ANALISIS DETEKSI KESALAHAN PENULISAN (*TYPO*) PADA TUGAS MAHASISWA MENGGUNAKAN ALGORITMA LEVENSHTEIN DISTANCE

Ahmad Khoir Alhaq, Nur Asitah, dan Rizky Aditya Nugroho
Informatika, Universitas Nadhlatul Ulama Sidoarjo
Jl. Lingkar Timur KM 5,5 Rangkah Kidul, Kec. Sidoarjo, Kab. Sidoarjo, Indonesia
ahmadkhoiralhaq@unusida.ac.id

ABSTRAK

Penelitian ini bertujuan menganalisis deteksi kesalahan penulisan atau *typographical error* pada tugas mahasiswa menggunakan *Algoritma Levenshtein Distance* berbasis wordlist bahasa Indonesia. *Algoritma Levenshtein Distance* dipilih karena mampu menghitung jarak minimum perubahan karakter melalui operasi *insertion, deletion,* dan *substitution*, sehingga sesuai untuk mendeteksi kesalahan ketik mekanis yang umumnya memiliki kemiripan bentuk dengan kata baku. Analisis eksperimental kuantitatif memproses 80 dokumen yang dikonversi ke PDF melalui *preprocessing* teks, normalisasi token, perbandingan daftar kata Indonesia yang terstruktur, dan perhitungan jarak edit. Kesalahan diklasifikasikan sebagai *insertion*, *deletion*, atau *subtitution*. *Error rate*: 0,45%–14,7% per dokumen. Dari total 536 operasi *edit distance*, kesalahan paling dominan adalah *substitution* sebanyak 303 operasi atau 56,53%, diikuti *deletion* sebanyak 177 operasi atau 33,02%, dan *insertion* sebanyak 56 operasi atau 10,45%. Pola dominan ini, terkait dengan kecepatan mengetik dan kedekatan tata letak pada *keyboard* QWERTY. Integrasi *wordlist* standar meningkatkan akurasi dalam membedakan variasi ortografis dari kesalahan mekanis. Tingkat rendah hingga sedang dominan, namun penyimpangan memerlukan pendekatan pedagogis yang terarah. Peneliti fokus pada karakter mengesampingkan aspek semantik. Kerangka kerja ini memfasilitasi evaluasi otomatis dengan kompleksitas rendah untuk platform digital, meningkatkan literasi akademik dan kesadaran linguistik.

Kata kunci : *Levenshtein Distance, kesalahan penulisan, typo, tugas mahasiswa, wordlist*

1. PENDAHULUAN

Perkembangan teknologi dalam dunia pendidikan khususnya digital membawa dampak serius terhadap proses pengerjaan tugas mahasiswa. Kemudahan akses informasi di internet, ditambah dengan maraknya penggunaan alat bantu berbasis kecerdasan buatan (*Artificial Intelligence*), menyebabkan meningkatnya praktik *copy–paste* dan *paraphrase* otomatis dalam penyusunan tugas akademik [1], [2].

Kondisi ini tidak hanya memengaruhi orisinalitas karya mahasiswa, tetapi juga berpotensi menurunkan kualitas penulisan, khususnya dari sisi ketepatan ejaan dan struktur kata [3].

Penggunaan teknik *paraphrase* berbasis AI sering kali menghasilkan perubahan kata yang bersifat mekanis tanpa mempertimbangkan kaidah bahasa yang benar [4].

Akibatnya, kesalahan penulisan seperti *typographical error* (*typo*) [5], penggunaan kata tidak baku [6], maupun kesalahan tata bahasa menjadi semakin sulit terdeteksi secara manual oleh dosen, terutama ketika jumlah tugas yang harus dievaluasi sangat banyak [7], [8].

Hal ini menimbulkan kebutuhan akan sistem otomatis yang mampu membantu proses analisis kesalahan penulisan secara objektif dan efisien.

Penelitian sebelumnya mengenai deteksi kesalahan penulisan telah berkembang melalui berbagai pendekatan algoritma dan studi perilaku pengguna. Oleh peneliti sebelumnya sistem KEBI 1.0 yang mengintegrasikan metode *Dictionary Lookup* untuk deteksi kata non-standar dan Peter Norvig *Spelling Corrector* untuk kesalahan tipografi pada karya ilmiah, mencapai akurasi 100% pada kata non-standar [9].

Sementara itu, peneliti berbeda menguji kombinasi algoritma Jaro-Winkler Distance dan N-Gram, di mana N-Gram berfungsi untuk mencari saran perbaikan terbaik dari daftar kandidat yang dihasilkan, dengan tingkat akurasi tertinggi mencapai 85,7% [10].

Penggunaan algoritma Levenshtein Distance juga telah terbukti efektif dalam konteks pencarian informasi pariwisata, dengan menghasilkan nilai presisi sebesar 1,0 [11].

Berbagai metode telah dikembangkan untuk mendeteksi kesalahan penulisan dalam teks, mulai dari pendekatan berbasis aturan (*rule-based*), kamus (*dictionary-based*), hingga algoritma pengukuran kesamaan *string* seperti N-gram, Jaro–Winkler, dan Levenshtein Distance [12], [13].

Masing-masing metode memiliki kelebihan dan keterbatasan, baik dari sisi akurasi, kompleksitas komputasi, maupun kemampuan dalam menangani variasi kesalahan penulisan yang beragam.

