance of the combined OrthoPhonoMatch. This finding reinforces the importance of integrating both components for effective catalog search engine spell correction.

This paper is structured as follows. Section 2 presents related work on search engines, spell correction, and language-specific challenges motivating this study. Section 3 describes the key components and principles of the OrthoPhonoMatch algorithm. Section 4 presents the evaluation methodology. Section 5 reports experimental results, and Section 6 discusses findings, component analysis, generalizability, and potential threats to validity. Finally, Section 7 summarizes key contributions and outlines directions for future research.

2. **Related Work.**

2.1. **Search engines and catalog search engines.** Conventional search engines retrieve and rank information from large-scale, unstructured data sources, such as web pages [9]. These systems employ core Information Retrieval (IR) techniques, including data acquisition, indexing, query processing, and ranking [6, 10]. Data acquisition involves extracting and preprocessing raw content from web crawlers or structured databases. Indexing structures this information using inverted indexes, which map terms to documents while maintaining Term Frequency (TF) and Inverse Document Frequency (IDF) statistics [11]. TF-IDF scoring determines term importance by emphasizing words that frequently appear within a document but infrequently across the corpus. Query processing maps user queries to indexed terms, and ranking algorithms prioritize search results based on TF-IDF scores or advanced ranking techniques, such as PageRank [12] and learning to rank [13].

A catalog search engine facilitates structured data retrieval in domain-specific repositories, such as retail product catalogs and technical databases [1]. While similar to

conventional search engines, catalog search presents distinct challenges [14]. It processes structured product attributes, such as brand, model, and specifications, while supporting attribute-based queries. Unlike web search, catalog search relies on proper nouns from specialized domain vocabularies, which often lack contextual cues necessary for effective relevance ranking. Additionally, users frequently submit non-exact queries that do not align with predefined catalog terminology. Traditional ranking models like TF-IDF assume that term frequency and term rarity correlate with relevance. This assumption, combined with non-exact user queries, often leads to retrieval inefficiencies and suboptimal ranking performance [15].

2.2. **Spell correction in search queries.** Typographical errors in search queries present a major challenge, particularly in structured catalog search, where domain-specific terminology provides limited contextual cues for error-tolerant retrieval. User spelling variations and transliteration inconsistencies contribute to query mismatches, negatively affecting retrieval effectiveness. These challenges necessitate robust spelling correction techniques to enhance retrieval performance in structured search applications [16].

Spell correction enhances retrieval effectiveness in modern search engines [17]. Foundational approaches often use edit distance, a string similarity measure quantifying the minimum character edits (insertions, deletions, substitutions) required to transform one string into another [18]. Common variants, such as Hamming and Levenshtein distance, provide structured similarity measurements [19] and are frequently used to rank candidate corrections by similarity or other heuristics. Practical algorithms often combine methods; dictionary-based approaches compare misspellings against predefined lexicons [17], while morphological analysis examines word structures (e.g., prefixes and suffixes) [18].

Recent advancements employ large-scale deep learning models, commonly adopted by major search engines, to analyze query context using neural networks for complex error correction [20]. In domain-specific settings like e-commerce, Retrieval-Augmented Generation (RAG) frameworks integrate language models with external knowledge bases to improve correction accuracy, especially for specialized vocabularies such as brand names [21]. While powerful, these methods typically require substantial data and computational resources, potentially limiting their applicability in specialized or lower-resource environments such as Thai catalog search. Efficiency is another critical consideration. For example, Xuan et al. [22] proposed hashing techniques for fast document retrieval. Although their focus differs from term-level spell correction, their study illustrates ongoing efforts to optimize retrieval at scale.

Despite these varied approaches, significant challenges remain, especially for tonal languages. Conventional techniques, primarily relying on character similarity and linguistic rules, fundamentally struggle when phonological variations impact spelling accuracy. In such languages, minor tone or phonetic shifts can produce entirely different meanings, making purely orthographic correction ineffective and highlighting the need for phonologically aware methods.

2.3. **Phonological challenges in tonal languages.** Tonal languages such as Thai present unique challenges for spell correction systems due to their tone-dependent semantics. For instance, the Thai syllable /kʰâːw/ carries distinct meanings depending on tonal variation: ข้าว (/kʰâːw/) refers to rice, whereas ข่าว (/kʰǎːw/) denotes news. Modifying the initial consonant sound, as from /kʰâːw/ to /gâːw/, results in further semantic shifts: ข้าว (/gâːw/) means stepping forward, while กาว (/gaːw/) signifies glue. Conventional approaches relying solely on character similarity metrics, such as edit distance, lack phonological modeling, making them ineffective for resolving these tonal ambiguities.

Phonetic representations have been widely explored to improve retrieval accuracy and spell correction in tonal languages [23, 24, 25]. One commonly used approach is phonetic encoding, such as Soundex, which groups words with similar pronunciations under the same code to facilitate approximate matching. However, Soundex and similar encoding schemes compress phonetic structures into fixed-length numeric representations, limiting their ability to capture subtle tonal variations.

To address these limitations, alternative phonetic transcription systems, such as the International Phonetic Alphabet (IPA) [26], provide finer-grained phonological representations. Unlike text-based edit distance models, IPA-based phonological similarity can account for subtle pronunciation differences, enabling improved similarity assessments. For example, comparing fish (/fɪʃ/) and fitch (/fɪtʃ/) highlights the improved similarity assessment enabled by IPA. While the orthographic edit distance is two, the IPA edit distance is one, demonstrating the potential of phonological modeling.

Despite their advantages, phonetic representations alone often prove insufficient. Ambiguities related to homophones, silent letters, and subtle phonetic variations persist. This suggests effective similarity assessment requires integrating both orthographic and phonological features to accurately capture linguistic distinctions.

2.4. **Recent advances in Thai Natural Language Processing (NLP).** Given the complexities of the Thai script, such as multi-level characters and the absence of explicit word boundaries [27], substantial effort has been directed toward fundamental tasks like word segmentation and developing language-specific resources [28]. For Thai spell correction, recent studies have applied deep learning methods, including sequence-to-sequence models with Bi-LSTM architectures [29] or fine-tuning large pre-trained models [30]. Beyond spell correction, Palahan [31] focused on improving Thai document retrieval in domain-specific settings, i.e., trade information, by combining traditional IR models with deep learning techniques. These studies illustrate the increasing adoption of modern approaches to enhance Thai-language information access. While deep learning approaches show promise for Thai NLP, they typically require substantial annotated training data, which may not always be available for domain-specific settings like industrial catalogs.

In summary, this literature review identifies two key limitations: 1) conventional spell correction methods are not well-suited for tonal languages, and 2) recent deep learning approaches often require high computational resources and large annotated datasets.

3. **OrthoPhonoMatch.** To overcome the limitations identified in the literature review, this study proposes OrthoPhonoMatch, a novel hybrid spell correction algorithm designed specifically for tonal language catalog search. By incorporating orthographic and phonological representations into the similarity assessment process at the syllable level, OrthoPhonoMatch improves query correction, particularly for specialized vocabulary and complex tonal structures. Figure 1 provides a visual overview of the entire process, illustrating the core components and their interactions. As shown in the flowchart, the core data processing steps are applied to both input documents prior to indexing and input queries prior to searching. The following sections present these main components of OrthoPhonoMatch in detail: (i) the common data processing pipeline (Section 3.1) and (ii) the subsequent indexing and searching procedures (Section 3.2).