Namun demikian, sebagian besar penelitian sebelumnya masih berfokus pada deteksi kesalahan penulisan secara umum tanpa mengoptimalkan pemanfaatan *wordlist* sebagai basis pembanding kata baku. Meskipun, *wordlist* yang terstruktur dapat berperan penting dalam meningkatkan ketepatan identifikasi kesalahan, khususnya pada konteks penilaian tugas mahasiswa yang menggunakan bahasa formal dan istilah akademik tertentu [12].

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pendekatan deteksi kesalahan penulisan pada tugas mahasiswa dengan memanfaatkan algoritma Levenshtein Distance berbasis *wordlist*. Pendekatan ini diharapkan mampu mengidentifikasi tingkat kesalahan penulisan kata secara lebih akurat dengan membandingkan kata dalam teks tugas terhadap daftar kata baku yang telah ditentukan, sehingga dapat menjadi solusi pendukung dalam proses evaluasi tugas akademik secara otomatis dan sistematis.

Penelitian-penelitian sebelumnya telah membahas deteksi kesalahan penulisan dengan berbagai pendekatan, seperti *dictionary lookup*, Peter Norvig Spelling Corrector, Jaro-Winkler Distance, N-Gram, dan Levenshtein Distance [9]–[13].

Namun, sebagian besar penelitian tersebut masih berfokus pada kinerja algoritma koreksi kata secara umum, bukan pada analisis pola kesalahan penulisan dalam dokumen tugas mahasiswa. Selain itu, penelitian terdahulu lebih banyak menempatkan algoritma sebagai alat koreksi otomatis, sedangkan kajian mengenai pemanfaatan hasil operasi edit distance sebagai instrumen analisis perilaku pengetikan akademik masih terbatas.

Fokus penelitian ini berbeda karena tidak hanya mendeteksi kata yang salah, tetapi juga menganalisis jenis operasi kesalahan yang muncul, yaitu *insertion*, *deletion*, dan *substitution*, serta mengaitkannya dengan kemungkinan penyebab mekanis seperti kesalahan pengetikan, hilangnya spasi, kedekatan tombol keyboard QWERTY, dan dampak penggunaan teks hasil parafrasa otomatis. Dengan demikian, kontribusi penelitian ini terletak pada penggunaan Levenshtein Distance berbasis wordlist sebagai alat evaluasi teknis terhadap kualitas penulisan tugas mahasiswa, bukan sekadar sebagai sistem koreksi kata. Pendekatan ini penting karena dosen memerlukan alat bantu yang tidak hanya menunjukkan adanya kesalahan, tetapi juga memberikan gambaran pola kesalahan yang dapat digunakan untuk intervensi pembelajaran, pelatihan *proofreading*, dan peningkatan literasi akademik mahasiswa.

2. TINJAUAN PUSTAKA

Kesalahan penulisan atau *typographical error* (*typo*) adalah masalah krusial dalam pembuatan teks yang sering terjadi akibat ketidaktelitian saat menggunakan *keyboard* pada komputer maupun *smartphone* [13].

Typo didefinisikan sebagai kesalahan dalam pengejaan kata-kata yang dapat menyebabkan ambiguitas bagi pembaca dan bahkan mengubah arti serta makna suatu kalimat [14].

Secara teknis, kesalahan penulisan ini dapat dikategorikan menjadi tiga jenis operasi dasar, yaitu: penambahan (*insertion*), penghapusan (*deletion*), dan penggantian (*substitution*) suatu karakter di dalam kata atau frasa.

Pada lingkungan akademisi, khususnya bagi mahasiswa dalam penyusunan karya ilmiah atau Tugas Akhir, pemahaman tata bahasa dan kaidah kebahasaan sangat penting untuk menciptakan tulisan yang runtut. Kesalahan ejaan sering kali muncul karena pengaruh tata letak huruf pada *keyboard* yang berdekatan atau kurangnya pemahaman terhadap Pedoman Umum Ejaan Bahasa Indonesia (PUEBI) [14].

Oleh karena itu, diperlukan sistem otomatis yang efektif dan efisien untuk mendeteksi dan mengoreksi kesalahan tersebut guna meningkatkan kualitas karya tulis ilmiah mahasiswa.

Algoritma Levenshtein Distance, yang juga dikenal sebagai *Edit Distance*, pertama kali ditemukan oleh ilmuwan Rusia bernama Vladimir Levenshtein pada tahun 1965 [11].

Algoritma ini merupakan matriks *string* yang berfungsi untuk mengukur perbedaan atau jarak (*distance*) antara dua buah *string*. Nilai jarak tersebut ditentukan oleh jumlah minimum operasi perubahan yang diperlukan untuk mentransformasikan satu *string* sumber menjadi *string* target [15].

Tiga operasi utama dalam algoritma ini adalah: *Insertion* (Penyisipan): Menambahkan satu karakter baru ke dalam *string*, *Deletion* (Penghapusan): Menghapus satu karakter dari *string*, dan *Substitution* (Penggantian): Mengganti satu karakter dengan karakter lain [16].

Perhitungan jarak ini dilakukan menggunakan matriks dua dimensi. Hasil akhir atau nilai jarak Levenshtein terletak pada elemen pojok kanan bawah matriks tersebut; semakin kecil nilainya, maka tingkat kemiripan (*similarity*) antara kedua *string* tersebut semakin besar.

Deteksi kesalahan penulisan termasuk dalam bidang riset Natural Language Processing (NLP), yaitu cabang kecerdasan buatan yang memberikan kemampuan pada mesin untuk membaca, memahami, dan memperoleh makna dari bahasa manusia [15].

Dalam implementasinya, algoritma Levenshtein Distance sering kali dikombinasikan dengan tahapan *text preprocessing* untuk memaksimalkan kinerja sistem. Tahapan *preprocessing* ini umumnya terdiri dari: *Case Folding*: Mengubah semua huruf dalam dokumen menjadi huruf kecil. *Tokenizing*: Memecah kalimat menjadi kumpulan kata tunggal (*token*) . *Stopword Removal*: Menghilangkan kata-kata umum yang tidak memiliki makna penting seperti "yang", "di", atau "dari". *Stemming*: Menghapus kata imbuhan untuk mendapatkan kata dasar [13], [17].

Penerapan *preprocessing* terbukti dapat mempercepat waktu eksekusi sistem dan menghasilkan akurasi yang lebih stabil dalam mendeteksi kemiripan teks.

Beberapa penelitian sebelumnya telah mengeksplorasi penggunaan Levenshtein Distance dalam berbagai konteks. Salah satu studi membandingkan kinerjanya dengan algoritma Jaro-Winkler, di mana Jaro-Winkler cenderung lebih unggul dalam kecepatan eksekusi untuk kesalahan

kecil, sementara Levenshtein Distance menunjukkan kemampuan yang sangat baik dalam menangani kondisi teks yang lebih kompleks [14].

Penelitian lain menggunakan kombinasi N-gram dan Levenshtein Distance untuk memberikan rekomendasi kata perbaikan berdasarkan nilai *cosine similarity*, yang memberikan hasil presisi terbaik pada jenis kesalahan *insertion* [13].

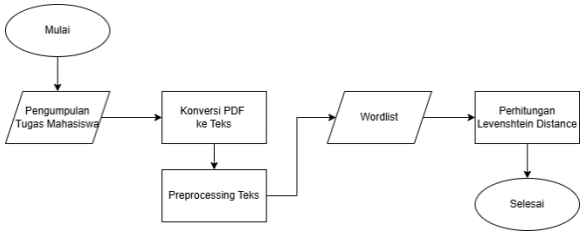
Selain itu, algoritma ini juga telah sukses diimplementasikan dalam pengembangan fitur *Autocomplete* dan *Spell Checker* pada berbagai platform, seperti kamus digital bahasa daerah dan sistem pencarian judul buku di perpustakaan, dengan tingkat akurasi yang mencapai lebih dari 90% dalam memberikan rekomendasi kata yang tepat [18], [19].

Dalam studi kasus tugas mahasiswa, Levenshtein Distance efektif digunakan sebagai alat bantu koreksi otomatis (*auto-correct*) guna mengefisiensi waktu pemeriksaan manual oleh dosen atau pembimbing. Implementasi algoritma ini dalam sistem pendeteksi *typo* pada tugas mahasiswa sangat krusial untuk menjamin kualitas ejaan sesuai dengan kaidah bahasa Indonesia yang berlaku. Dengan adanya otomatisasi ini, proses koreksi yang sebelumnya memakan waktu secara manual dapat dilakukan dengan lebih efisien dan akurat, sehingga secara langsung berkontribusi pada peningkatan standar kualitas karya tulis ilmiah mahasiswa di lingkungan perguruan tinggi.

3. METODE PENELITIAN

Penelitian ini secara garis besar dapat ditunjukkan pada Gambar 1. Pada Gambar 1 menunjukkan bahwa Penelitian ini menerapkan metode analisis kuantitatif berbasis eksperimen untuk mendeteksi kesalahan penulisan (*typographical error*) pada tugas mahasiswa.

Dataset penelitian terdiri atas 80 dokumen tugas mahasiswa dalam format PDF. Dokumen tersebut diekstraksi menjadi format TXT untuk memungkinkan pemrosesan pengolahan data. *Dataset* dipilih untuk merepresentasikan teks akademik mahasiswa yang berpotensi mengandung kesalahan penulisan akibat tindakan *copy–paste* dan *paraphrase* berbasis kecerdasan buatan.



Gambar 1. Bagan Alur Penelitian

Tahap awal pengolahan data dilakukan dengan mengonversi dokumen PDF menjadi teks (.txt). Hasil konversi selanjutnya diproses melalui tahap *preprocessing* untuk meningkatkan kualitas data. Tahapan *preprocessing* meliputi *case folding* untuk menyeragamkan huruf, *tokenizing* untuk memisahkan teks menjadi *token* kata, serta penghapusan tanda baca, angka, dan karakter non-alfabet. Tahap ini bertujuan untuk meminimalkan *noise* yang dapat memengaruhi hasil perhitungan jarak antar kata.

Setiap *token* kata hasil *preprocessing* kemudian dibandingkan dengan *wordlist* kata baku bahasa Indonesia, di mana *dataset* berisikan 105.226 kata dari Kamus Besar Bahasa Indonesia [20], [21].

Dari *dataset* ini selanjutnya digunakan sebagai *wordlist*. Perbandingan dilakukan menggunakan algoritma Levenshtein Distance yang mengukur tingkat kemiripan dua *string* berdasarkan jumlah minimum operasi pengeditan. Operasi yang digunakan dalam algoritma ini meliputi penyisipan (*insertion*), penghapusan (*deletion*), dan substitusi (*substitution*) [22], [23].