3.1. **Data processing.** OrthoPhonoMatch employs a two-stage tokenization process, where each query is divided into syllable sequences. First, a word tokenizer processes the input by segmenting it into individual words, which are then further decomposed into syllables. Empirical results show that syllable-level tokenization improves accuracy for individual words compared with sentence-level processing. To ensure a standardized

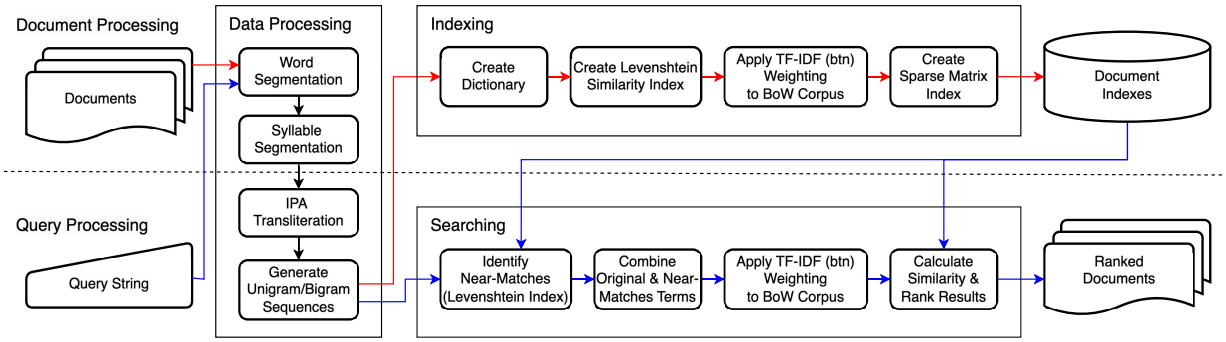

FIGURE 1. Overview of the OrthoPhonoMatch algorithm, illustrating the data processing pipeline, and indexing and searching procedures

phonetic representation, syllables containing non-alphabetic characters are transliterated into the International Phonetic Alphabet (IPA). This unified framework enables syllables, as minimal linguistic units encapsulating both orthographic and phonological properties, to serve as the basis for similarity assessment.

Subsequently, OrthoPhonoMatch generates unigram and bigram sequences from tokenized syllables, which are merged into arrays for efficient processing. The unigram array directly replicates the syllable sequence. In contrast, the bigram array is constructed from overlapping pairs of syllables, with a padding symbol (e.g., an _) inserted at the beginning and end of words to indicate initial and final syllable positions.

Table 1 presents sample records from the Autopair product database [5] to illustrate OrthoPhonoMatch's practical application. This database contains structured information about automotive components, brands, models, and related details used in this study. English translations for the Thai terms presented are provided in footnotes to aid international readers.

TABLE 1. Example records from a product database provided by Autopair [5]. English translations are provided in table notes.

SKU	Part name	Part brand	Fitment	Car brand	Car model	Car details
EX0001	โช้คอัพ[1]	KYB	หน้า[4]	ฮอนด้า[6]	แอคคอร์ด[8]	โฉมปี[11] 2003-2007 (G7)
EX0002	โช้คอัพ[1]	KYB	หลัง[5]	ฮอนด้า[6]	แอคคอร์ด[8]	โฉมปี[11] 2003-2007 (G7)
EX0003	สายพานไดร์ชาร์จ[2]	มิซูโบชิ[3]	–	โตโยต้า[7]	อแวนซ่า[9]	โฉมปี[11] 2004-2011 (F601, F602)
EX0003	สายพานไดร์ชาร์จ[2]	มิซูโบชิ[3]	–	โตโยต้า[7]	อแวนซ่า[9]	โฉมปี[11] 2012-2016 (F651, F652)
EX0003	สายพานไดร์ชาร์จ[2]	มิซูโบชิ[3]	–	โตโยต้า[7]	โคโรลล่า อัลติส[10]	โฉมปี[11] 2001-2007 (ZZE121)
EX0003	สายพานไดร์ชาร์จ[2]	มิซูโบชิ[3]	–	โตโยต้า[7]	โคโรลล่า อัลติส[10]	โฉมปี[11] 2008-2012 (ZZE141)

Translations: 1. โช้คอัพ = Suspension; 2. สายพานไดร์ชาร์จ = Alternator belt; 3. มิซูโบชิ = Mitsuboshi; 4. หน้า = Front; 5. หลัง = Back; 6. ฮอนด้า = Honda; 7. โตโยต้า = Toyota; 8. แอคคอร์ด = Accord; 9. อแวนซ่า = Avanza; 10. โคโรลล่า อัลติส = Corolla Altis; 11. โฉมปี = Year model. Parenthesized codes refer to specific vehicle generations or sub models.

For example, consider the input term สายพานไดร์ชาร์จ (Alternator belt; approximately pronounced as sai pan dai chat). Table 2 presents the transformation steps of this example along with an explanation of each step. By applying these processing rules, each record is encoded into a hybrid orthographic-phonological array (i.e., composed of concatenated unigram and bigram sequences), forming the document collection for subsequent indexing and facilitating robust error-tolerant matching. To enhance probabilistic inference during candidate term ranking for misspelled queries, redundant unigram and bigram terms are removed from the arrays. Finally, each array is padded with unique, non-matching symbol

TABLE 2. Example of data processing steps for the input term สายพานไดร์ชาร์จ (Alternator belt)

Step	Explanation	Transformation
Word segmentation	Divide the text into individual words	สายพาน, ไดร์ชาร์จ
Syllable segmentation	Further segment words into syllables	สาย, พาน, ไดร์, and ชาร์จ
IPA transliteration	Convert syllables into IPA notation	/saːj/, /pʰaːn/, /daj/, and /t͡ɕʰaːt/
Unigram construction	Store syllables as independent units	[สาย, พาน, ไดร์, ชาร์จ, /saːj/, /pʰaːn/, /daj/, /t͡ɕʰaːt/]
Bigram construction	Pair adjacent syllables with padding markers	[_สาย, สาย_พาน, พาน_, _ไดร์, ไดร์_ชาร์จ, ชาร์จ_, _/saːj/, /saːj/_/pʰaːn/, /pʰaːn/_, _/daj/, /daj/_/t͡ɕʰaːt/, /t͡ɕʰaːt/_]
Final representation	Combine orthographic and phonological features	[สาย, พาน, ไดร์, ชาร์จ, _สาย, สาย_พาน, พาน_, _ไดร์, ไดร์_ชาร์จ, ชาร์จ_, /saːj/, /pʰaːn/, /daj/, /t͡ɕʰaːt/, _/saːj/, /saːj/_/pʰaːn/, /pʰaːn/_, _/daj/, /daj/_/t͡ɕʰaːt/, /t͡ɕʰaːt/_]

sequences (e.g., [!#, !$, !%]) to ensure uniform term weighting within documents without affecting similarity calculations during search.

To ensure reproducibility, tools from the `PyThaiNLP` Python library (version 4.0.2) were used for both tokenization and transliteration, with default settings applied as specified in the official documentation [27].

3.2. **Indexing and searching.** Term weighting in OrthoPhonoMatch is based on the SMART Information Retrieval System's convention [32]. This convention defines various schemes using a `ddd.qqq` notation, where `ddd` represents the document vector configuration and `qqq` represents the query vector configuration; each three-character segment specifies the weighting configuration for TF, IDF, and document length normalization, respectively.

Specifically, OrthoPhonoMatch employs the `btn.btn` scheme for indexing and searching. This configuration uses a binary representation for TF (`b`), adopts the standard IDF calculation (`t`), and applies no document length normalization (`n`). Using binary TF is integral to OrthoPhonoMatch's design. Combined with special character padding introduced during data processing, it ensures that each syllable's contribution is distinctly represented, thereby enhancing indexing stability. This approach is particularly suitable for catalog search environments, where raw term frequency may not accurately reflect a term's importance within specialized product records.

To index documents, which have been transformed into arrays of padded unigram and bigram sequences, OrthoPhonoMatch utilizes the `Gensim` [33] Python library (version 4.3.1) and performs the following operations.