Semakin kecil nilai jarak yang dihasilkan, semakin tinggi tingkat kemiripan antara dua kata yang dibandingkan.

Secara matematis, jarak Levenshtein antara dua *string s* (*string*) dan t (*target*) dengan panjang masing-masing $|\ s\ |$dan $|\ t\ |$ didefinisikan sebagai persamaan berikut:

$$D(i,j) = \begin{cases} max\ (i,j), & \text{jika } min\ (i,j) = 0 \\ min \begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+cost \end{cases}, & \text{jika } i,j > 0 \end{cases} \quad (1)$$

dengan $D(i,j)$ menyatakan jarak edit antara substring $s[1..i]$ dan $t[1..j]$, serta $cost = 0$ jika $s_i = t_j$ (tidak ada huruf yang berbeda sama sekali), dan $cost = 1$ jika $s_i \neq t_j$(ada satu huruf yang berbeda sehingga dinyatakan dengan bobot satu) [24], [25].

Nilai Levenshtein Distance minimum antara token kata dan kata pada *wordlist* digunakan sebagai dasar klasifikasi kesalahan penulisan [26], [27].

Apabila nilai jarak edit melebihi ambang batas (*threshold*) yang telah ditentukan, maka *token* kata diklasifikasikan sebagai kesalahan penulisan (*typo*).

Pada penelitian ini, ambang batas atau *threshold* Levenshtein Distance ditetapkan maksimal 2, yaitu token yang tidak ditemukan dalam *wordlist* masih dapat dibandingkan dengan kandidat kata baku apabila memiliki jarak edit 1 atau 2. Nilai ini dipilih karena sebagian besar kesalahan ketik mekanis umumnya berupa perubahan karakter kecil, seperti huruf hilang, huruf bertambah, huruf terganti, atau kombinasi dua operasi sederhana. Penetapan *threshold* 2 digunakan sebagai batas kompromi antara sensitivitas dan ketepatan deteksi, karena *threshold* yang terlalu rendah berpotensi melewatkan kesalahan dengan dua perubahan karakter, sedangkan *threshold* yang terlalu tinggi dapat meningkatkan risiko *false positive*. Dengan demikian, token dikategorikan sebagai *typo* apabila tidak terdapat dalam *wordlist* dan memiliki kandidat kata baku terdekat dengan jarak edit maksimal 2.

Selanjutnya dilakukan perhitungan jumlah kesalahan penulisan pada setiap dokumen tugas mahasiswa untuk dianalisis secara kuantitatif. Hasil

analisis ini digunakan untuk mengevaluasi efektivitas algoritma Levenshtein Distance berbasis *wordlist* dalam mendeteksi kesalahan penulisan pada teks akademik mahasiswa.

4. HASIL DAN PEMBAHASAN

Hasil penelitian diperoleh dari proses deteksi kesalahan penulisan (*typographical error*) pada 80 dokumen tugas mahasiswa yang telah dikonversi dari format PDF ke teks. Setiap dokumen melalui tahap *preprocessing* sebelum dilakukan perhitungan Levenshtein Distance berbasis *wordlist*. *Output* utama dari proses ini berupa jumlah kata terdeteksi *typo*, total kata, serta persentase kealahan penulisan pada masing-masing dokumen.

Berdasarkan hasil pengolahan data, algoritma Levenshtein Distance mampu mengidentifikasi variasi tingkat kesalahan penulisan pada tugas mahasiswa. Perbedaan jumlah *typo* dipengaruhi oleh panjang dokumen, gaya penulisan, serta intensitas penggunaan hasil *paraphrase* otomatis. Dokumen dengan jumlah kata lebih besar cenderung memiliki jumlah kesalahan penulisan yang lebih tinggi, meskipun persentase kesalahan tidak selalu meningkat secara linear.

Tabel 1 menyajikan hasil estimasi deteksi kesalahan penulisan pada beberapa sampel dokumen tugas mahasiswa sebagai representasi dari keseluruhan *dataset*.

Tabel 1. Hasil Deteksi Typo

No.	Dokumen	Jumlah Kata	Jumlah Typo	Persentase Typo (%)
1	Tugas 1	447	17	3,80
2	Tugas 2	227	12	5,29
3	Tugas 3	97	12	12,37
4	Tugas 4	315	10	3,17
5	Tugas 5	131	10	7,63
…	…	…	…	…
76	Tugas 76	134	1	0,75
77	Tugas 77	221	1	0,45
78	Tugas 78	78	1	1,28
79	Tugas 79	144	1	0,69
80	Tugas 80	128	1	0,78

Hasil pada Tabel 1 menunjukkan bahwa kesalahan penulisan mahasiswa dapat diketahui melalui persentase *typo*. Kesalahan yang terdeteksi umumnya berupa kesalahan substitusi huruf, penghilangan huruf, serta penggunaan kata tidak baku yang memiliki jarak edit kecil terhadap kata pada wordlist. Hal ini menunjukkan bahwa algoritma Levenshtein Distance efektif dalam mendeteksi kesalahan penulisan yang bersifat ringan hingga sedang.

Tabel 1 menunjukkan tingkat kesalahan dokumen menunjukkan rentang yang lebar, dari 0,45% hingga 14,7%. *Error rate* di bawah 1% umumnya berasal dari dokumen yang dikerjakan secara manual dengan ketelitian tinggi, atau mahasiswa telah menggunakan fitur pemeriksa ejaan bawaan sebelum mengumpulkan tugas. Sebaliknya, *error rate* antara 12% hingga 14% tergolong anomali jika dianggap sebagai kesalahan ketik murni. Analisis lebih lanjut mengungkap dua penyebab utama. Pertama, praktik *copy-paste* dari PDF atau halaman web tanpa penyesuaian format, yang mengakibatkan hilangnya spasi antar kata (misalnya "yangakan", "dalamhal") sehingga algoritma membaca gabungan kata tersebut sebagai satu entitas yang salah. Kedua, penggunaan alat parafrasa atau penerjemah otomatis berkualitas rendah yang menghasilkan kata tidak baku, struktur kalimat kacau, atau istilah asing yang tidak tertranslasi dengan baik.

Untuk dapat memahami pola kesalahan yang terjadi secara lebih mendalam, dilakukan analisis terhadap 80 file dengan total 536 operasi *edit distance*. Hasilnya menunjukkan bahwa pola kesalahan terbanyak adalah penggantian karakter (*substitution*) yaitu sebesar 303 operasi atau 56,53% dari total keseluruhan. Disusul oleh penghapusan karakter (*deletion*) sebanyak 177 operasi dengan persentase 33,02%, sementara penambahan karakter (*insertion*) merupakan pola yang paling jarang terjadi yaitu hanya 56 operasi atau 10,45% dari total keseluruhan. Temuan ini mengonfirmasi hipotesis awal bahwa kemudahan teknologi seperti *copy-paste* dan parafrasa otomatis berkorelasi dengan penurunan kualitas mekanis penulisan, terutama jika tidak diimbangi dengan proses *proofreading* oleh mahasiswa.

4.1. Analisis Pola Tipografi Berbasis Frekuensi Kemunculan

Pola tipografi berbobot frekuensi dalam penelitian ini menyajikan representasi komprehensif mengenai tipologi kesalahan penulisan yang memiliki impak paling signifikan terhadap degradasi kualitas linguistik dokumen. Berbeda dengan analisis variasi kata statis, pendekatan berbasis frekuensi mampu mengidentifikasi intensitas kesalahan yang secara faktual "mengontori" korpus data. Hasil ekstraksi data menunjukkan bahwa pola dominan manifestasi kesalahan adalah penghilangan karakter (*deletion*) dan substitusi non-tetangga (*non-adjacent substitution*), di mana masing-masing kategori memberikan kontribusi sebesar 25% dari total durasi kemunculan. Pola berikutnya diikuti oleh kesalahan *multi-edit* (Jarak Levenshtein ≥ 2) serta penambahan karakter (insertion).

Secara mekanis, tingginya angka penghilangan karakter mengindikasikan adanya korelasi dengan kecepatan pengetikan yang tinggi (*high-speed typing*) atau tekanan pada *keyboard* yang tidak tuntas (*incomplete keystroke*). Sebaliknya, fenomena penambahan karakter secara tipografis sering kali diasosiasikan dengan gejala *double hit* atau *key bounce* pada *keyboard*.

Tabel 2. Distribusi Pola Kesalahan Tipografi Berdasarkan Analisis Levenshtein Distance

Pola	Jumlah (*freq*)	%	Dominan (salah/benar)	Interpretasi
Penghilangan karakter	24	24.24	*furniture→furnitur, journal→jurnal, pengelolahan→pengelolaan*	*Keystroke* "kepencet kurang" (huruf terlewat) saat mengetik cepat / tidak menekan penuh.
Substitusi non-tetangga	24	24.24	*industry→industri, meterial→material, berfikir→berpikir*	Banyak yang bukan murni salah pencet tombol tetangga, tetapi pergeseran ejaan (serapan/baku) atau kebiasaan penulisan.
Multi-edit (*distance* 2)	22	22.22	*astitah→istilah, compass→kompas*	Kombinasi (sub+del / sub+ins). Biasanya terjadi pada kata yang berubah ejaan (serapan) atau salah ketik bertingkat.
Penambahan karakter	16	16.16	*karna→karena*	*Key bounce* / "ter-tekan dobel" atau menambah huruf karena prediksi motorik saat mengetik cepat.
Substitusi tombol tetangga (QWERTY)	5	5.05	*demgan→dengan, nasa→masa, prosuktif→produktif*	Ini yang paling "*keyboard-driven*": *slip/fat-finger* ke tombol sekitar.
Jarak 4 (pergeseran besar)	3	3.03	*fashion→fesyen*	Umumnya bukan *typo keyboard*, tapi normalisasi ejaan serapan.
Spasi hilang	3	3.03	*seringkali→sering kali*	Pola "*layout*" (*spacebar*) → *spacebar* tidak ditekan atau *auto-join*.
Multi-sub tetangga	1	1.01	*memitigasi→memotivasi*	Ada beberapa substitusi yang kebetulan berada di area tombol berdekatan.
Transposisi	1	1.01	*bersitirahat→beristirahat*	Huruf tertukar (umum saat mengetik cepat).