(i) The arrays are converted into a `Gensim` dictionary object.
(ii) Levenshtein similarity, with a maximum distance of two, is precomputed for each term in the dictionary using `Gensim`'s built-in `LevenshteinSimilarityIndex`, facilitating efficient near-match lookup.
(iii) Each document in the dictionary object is converted into a Bag-of-Words (BoW) representation using the `doc2bow` method.
(iv) A TF-IDF model is trained on the BoW corpus using the `TfidfModel` class with the `btn` weighting scheme, ensuring a consistent weighted term representation for queries.

(v) A sparse matrix representing the weighted term-document relationships is generated within the `TfidfModel` class for efficient similarity calculations using the `SparseMatrixSimilarity` class.

The resulting models and indices are applied in the search process. Query processing follows the index processing steps to ensure consistent term representation. The search process is performed as follows.

(i) The query undergoes two-stage tokenization and is converted into an array of padded unigram and bigram sequences.
(ii) The Levenshtein similarity index identifies terms with minor spelling variations within a distance of two from the query.
(iii) The outputs of steps (i) and (ii) are concatenated and converted into a BoW representation.
(iv) The TF-IDF model generated in the indexing phase is applied to this BoW representation to transform each term into a TF-IDF weight.
(v) The TF-IDF weighted query is efficiently compared against indexed documents using the sparse matrix in order to calculate similarity scores.
(vi) The similarity scores between the processed query and the indexed documents are ranked, and the top-ranking documents are returned to the user.

4. **Evaluation Methodologies.** A structured evaluation framework was developed to assess the performance of OrthoPhonoMatch and compare it with established spell correction techniques. The framework comprised four key phases: (i) Generation of experimental datasets designed to capture the complexities of real-world search scenarios; (ii) Implementation of spell correction techniques within a catalog search environment to enable direct performance comparison; (iii) Measurement of retrieval effectiveness using a comprehensive suite of metrics; (iv) Statistical validation of results through win/tie/loss analysis and the robust percentile bootstrap test with a modified one-step M-estimator.

4.1. **Generation of experimental datasets.** The evaluation was conducted using query datasets designed to reflect real-world search behavior within the Autopair product database. These queries typically included a part name, a car brand, and a car model, providing insights into actual user search patterns. Since Thai lacks explicit spaces to separate words, query terms frequently appear concatenated, such as in โช้คอัพฮอนด้าแอคคอร์ด, which represents a search for Honda Accord suspension parts. This absence of word boundaries poses additional challenges in detecting and correcting erroneous queries.

Autopair identified two primary types of errors in these queries.

- **Homophonic errors** commonly arise from the mixture of Thai and transliterated foreign terms. For instance, in โช้คอัพฮอนด้าแอคคอร์ด, the term โช้คอัพ originates from Thai, whereas ฮอนด้า (Honda) and แอคคอร์ด (Accord) are transliterations.
- **Omission errors** frequently occur due to the use of virtual keyboards, which tend to introduce a higher rate of input errors compared with physical keyboards.

A subset of queries was sampled from the Cartesian product of 117 part names, 357 car models, and 32 car brands in Autopair's product database, which contains 13,469 SKUs. Queries were selected based on a minimum threshold of ten relevant SKUs per query to mitigate bias arising from low-relevance counts when evaluating spell correction techniques.

Table 3 presents the six generated query datasets, consisting of a total of 425,729 queries. Each dataset is named following the format [Error Type]–[Number of Terms]–[Number of Erroneous Terms]. For instance, H–3–2 denotes all possible queries generated from three terms (part name, car brand, and car model), each containing two homophonic

TABLE 3. Characteristics of the generated query datasets

Dataset name	Error type	#Terms	#Erroneous terms	#Queries	#Relevant SKUs per query	#Characters per query
					Mean (SD)	Mean (SD)
H–3–1	homophonic	3	1	7,904	15.33 (6.50)	27.31 (4.91)
H–3–2	homophonic	3	2	67,885	14.49 (5.86)	28.33 (4.83)
H–3–3	homophonic	3	3	173,525	13.35 (4.89)	29.23 (4.71)
O–3–1	omission	3	1	6,708	15.92 (7.09)	26.29 (5.34)
O–3–2	omission	3	2	49,710	15.74 (7.01)	26.38 (5.74)
O–3–3	omission	3	3	119,997	15.56 (6.89)	26.40 (6.05)
			Total	425,729		

errors from any two of the three terms. An example from this dataset is โช้คอัปฮอลด้าแอคคอร์ด, which corresponds to โช้คอัพฮอนด้าแอคคอร์ด (Honda Accord suspension).

4.2. **Implementation of spell correction techniques within a catalog search engine.** OrthoPhonoMatch was evaluated against eight spell correction configurations, formed by combining four spell correction algorithms with two tokenization methods. These configurations are denoted as x_y, where x represents the algorithm and y represents the tokenizer. The selected algorithms are specifically designed for search environments with limited contextual information, a defining characteristic of catalog search engines. In contrast to deep learning and BERT-based methods, which rely on large, context-rich corpora for optimal performance, these algorithms are well-suited for handling low-context queries in structured databases [34].

In Thai NLP, the absence of explicit word boundaries necessitates specialized tokenization methods, as segmentation plays a critical role in spell correction effectiveness [28]. To evaluate this impact, two tokenization methods were assessed within the catalog search context, providing a comprehensive comparison of their influence on retrieval accuracy.

4.2.1. *Spell correction algorithms.* This study evaluates four spell correction algorithms: Hunspell [35], Norvig [36], SymSpell [37], and Unsupervised FastText [38, 39]. Their implementations were obtained from the following Python libraries in the PyPI repository: `spylls` [40] (version 0.1.7) for Hunspell, `PyThaiNLP` [27] (version 4.0.2) for Norvig and SymSpell, and `fastText` [41] (version 0.9.2) for Unsupervised FastText. A brief overview of each algorithm is provided below.

- **Hunspell**: A dictionary-based spell checker that incorporates morphological analysis. It applies affix rules to modeling word variations, enabling efficient correction of affix-related errors. Widely adopted in operating systems and text editors, Hunspell is optimized for real-time spell checking.
- **Norvig**: A probabilistic spell correction algorithm that generates candidate corrections based on word frequency and common error patterns. It utilizes a language model to rank corrections, making it adaptable to diverse linguistic structures.
- **SymSpell**: A high-speed spell correction algorithm based on Damerau-Levenshtein distance [42]. By precomputing minimal edit distances, SymSpell enables rapid lookup of candidate corrections, making it highly efficient for large-scale text processing.
- **Unsupervised FastText**: A word embedding model that represents words as subword units, effectively capturing morphological structures [39]. This method is particularly well-suited for handling complex morphology and low-resource languages.

FastText supports both Continuous Bag-of-Words (CBOW) and skip-gram models, with skip-gram selected in this study due to its empirically superior performance in word representation.

4.2.2. *Word tokenization techniques.* This study evaluates two word tokenization methods: DeepCut [43] and the Multi-cut Maximum Matching tokenizer (NewMM) [27]. DeepCut is a machine learning-based tokenizer that leverages Convolutional Neural Networks (CNNs), whereas NewMM is a rule-based tokenizer. Both methods are implemented in the `PyThaiNLP` Python library, with NewMM serving as the default tokenizer, highlighting its broad applicability in Thai text processing tasks.

- **DeepCut**: A CNN-based Thai word tokenizer trained on large-scale datasets, DeepCut addresses segmentation challenges in Thai text, particularly ambiguity resulting from the absence of whitespace between words. Its performance is influenced by the training data and model architecture, making it sensitive to domain-specific vocabulary.
- **NewMM**: A rule-based Thai tokenizer that integrates dictionary-based Maximum Matching with Thai Character Cluster (TCC) analysis [44]. It segments text by matching dictionary entries while applying TCC rules to refining word boundaries. NewMM is designed for efficiency and general-purpose Thai word tokenization, achieving a balance between speed and accuracy.