Namun, temuan ini mengungkap dimensi lain di mana sebagian pasangan kata dalam kategori penghilangan dan substitusi tidak murni disebabkan oleh malfungsi mekanis, melainkan representasi dari upaya normalisasi ejaan (misalnya, penyesuaian bentuk serapan atau konversi ke bentuk baku bahasa Indonesia). Hal ini memberikan sinyal adanya ambiguitas antara kesalahan tipografi murni (*mechanical typo*) dengan variasi ejaan atau ketidakpahaman terhadap standarisasi istilah. Oleh karena itu, dalam pengembangan sistem deteksi di masa mendatang, diperlukan pemisahan antara distorsi berbasis *keyboard layout* dan inkonsistensi ortografis guna meningkatkan presisi analisis kesalahan penulisan.

4.2. Analisis Kedekatan Tombol QWERTY (*Adjacency Keyboard*) dan Korelasi Motorik

Pemetaan *typing* QWERTY dalam penelitian ini berfungsi untuk mengidentifikasi pola substitusi karakter yang memiliki indikasi kuat sebagai *mechanical error* akibat *proksimitas* fisik antar tombol pada *keyboard*. *Typing* ini bertindak sebagai parameter untuk mendeteksi area *keyboard* yang rentan terhadap fenomena slip, seperti pergeseran lateral maupun diagonal jari saat proses pengetikan dengan intensitas kecepatan tinggi. Secara teoretis, ketika pasangan substitusi karakter yang berdekatan secara fisik (seperti karakter 'A' menjadi 'S', 'M' menjadi 'N', atau 'I' menjadi 'O') muncul secara repetitif, validitas interpretasi kesalahan lebih condong pada defisiensi motorik (*fat-finger error*) daripada ketidaktahuan konseptual terhadap ortografi bahasa.

Tabel 3. Pola Substitusi Huruf Tetangga pada *Keyboard* QWERTY

Pola Tombol	Jumlah	Deskripsi Pola Typo
h→y	3	*fashion → fesyen (salah satu operasinya terdeteksi h→y di proses edit)*
h→u	1	*bhs → bus*
w→s	1	*browsing → crossing*
m→n	1	*demgan → dengan*
n→m	1	*nasa → masa*
r→t	1	*poenaru → penatu*
s→d	1	*prosuktif → produktif*
a→s	1	*puataka → pustaka*
i→o	1	*memitigasi → memotivasi*
g→v	1	*memitigasi → memotivasi*

Meskipun proporsi substitusi karakter bertetangga (*neighboring substitution*) dalam *dataset* ini tidak menunjukkan dominasi kuantitatif dibandingkan pola kesalahan lainnya, kategori ini memegang peran krusial sebagai bukti empiris yang menguatkan hipotesis bahwa tata letak (*layout*) QWERTY memengaruhi tipologi kesalahan penulisan secara signifikan. Urgensi dari analisis *typing* ini tidak terletak pada besaran ukurannya, melainkan pada kemampuannya dalam mengeksplanasi mekanisme penyebab kesalahan secara spesifik (*keyboard-driven mechanism*). Secara praktis, temuan ini dapat diimplementasikan sebagai parameter fundamental dalam perancangan algoritma koreksi otomatis berbasis *adjacency* guna meningkatkan akurasi sistem dalam membedakan antara kesalahan ketik teknis dan kesalahan linguistik.

Analisis pola tipografi berbasis frekuensi dalam penelitian ini menunjukkan bahwa degradasi kualitas

linguistik dokumen didominasi oleh fenomena penghilangan karakter (*deletion*) dan substitusi non-tetangga, di mana identifikasi terhadap "zona rawan" (*vulnerability zones*) pada klaster tombol H–Y–U dan M–N menjadi bukti empiris adanya korelasi antara posisi fisik *keyboard* dengan defisiensi motorik penulis. *Typing* QWERTY ini berfungsi sebagai parameter krusial untuk mendeteksi area yang rentan terhadap pergeseran lateral maupun diagonal jari saat pengetikan cepat, sehingga kemunculan repetitif pada pasangan karakter yang berdekatan secara spasial lebih valid diinterpretasikan sebagai kesalahan mekanis (*fat-finger error*) daripada ketidaktahuan konseptual terhadap ortografi. Meskipun secara kuantitatif proporsi substitusi tetangga tidak selalu dominan, signifikansi kualitatifnya sangat tinggi dalam mengeksplanasi mekanisme penyebab kesalahan secara spesifik (*keyboard-driven mechanism*). Integrasi temuan mengenai zona rawan dan bobot probabilitas kedekatan karakter (*adjacency weight probability*) ini memberikan kontribusi strategis bagi pengembangan algoritma koreksi otomatis yang lebih presisi dalam membedakan antara distorsi mekanis-motorik dan inkonsistensi variasi ejaan baku.

4.3. Analisis Perilaku Pengetikan Berbasis Operasi *Insertion* dan *Deletion*

Analisis mengenai operasi penambahan (*insertion*) dan penghilangan (*deletion*) karakter dalam penelitian ini memberikan gambaran komprehensif mengenai profil perilaku pengetikan (*typing behavior*) pada korpus data yang diuji. Secara mekanis, frekuensi tinggi pada pola penghilangan karakter merepresentasikan fenomena skip *keystroke*, yang umumnya dipicu oleh tempo pengetikan yang melampaui limitasi perekaman *input* perangkat, tekanan tombol yang inadekuat, atau transisi motorik jari yang prematur sebelum karakter terekam secara sempurna. Sebaliknya, persistensi operasi penambahan karakter mengindikasikan gejala *double press* atau *key bounce*, di mana terjadi penekanan ganda secara involunter atau aktivasi tombol yang tidak disengaja selama fase transisi antar karakter.

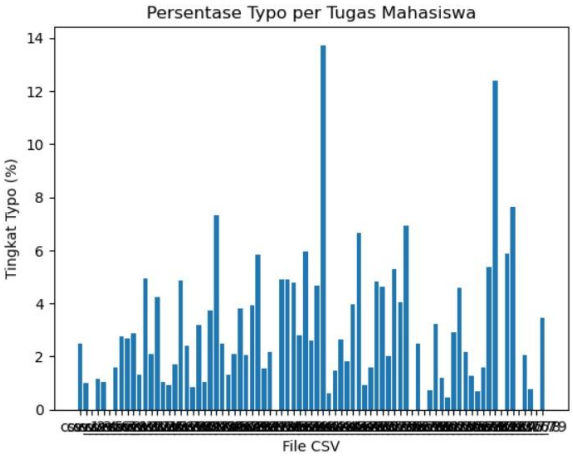
Tabel 4. Analisis Representasi Perilaku Pengetikan (*Typing Behavior*)

Operasi Karakter	Frekuensi Karakter Dominan	Fenomena Tipografi	Interpretasi Mekanis & Perilaku
Penambahan (*Insertion*)	e (6), k (4), n (3), s (2)	*Double Hit / Key Bounce*	Terjadi penekanan ganda involunter pada karakter dengan frekuensi penggunaan tinggi atau akibat sensitivitas switch keyboard pada posisi jari yang nyaman.
Penghilangan (*Deletion*)	o (7), e (5), h (4), a (4), s (3), c (3)	*Skip Keystroke*	Karakter terlewat akibat tempo pengetikan yang terlalu cepat atau tekanan tombol yang inadekuat (*incomplete stroke*) pada vokal dan konsonan umum.
Pelesapan Spasi	Karakter Spasi (n/a)	*Spacebar Discipline*	Ketidakpatuhan terhadap aturan ortografi pada kata majemuk atau kegagalan *input* pada *spacebar*, memerlukan penguatan pada *auto-correction rules*.

Sebagaimana disajikan dalam Tabel 4, frekuensi operasi karakter didominasi oleh huruf vokal (e, o, a) dan konsonan umum (s, n, k). Hal ini mengindikasikan bahwa kesalahan mekanis secara statistik berbanding lurus dengan tingkat penggunaan karakter dalam bahasa Indonesia. Fenomena *double hit* pada operasi *insertion* serta *skip keystroke* pada operasi *deletion* menunjukkan bahwa sebagian besar residu dokumen dipengaruhi oleh perilaku pengetikan cepat, bukan sekadar ketidaktahuan terhadap ejaan baku.

Temuan menunjukkan bahwa dominansi huruf yang terlibat dalam operasi *insertion*/*deletion* merupakan karakter dengan tingkat frekuensi penggunaan tinggi dalam ortografi bahasa Indonesia (seperti vokal dan konsonan umum), sehingga meningkatkan probabilitas terjadinya kesalahan mekanis secara stokastik. Secara praktis, identifikasi pola ini memungkinkan diferensiasi antara kesalahan yang berakar pada faktor motorik-kecepatan dan kesalahan konseptual berbasis variasi ejaan. Implikasinya terhadap perancangan sistem adalah tersusunnya strategi purifikasi data yang lebih presisi, dengan memprioritaskan automasi koreksi pada kasus *insertion*/*deletion* berskala minor (Jarak Levenshtein ɗ = 1) dan memberlakukan kontrol validasi yang lebih ketat pada kasus yang melibatkan perubahan ejaan kompleks atau *multi-edit*.

4.4. Analisis Distribusi Persentase Kesalahan Penulisan



Gambar 2. Grafik Sebaran Presentase *Typo* Pada Tugas Mahasiswa

Berdasarkan Gambar 2, algoritma Levenshtein Distance berbasis *wordlist* terbukti mampu mendeteksi kesalahan penulisan pada tugas mahasiswa secara konsisten. Metode ini efektif dalam mengidentifikasi kesalahan yang bersifat tipografi, seperti penghilangan huruf, penambahan huruf, dan substitusi karakter, yang umum muncul akibat proses pengetikan manual maupun hasil *paraphrase* otomatis berbasis AI. Nilai jarak edit yang kecil menunjukkan tingkat kemiripan tinggi antara kata tidak baku dan kata baku dalam *wordlist*, sehingga kesalahan dapat terdeteksi dengan baik.

Penggunaan *wordlist* sebagai basis pembanding memberikan keunggulan dibandingkan pendekatan deteksi kesalahan penulisan tanpa referensi kata baku. Dengan adanya *wordlist*, sistem mampu membedakan antara kata yang benar secara leksikal dan kata yang hanya mirip secara struktur karakter. Hal ini penting dalam konteks teks akademik mahasiswa yang sering menggunakan istilah formal dan teknis, sehingga kesalahan penulisan dapat dikenali tanpa mengandalkan pemeriksaan manual oleh dosen.