4.3. **Performance measurement.** In IR, search engine performance is assessed using metrics that measure accuracy, sensitivity, and graded relevance within the top-k retrieved results [6, 11]. This study employs four widely used evaluation metrics.

- **Mean Precision@k**: It measures the proportion of relevant entries within the top-k retrieved results, averaged across all queries. A higher Mean Precision@k indicates that the search engine retrieves a greater number of relevant documents at higher ranks.

$$\text{Mean Precision@}k = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\sum_{i=1}^{k} rel_{qi}}{k}, \tag{1}$$

  where Q is the total number of queries, and rel_{qi} denotes the relevance of document i for query q (0 for irrelevant and 1 for relevant).
- **Mean Recall@k**: It measures the proportion of all relevant documents retrieved within the top-k results, reflecting the sensitivity of the search engine.

$$\text{Mean Recall@}k = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\sum_{i=1}^{k} rel_{qi}}{R_q}, \tag{2}$$

  where R_q represents the total number of relevant documents for query q.
- **Mean Average Precision@k (Mean AP@k)**: It measures retrieval performance by considering precision at the rank position of each relevant document.

$$\begin{aligned} \text{Mean AP@}k &= \frac{1}{Q}\sum_{q=1}^{Q} AP_q@k, \text{ where} \\ AP_q@k &= \frac{1}{R_{q@k}}\sum_{i=1}^{k}\left(\frac{rel_{qi}}{i} \times \text{Precision@}i\right), \end{aligned} \tag{3}$$

  where $R_{q@k}$ represents the total number of relevant documents within the top-k ranks, and Precision@i denotes the precision at rank i.

- **Mean Normalized Discounted Cumulative Gain@k (Mean NDCG@k)**: It measures graded relevance while prioritizing higher-ranked relevant documents.

$$\text{Mean NDCG@}k = \frac{1}{Q}\sum_{q=1}^{Q}\frac{DCG_q@k}{IDCG_q@k}, \text{ where}$$

$$DCG_q@k = \sum_{i=1}^{k}\frac{2^{rel_{qi}}-1}{\log_2(i+1)}, \text{ and} \tag{4}$$

$$IDCG_q@k = \sum_{i=1}^{R_{q@k}}\frac{2^{rel_{qi}}-1}{\log_2(i+1)},$$

  where rel_{qi} represents graded relevance (e.g., 0 for irrelevant, 1 for partially relevant, and 2 for highly relevant). IDCG@k denotes the ideal ranking with perfect relevance order, ensuring that NDCG@k values remain bounded between 0 and 1.

These four metrics, applied at different k levels, offer a comprehensive evaluation of spell correction performance by accounting for both retrieval accuracy and completeness. Analyzing these metrics across various cut-off points (k values) enables a detailed assessment of each technique's effectiveness in retrieving and ranking relevant results.

4.4. **Statistical validation.** Various statistical tests have been explored in IR; however, most fail to provide reliable significance estimates. One of the primary challenges is that bounded and discrete metrics, such as Precision@k, generate a limited set of possible outcomes, particularly for small k values. For instance, Precision@5 can only produce values of 0/5, 1/5, ..., 5/5, leading to non-normal distributions. This discreteness, combined with identical results across multiple queries, violates the normality assumption required by conventional tests, such as Student's t-test, making them unsuitable [45].

Smucker et al. [46] demonstrated that, when applied to IR evaluation, the signed test and Wilcoxon signed-rank test often produce unstable results, potentially leading to misleading conclusions. Although bootstrap methods were recommended, computational constraints at the time limited their practical feasibility [6]. In empirical software engineering [47, 48], where small sample sizes and non-normal distributions are common, a combination of win/tie/loss statistics and rank-based tests (e.g., Wilcoxon rank-sum and Brunner test) has been proposed for method comparisons [8]. However, these tests assume continuous distributions, an assumption violated in this study due to the discrete nature of Precision@k and other retrieval metrics.

Given these constraints, this study employed a parallel computing framework and adopted bootstrapping methods in conjunction with win/tie/loss statistics. The robust percentile bootstrap test with a modified one-step M-estimator [8] was integrated with win/tie/loss statistics to enable reliable comparisons across techniques. This methodology utilizes resampled distributions for statistical comparisons, with the modified one-step M-estimator mitigating the impact of heavy-tailed distributions and outliers. Compared with traditional methods, this approach enhances statistical power and reduces Type I error rates across diverse data distributions commonly observed in IR.

Based on the win/tie/loss statistics framework, each spell correction technique is evaluated using three counters: *win*, *tie*, and *loss* [49, 50], which are incremented through pairwise comparisons. When comparing techniques i and j on metric m (e.g., Precision@5), a statistically significant difference, as determined by the robust percentile bootstrap test with a modified one-step M-estimator ($p < 0.05$), results in the superior technique receiving a win increment (win_i) and the inferior technique receiving a loss increment ($loss_j$). If no significant difference is detected, both techniques receive a *tie* increment. This process

is repeated across all metric pairs (m_i and m_j) and k values up to 20. Techniques with the highest win-minus-loss difference are considered the most effective.

The `rmmest` function from the `wrs` R package [51] (version 0.24) was used to implement the robust percentile bootstrap test with a modified one-step M-estimator. The test configuration included enabling bootstrapping (`BA=True`), setting 500 bootstrap replicates (`nboot=500`), using the measure of location associated with marginal distributions rather than difference scores (`dif=False`), and applying a modified one-step M-estimator to controlling the probability of Type I errors (`est=MOM`).

5. **Results.**

5.1. **Homophonic errors.**

Table 4 presents the comparative performance of nine spell correction techniques across three test cases with varying query lengths and proportions of homophonic errors. Performance was assessed using the win-minus-loss score, computed from four evaluation metrics: Mean Precision@k, Mean Recall@k, Mean AP@k, and Mean NDCG@k, each measured at k values of 5, 10, and 20. A higher win-minus-loss score indicates superior overall performance, while a positive score signifies that a technique outperformed others more frequently than it was outperformed.

TABLE 4. Comparison of spell correction techniques for **homophonic errors**, evaluated using aggregated win-minus-loss scores across datasets H–3–1, H–3–2, and H–3–3. Higher scores indicate superior performance.

Rank	Techniques	#win-minus-loss of Mean				Total
		Precision@ 5, 10, 20	Recall@ 5, 10, 20	AP@ 5, 10, 20	NDCG@ 5, 10, 20	
1	OrthoPhonoMatch	39	27	55	36	157
2	$\text{Norvig}_{\text{NewMM}}$	29	25	47	23	124
3	$\text{SymSpell}_{\text{NewMM}}$	18	17	33	15	83
4	$\text{Hunspell}_{\text{NewMM}}$	18	9	21	16	64
5	$\text{Norvig}_{\text{DeepCut}}$	−3	−3	−6	−7	−19
6	$\text{Hunspell}_{\text{DeepCut}}$	−8	0	−10	−6	−24
7	$\text{SymSpell}_{\text{DeepCut}}$	−17	−13	−22	−18	−70
8	Unsupervised $\text{FastText}_{\text{NewMM}}$	−38	−31	−59	−21	−149
9	Unsupervised $\text{FastText}_{\text{DeepCut}}$	−38	−31	−59	−38	−166

As shown in the table, OrthoPhonoMatch consistently outperformed the eight baseline spell correction techniques evaluated, achieving a total win-minus-loss score of 157, with no negative scores across any metric. Following OrthoPhonoMatch, $\text{Norvig}_{\text{NewMM}}$ (124), $\text{SymSpell}_{\text{NewMM}}$ (83), and $\text{Hunspell}_{\text{NewMM}}$ (64) also demonstrated strong performance. When analyzed by individual metrics, no technique outperformed OrthoPhonoMatch on any evaluation metric.

While the win-minus-loss scores provide an overall ranking, they may not fully capture performance variations across specific queries or evaluation metrics. To address this, Figure 2 presents a detailed comparison across all four evaluation metrics at k values from 1 to 20, focusing on techniques with positive win-minus-loss scores from Table 4, which are OrthoPhonoMatch, $\text{Norvig}_{\text{NewMM}}$, $\text{SymSpell}_{\text{NewMM}}$, and $\text{Hunspell}_{\text{NewMM}}$.

In Figure 2, the x-axis represents k values (1 to 20), while the y-axis denotes performance scores, where all metrics have a maximum possible value of 1. Columns correspond to evaluation metrics, and rows represent datasets. The results clearly demonstrate

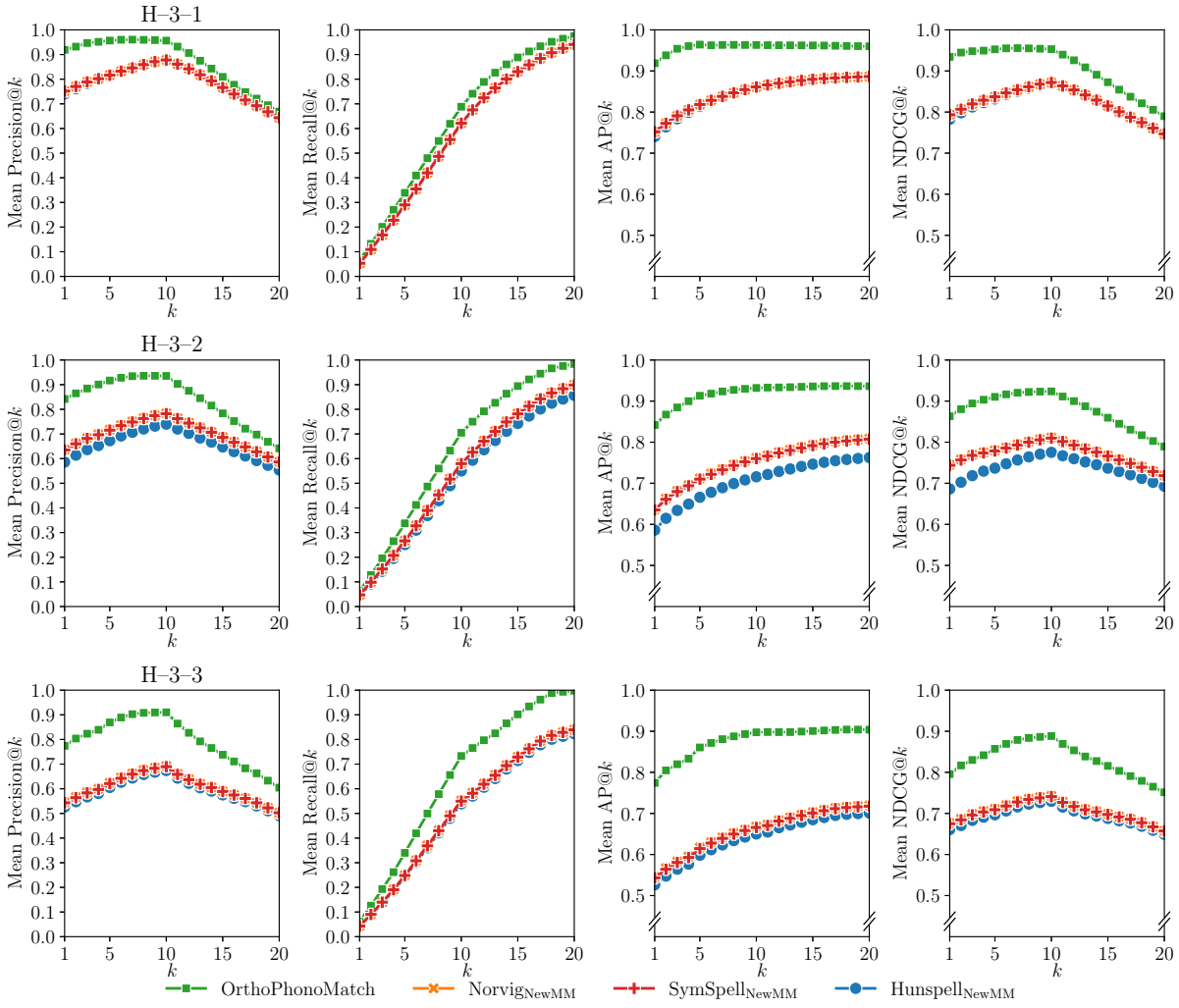

FIGURE 2. Performance of spell correction techniques with positive win-minus-loss scores in addressing **homophonic errors**

OrthoPhonoMatch's overall superiority, with its performance advantage becoming more noticeable in datasets containing a higher proportion of errors (i.e., H–3–2 and H–3–3).

In all three datasets, OrthoPhonoMatch's Precision@1, AP@1, and NDCG@1 scores surpassed the highest scores achieved by the competing techniques. In the most challenging scenario (i.e., H–3–3), OrthoPhonoMatch achieved 50% recall by $k = 6$, whereas competing techniques required at least $k = 9$ to reach comparable levels.

5.2. **Omission errors.** Table 5 presents the performance results for queries containing omission errors. Consistent with findings for homophonic errors, OrthoPhonoMatch achieved the highest win-minus-loss score (210) across all four metrics. The relative ranking of spell correction techniques remained largely unchanged from Table 4, except for Norvig$_{\text{DeepCut}}$ surpassing Hunspell$_{\text{NewMM}}$ to take fourth place. This further confirms the stability of the results, as the relative rankings of top- and bottom-performing techniques remain consistent.

Figure 3 provides a detailed comparison of the top-performing techniques: OrthoPhonoMatch, Norvig$_{\text{NewMM}}$, SymSpell$_{\text{NewMM}}$, and Norvig$_{\text{DeepCut}}$. Following the same format as Figure 2, it illustrates performance across k values from 1 to 20. The results confirm OrthoPhonoMatch's consistent superiority over other techniques across all datasets and

TABLE 5. Comparison of spell correction techniques for **omission errors**, evaluated using aggregated win-minus-loss scores across datasets O–3–1, O–3–2, and O–3–3. Higher scores indicate superior performance.

Rank	Techniques	#win-minus-loss of Mean				Total
		Precision@ 5, 10, 20	Recall@ 5, 10, 20	AP@ 5, 10, 20	NDCG@ 5, 10, 20	
1	OrthoPhonoMatch	57	27	75	51	210
2	$\text{Norvig}_{\text{NewMM}}$	45	22	62	44	173
3	$\text{SymSpell}_{\text{NewMM}}$	41	22	57	37	157
4	$\text{Norvig}_{\text{DeepCut}}$	7	5	2	5	19
5	$\text{Hunspell}_{\text{NewMM}}$	−9	1	−4	−5	−17
6	$\text{Hunspell}_{\text{DeepCut}}$	−19	−12	−20	−13	−64
7	$\text{SymSpell}_{\text{DeepCut}}$	−20	−9	−24	−19	−72
8	Unsupervised $\text{FastText}_{\text{NewMM}}$	−49	−21	−73	−43	−186
9	Unsupervised $\text{FastText}_{\text{DeepCut}}$	−54	−31	−77	−59	−221

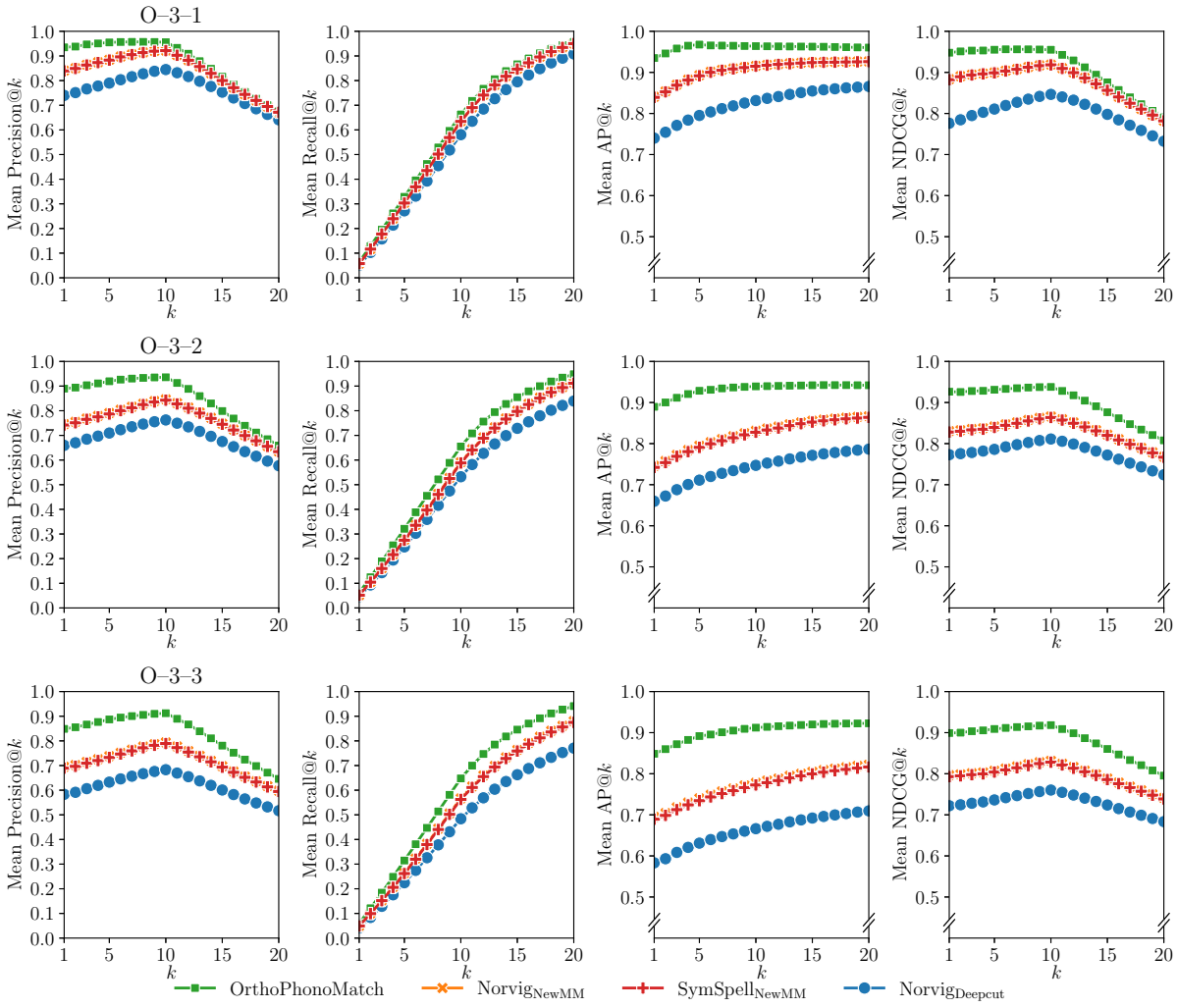

FIGURE 3. Performance of spell correction techniques with positive win-minus-loss scores in addressing **omission errors**

metrics at nearly all evaluated k levels. $\text{Norvig}_{\text{NewMM}}$ and $\text{SymSpell}_{\text{NewMM}}$ demonstrated comparable performance, trailing OrthoPhonoMatch, while the fourth-ranked method (i.e., $\text{Norvig}_{\text{DeepCut}}$) consistently lagged behind the top three.

Consistent with the findings for homophonic errors, OrthoPhonoMatch demonstrated strong top-rank performance overall for omission errors. While several competing techniques also performed well on the simpler single-error dataset (i.e., O–3–1), in the multi-error scenarios (i.e., O–3–2 and O–3–3 datasets), OrthoPhonoMatch's Precision@1, AP@1, and NDCG@1 scores generally surpassed the highest scores of competing techniques.

In summary, the experimental results confirm OrthoPhonoMatch's superiority across multiple metrics and datasets, demonstrating the effectiveness of its integrated orthographic and phonological approach.

6. **Discussion.**

6.1. **Statistical significance.** Win/tie/loss statistics, as suggested by Phannachitta and Matsumoto [50] and Keung et al. [52], can help assess the stability of rankings in experimental evaluations, particularly when results exhibit consistent patterns across multiple metrics and datasets. Although perfect agreement across all rankings is unlikely, the techniques tend to form three distinct performance tiers: high, moderate, and poor performers. Rankings may fluctuate within a tier but rarely shift across tiers, reinforcing the stability of relative comparisons. Stability is generally higher in the top and bottom tiers, where performance differences are more pronounced, than in the middle tier, where techniques often perform comparably. The results in Tables 4 and 5 clearly follow this pattern, grouping spell correction techniques into three performance tiers:

- **High performers:** OrthoPhonoMatch, $\text{Norvig}_{\text{NewMM}}$, and $\text{SymSpell}_{\text{NewMM}}$
- **Moderate performers:** $\text{Hunspell}_{\text{NewMM}}$ and $\text{Norvig}_{\text{DeepCut}}$
- **Poor performers:** $\text{Hunspell}_{\text{DeepCut}}$, $\text{SymSpell}_{\text{DeepCut}}$, Unsupervised $\text{FastText}_{\text{NewMM}}$, and Unsupervised $\text{FastText}_{\text{DeepCut}}$

This tier-based classification reinforces the reliability of the experimental methodology. The observed consistency in the win/tie/loss analysis, combined with the bootstrap test's ability to statistically differentiate techniques, aligns with findings from Büttcher et al. [6]. These results support the combined use of win/tie/loss statistics and the robust percentile bootstrap test with a modified one-step M-estimator as a robust approach for evaluating methods that rely on discrete and bounded evaluation metrics.

6.2. **Component analysis of OrthoPhonoMatch.** To further assess OrthoPhonoMatch's effectiveness, this section decomposes its two core components and examines their individual contributions: Ortho, which relies solely on orthographic similarity, and Phono, which considers only phonological similarity. These components, along with the integrated OrthoPhonoMatch model, were evaluated against four spell correction techniques pairing with the NewMM tokenizer, as prior analyses have consistently demonstrated its superiority over DeepCut. Table 6 presents the results using the same methodology as Tables 4 and 5 and aggregates win-minus-loss scores across four evaluation metrics at k values of 5, 10, and 20.

The results indicate that OrthoPhonoMatch and its Ortho-only variant were the only spell correction techniques consistently achieved top rankings based on win-minus-loss counts across all error types and metrics evaluated. When analyzed independently, neither the Ortho nor Phono component alone matched the effectiveness of the full OrthoPhonoMatch model, reinforcing the importance of integrating both approaches. Notably, the

TABLE 6. Comparison of spell correction techniques in analyzing the key components of OrthoPhonoMatch based on win-minus-loss scores

Rank	Techniques	Error types	#win-minus-loss of Mean				Total	
			Precision@ 5, 10, 20	Recall@ 5, 10, 20	AP@ 5, 10, 20	NDCG@ 5, 10, 20		
1	OrthoPhonoMatch	Homophonic	44	25	65	43	177	366
		Omission	50	24	69	46	189	
2	OrthoPhonoMatch (Ortho component)	Homophonic	4	8	6	2	20	88
		Omission	17	10	23	18	68	
3	$\text{Norvig}_{\text{NewMM}}$	Homophonic	−10	−3	−10	−11	−34	21
		Omission	11	10	23	11	55	
4	$\text{SymSpell}_{\text{NewMM}}$	Homophonic	−10	−3	−10	−11	−34	16
		Omission	13	8	19	10	50	
5	OrthoPhonoMatch (Phono component)	Homophonic	22	15	35	23	95	−53
		Omission	−37	−20	−53	−38	−148	
6	$\text{Hunspell}_{\text{NewMM}}$	Homophonic	−30	−16	−39	−28	−113	−102
		Omission	4	0	3	4	11	
7	Unsupervised $\text{FastText}_{\text{NewMM}}$	Homophonic	−7	−14	−20	−11	−52	−110
		Omission	−19	−7	−19	−13	−58	

Ortho component was the dominant contributor, outperforming $\text{Norvig}_{\text{NewMM}}$, the second-best performer in previous experiments.

Each component exhibited distinct strengths. The Ortho component was the most effective in handling omission errors, while the Phono component performed better in correcting homophonic errors. Although the Ortho component handled omission errors well, its performance on homophonic errors was weaker, though still superior to other algorithms. Conversely, the Phono component demonstrated strong performance in correcting homophonic errors but was ineffective against omission errors, producing the lowest scores in this category. However, when combined, OrthoPhonoMatch significantly outperformed all competing techniques.

These findings confirm that OrthoPhonoMatch's superiority arises from its ability to integrate orthographic and phonological similarity. Orthographic similarity captures character-level patterns, while phonological representation leverages sound-based similarity using IPA. This supports the hypothesis that users may recall either the pronunciation or specific characters of a word when making spelling errors, increasing OrthoPhonoMatch's adaptability to different types of mistakes.

6.3. **Generalizability to other tonal languages.** OrthoPhonoMatch integrates orthographic and phonological features in a hybrid approach, specifically designed and evaluated for Thai. This approach suggests potential applicability to other tonal languages, if suitable linguistic resources and necessary adaptations are made. Key requirements include robust syllable segmentation and reliable phonetic transcription. This study employed IPA representations generated using conversion libraries (e.g., `PyThaiNLP`). Therefore, extension to other tonal languages is possible if robust segmentation and accurate IPA conversion tools are available for them. Early experiments indicated that romanized transcription yielded less effective results compared to IPA, confirming the dependency on detailed phonetic representation. Nonetheless, romanization remains a possible alternative for these languages if IPA resources are unavailable, although its accuracy is likely lower based on preliminary results.

For other tonal languages (e.g., Vietnamese or Mandarin), the phonological component is particularly crucial. It directly addresses the limitations of conventional orthographic methods by being essential for disambiguating tone-dependent meanings, provided suitable phonetic resources are available. Future work should primarily focus on verifying the availability and robustness of the necessary linguistic resources for specific target tonal languages, as the core methodology is expected to be applicable.

6.4. **Threats to validity.** Threats to validity were mitigated through careful experimental design. To ensure internal validity, this study specifically focused on homophonic and omission errors common in the Autopair product catalog, using standardized query lengths to maintain fair comparisons, as detailed in Table 3. The observed consistency of performance across multiple metrics and datasets (Figures 2 and 3) reinforces the reliability of the experimental setup. Construct validity was maintained by employing the robust percentile bootstrap test with a modified one-step M-estimator, ensuring robust statistical comparisons aligned with the study's objectives. While the results demonstrated strong internal consistency, external validity could be strengthened by evaluating OrthoPhonoMatch on additional error types, such as insertion and substitution, and by expanding the analysis to queries from other domains or industrial contexts.

7. **Conclusion.** This study introduces OrthoPhonoMatch, a retrieval-enhancing approach designed for Thai-language catalog search engines, particularly in industrial applications where misspelled queries frequently reduce search efficiency. The method was evaluated using data from Autopair [5], a Thai automotive aftermarket platform where query misspellings affect auto parts ordering and workshop management systems. The evaluation focused on homophonic and omission errors, two of the most common error types in Thai catalog search queries. Experimental results, evaluated using benchmarking metrics and statistical analysis, confirmed OrthoPhonoMatch's effectiveness and robustness across diverse retrieval conditions.

OrthoPhonoMatch outperformed eight non-contextual spell correction techniques, achieving higher retrieval accuracy by returning a greater proportion of relevant documents, regardless of whether relevance was assessed in binary or graded terms. Component analysis confirmed that neither the orthographic nor phonological component alone matched the effectiveness of the combined OrthoPhonoMatch approach, reinforcing the importance of integrating both similarity measures for improved spell correction. The orthographic component was more effective in handling omission errors, while the phonological component performed better in correcting homophonic errors, highlighting their complementary roles in addressing different types of misspellings. These results align with the hypothesis that users may rely on either pronunciation or specific orthographic cues when making spelling errors, enhancing OrthoPhonoMatch's adaptability to diverse error types.

The findings of this study highlight OrthoPhonoMatch's practical value in improving search accuracy for Thai-language catalog queries, particularly in industrial applications such as Autopair's online auto parts retrieval system. Future work could extend the evaluation to a broader range of spelling errors, assess its applicability in additional search domains, and optimize its computational efficiency for large-scale deployments. Advancements in phonological and orthographic representations could further enhance its generalizability across Thai and other tonal languages, strengthening its potential for real-world adoption in commercial catalog search engines.

REFERENCES

[1] D. A. Hanauer, Q. Mei, J. Law, R. Khanna and K. Zheng, Supporting information retrieval from electronic health records: A report of University of Michigan's nine-year experience in developing and

using the electronic medical record search engine (EMERSE), *J. Biomed. Inform.*, vol.55, pp.290-300, 2015.

[2] J. Misutka and L. Galambos, Mathematical extension of full text search engine indexer, *Proc. of the 3rd Int. Conf. on Information and Communication Technologies: From Theory to Applications*, pp.1-6, 2008.

[3] W. J. Wilbur, W. Kim and N. Xie, Spelling correction in the PubMed search engine, *Inf. Retr.*, vol.9, pp.543-564, 2006.

[4] S. G. Desta and G. S. Lehal, Automatic spelling error detection and correction for Tigrigna information retrieval: A hybrid approach, *Bull. Electr. Eng. Inform.*, vol.12, no.1, pp.387-394, 2023.

[5] AutoPair Co., Ltd., *AutoPair: Your Digitalization Partner for All Automotive Aftermarket Industry Related*, 2021, https://www.autopair.co/, Accessed on March 1st, 2025.

[6] S. Büttcher, C. L. Clarke and G. V. Cormack, *Information Retrieval: Implementing and Evaluating Search Engines*, MIT Press, 2016.

[7] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.*, vol.7, pp.1-30, 2006.

[8] R. Wilcox, *Modern Statistics for the Social and Behavioral Sciences: A Practical Introduction*, CRC Press, 2011.

[9] W. B. Croft, D. Metzler and T. Strohman, *Search Engines: Information Retrieval in Practice*, Addison-Wesley Reading, 2010.

[10] C. D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[11] M. H. Nguyen, A label-oriented approach for text classification, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1593-1609, 2020.

[12] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Comp. Netw. ISDN Syst.*, vol.30, nos.1-7, pp.107-117, 1998.

[13] T. Y. Liu, Learning to rank for information retrieval, *Found. Trends Inf. Retr.*, vol.3, no.3, pp.225-331, 2009.

[14] B. G. Silverman, M. Bachann and K. Al-Akharas, Do what I mean: Online shopping with a natural language search agent, *IEEE Intell. Syst.*, vol.16, no.4, pp.48-53, 2001.

[15] J. A. De Blasio, T. Kawamura and T. Hasegawa, Catalog search engine: Semantics applied to products search, *Proc. of the 3rd Int. Semantic Web Conf.*, pp.11-20, 2004.

[16] W. Satriady, M. A. Bijaksana and K. M. Lhaksmana, Quranic Latin query correction as a search suggestion, *Procedia Comput. Sci.*, vol.157, pp.183-190, 2019.

[17] B. Martins and M. J. Silva, Spelling correction for search engine queries, *Proc. of the 4th Int. Conf. on Advances in Natural Language Processing*, pp.372-383, 2004.

[18] K. Kukich, Techniques for automatically correcting words in text, *ACM Comput. Surv.*, vol.24, no.4, pp.377-439, 1992.

[19] S. Konstantinidis, Computing the Levenshtein distance of a regular language, *Proc. of the 2005 IEEE Information Theory Workshop*, pp.113-116, 2005.

[20] J. Zhang, X. Guo, S. Bodapati and C. Potts, Multi-teacher distillation for multilingual spelling correction, *Proc. of the 2023 Conf. on Empirical Methods in Natural Language Processing: Industry Track*, pp.142-151, 2023.

[21] X. Guo, R. Patki, D. Everaert and C. Potts, Retrieval augmented spelling correction for E-commerce applications, *Proc. of the 2024 Conf. on Empirical Methods in Natural Language Processing: Industry Track*, pp.73-79, 2024.

[22] R. Xuan, J. Shim and S.-G. Lee, Fast passage retrieval in weighted Hamming space for open-domain question answering, *ICIC Express Letters, Part B: Applications*, vol.15, no.4, pp.373-380, 2024.

[23] S. Thaiprayoon, A. Kongthon and C. Haruechaiyasak, ThaiQCor 2.0: Thai query correction via soundex and word approximation, *Proc. of the 5th Int. Conf. on Advanced Informatics: Concept Theory and Applications*, pp.113-117, 2018.

[24] D. Li and D. Peng, Spelling correction for Chinese language based on Pinyin-Soundex algorithm, *Proc. of the Int. Conf. on Internet Technology and Applications*, pp.1-3, 2011.

[25] S. Gowri, P. Sathish Kumar, K. Geetha Rani, R. Surendran and J. Jabez, Usage of a binary integrated spell check algorithm for an upgraded search engine optimization, *Meas.: Sens.*, vol.24, pp.1-6, 2022.

[26] I. P. Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[27] W. Phatthiyaphaibun, K. Chaovavanich, C. Polpanumas, A. Suriyawongkul, L. Lowphansirikul, P. Chormai, P. Limkonchotiwat, T. Suntorntip and C. Udomcharoenchaikit, PyThaiNLP: Thai natural

language processing in Python, *Proc. of the 3rd Workshop for Natural Language Processing Open Source Software*, pp.25-36, 2023.
[28] R. Arreerard, S. Mander and S. S. Piao, Survey on Thai NLP language resources and tools, *Proc. of the 13th Language Resources and Evaluation Conf.*, pp.6495-6505, 2022.
[29] K. Suraratchai and S. Phoomvuthisarn, Thai language sentiment analysis with a hybrid method on WangchanBERTa-CNN-BiLSTM, *J. Inf. Sci. Technol.*, vol.14, no.2, pp.1-11, 2024.
[30] J. Satjathanakul and T. Siriborvornratanakul, Sentiment analysis in product reviews in Thai language, *Int. J. Inf. Technol.*, pp.1-7, 2024.
[31] S. Palahan, Improving access to trade and investment information in Thailand through intelligent document retrieval, *Int. J. Comput. Appl.*, vol.30, no.4, pp.402-411, 2023.
[32] G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval, *Inf. Process. Manag.*, vol.24, pp.513-523, 1988.
[33] R. Řehuřek, P. Sojka et al., *Gensim: Topic Modelling for Humans*, 2022, https://radimrehurek.com/gensim/, Accessed on March 1st, 2025.
[34] B. Heinzerling and M. Strube, Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation, *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.273-291, 2019.
[35] L. Német, *Hunspell*, 2022, https://hunspell.github.io/, Accessed on March 1st, 2025.
[36] P. Norvig, *How to Write a Spelling Corrector*, 2016, https://norvig.com/spell-correct.html, Accessed on March 1st, 2025.
[37] W. Garbe, *2012 Faster Spelling Correction Algorithm*, 2024, https://seekstorm.com/blog/1000x-spelling-correction/, Accessed on March 1st, 2025.
[38] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou and T. Mikolov, FastText.zip: Compressing text classification models, *arXiv Preprint*, arXiv: 1612.03651, 2016.
[39] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching word vectors with subword information, *Trans. Assoc. Comput. Linguist.*, vol.5, pp.135-146, 2017.
[40] V. Shepelev, *Spylls: Hunspell Ported to Python*, 2024, https://spylls.readthedocs.io/, Accessed on March 1st, 2025.
[41] Facebook Research, *FastText: Library for Efficient Text Classification and Representation Learning*, 2019, https://github.com/facebookresearch/fastText/, Accessed on March 1st, 2025.
[42] C. Zhao and S. Sahni, String correction using the Damerau-Levenshtein distance, *BMC Bioinformatics*, vol.20, pp.1-28, 2019.
[43] R. Kittinaradorn, K. Chaovavanich, T. Achakulvisut, K. Srithaworn, P. Chormai, C. Kaewkasi, T. Ruangrong and K. Oparad, *DeepCut: A Thai Word Tokenization Library Using Deep Neural Network*, 2019, https://doi.org/10.5281/zenodo.3457707, Accessed on March 1st, 2025.
[44] N. Tongtep and T. Theeramunkong, Simultaneous character-cluster-based word segmentation and named entity recognition in Thai language, *Proc. of the 5th Int. Conf. Knowledge, Information, and Creativity Support Systems*, pp.216-225, 2011.
[45] M. Sanderson and J. Zobel, Information retrieval system evaluation: Effort, sensitivity, and reliability, *Proc. of the 28th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp.162-169, 2005.
[46] M. D. Smucker, J. Allan and B. Carterette, A comparison of statistical significance tests for information retrieval evaluation, *Proc. of the 16th ACM Conf. on Information and Knowledge Management*, pp.623-632, 2007.
[47] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs and A. Pohthong, Robust statistical methods for empirical software engineering, *Empirical Softw. Eng.*, vol.22, no.2, pp.579-630, 2017.
[48] P. Phannachitta, On an optimal analogy-based software effort estimation, *Inf. Softw. Technol.*, vol.125, 106330, 2020.
[49] M. Li and L. Wang, Soft subspace clustering ensemble based on hedonic games, *International Journal of Innovative Computing, Information and Control*, vol.17, no.4, pp.1327-1343, 2021.
[50] P. Phannachitta and K. Matsumoto, Model-based software effort estimation – A robust comparison of 14 algorithms widely used in the data science community, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.569-589, 2019.
[51] R. R. Wilcox and F. Schönbrodt, *WRS: A Package of R. R. Wilcox' Robust Statistics Functions*, 2019, https://rdrr.io/rforge/WRS/, Accessed on March 1st, 2025.
[52] J. Keung, E. Kocaguneli and T. Menzies, Finding conclusion stability for selecting the best effort predictor in software effort estimation, *Automated Softw. Eng.*, vol.20, no.4, pp.543-567, 2013.

Author Biography

Passakorn Phannachitta received his M.Eng. and D.Eng. degrees from the Nara Institute of Science and Technology, Japan, in 2013 and 2016, respectively. He is currently an Assistant Professor at the College of Arts, Media and Technology, Chiang Mai University, Thailand. His research interests include empirical software engineering and information retrieval. His professional expertise, gained from industry, focuses on the integration of data science and cloud orchestration within software engineering practices.

Chartchai Doung sa-ard received his Master's degree from Chulalongkorn University, Thailand in 2004, and his Ph.D. degree from the University of Bradford, UK, in 2011. He is currently an Assistant Professor of Software Engineering at the College of Arts, Media and Technology, Chiang Mai University, Thailand. His research interests encompass software design, software testing, enterprise business solutions, and software process improvement.