Visualisasi persentase kesalahan penulisan (*typo*) per tugas mahasiswa mengindikasikan bahwa mayoritas dokumen memiliki tingkat kesalahan yang relatif rendah, dengan prevalensi dominan berada pada rentang 1% hingga 5%. Pola distribusi ini menunjukkan bahwa kesalahan ortografis yang muncul bersifat sporadis dan masih berada dalam ambang batas toleransi untuk aktivitas produksi teks spontan tanpa proses penyuntingan (*proofreading*) yang mendalam. Sebaran data yang cukup merata pada diagram batang merepresentasikan bahwa fenomena *typo* dalam korpus ini bukan merupakan isu sistemik yang bersifat kolektif, melainkan manifestasi variasi individual yang lazim terjadi dalam proses penulisan. Secara visual, estimasi rerata tingkat kesalahan berada pada kisaran 3%–4%, yang dalam konteks penulisan akademik dikategorikan sebagai tingkat moderat namun tetap mengindikasikan urgensi peningkatan kualitas teknis penulisan.

Di sisi lain, identifikasi terhadap data pencilan (*outlier*) menunjukkan adanya sejumlah kecil dokumen dengan persentase kesalahan yang signifikan, yakni mencapai 12%–14%. Kehadiran *outlier* ini mengisyaratkan adanya segmentasi tugas yang memerlukan atensi khusus, yang kemungkinan dipicu oleh manajemen teks yang tidak optimal, laju pengetikan yang melampaui limitasi akurasi motorik, serta minimnya kontrol kualitas pasca-penulisan. Secara pedagogis, temuan ini berfungsi sebagai indikator krusial bagi perlunya intervensi instruksional, seperti pelatihan penyuntingan mandiri (*self-editing*) maupun integrasi alat koreksi otomatis berbasis algoritma. Secara komprehensif, data ini menggambarkan bahwa meskipun fenomena kesalahan penulisan merupakan residu normal dari proses produksi teks, peluang penguatan kualitas masih terbuka lebar melalui strategi pembelajaran yang berfokus pada keterampilan revisi dan peningkatan kesadaran bahasa tulis (*linguistic awareness*).

Namun demikian, metode Levenshtein Distance memiliki keterbatasan dalam mendeteksi kesalahan penulisan yang bersifat semantik. Kata yang secara ejaan benar tetapi tidak sesuai dengan konteks kalimat tidak dapat teridentifikasi sebagai kesalahan. Selain itu, penggunaan ambang batas jarak edit yang terlalu besar berpotensi menghasilkan *false positive*, yaitu kata yang sebenarnya benar tetapi terklasifikasi sebagai *typo*. Oleh karena itu, penentuan ambang batas menjadi faktor penting dalam meningkatkan akurasi deteksi.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengonfirmasi efektivitas algoritma Levenshtein Distance berbasis *wordlist* dalam mengidentifikasi kesalahan penulisan (*typographical error*) pada dokumen akademik mahasiswa secara otomatis dan terukur. Analisis terhadap 80 dokumen tugas menunjukkan bahwa intensitas kesalahan penulisan bersifat variatif dengan prevalensi antara 0,45% hingga 14,7% per dokumen. Temuan empiris mengindikasikan bahwa pola kesalahan didominasi oleh operasi *deletion*, *subtitution*, dan *insertion*, yang secara mekanis berkorelasi erat dengan perilaku pengetikan cepat serta pengaruh proksimitas tata letak *keyboard* QWERTY. Integrasi *wordlist* kata baku terbukti krusial dalam meningkatkan presisi deteksi, sehingga memungkinkan sistem membedakan antara diskrepansi ortografis dan kesalahan tipografi murni. Secara teoretis, penelitian ini memperkuat posisi metode jarak edit sebagai instrumen evaluasi linguistik yang reliabel dalam konteks penjaminan kualitas karya tulis ilmiah.

Terlepas dari kontribusi tersebut, penelitian ini memiliki batasan pada analisis yang bersifat karakter-sentris, sehingga belum mampu menjangkau anomali semantik atau struktur sintaksis yang lebih kompleks. Sebagai implikasi untuk penelitian selanjutnya, disarankan adanya integrasi antara algoritma jarak edit dengan model *Natural Language Processing* (NLP) berbasis pembelajaran mesin, seperti *Neural Language Model* atau *Transformer*, guna menghadirkan sistem koreksi yang lebih kontekstual. Selain itu, perluasan *dataset* yang mencakup variasi genre tulisan dan strata pendidikan sangat direkomendasikan untuk meningkatkan generalisasi model. Pengembangan fitur umpan balik adaptif yang bersifat pedagogis juga menjadi peluang riset potensial di masa depan, agar sistem ini tidak hanya berfungsi sebagai alat pendeteksi kesalahan, tetapi juga sebagai instrumen penguatan literasi dan kesadaran bahasa tulis mahasiswa secara berkelanjutan.

Penelitian berikutnya juga dapat menambahkan modul deteksi kesalahan spasi, deteksi kata gabungan akibat *copy-paste*, serta klasifikasi antara *typo* mekanis dan kesalahan akibat penggunaan alat parafrasa otomatis. Dengan pengembangan tersebut, sistem

deteksi typo tidak hanya berfungsi sebagai alat koreksi teknis, tetapi juga dapat menjadi instrumen pembelajaran untuk meningkatkan keterampilan penyuntingan mandiri dan literasi akademik mahasiswa.

DAFTAR PUSTAKA

[1] M. A. Pratiwi and N. Aisya, "Fenomena plagiarisme akademik di era digital," *Publ. Lett.*, vol. 1, no. 2, pp. 16–33, Jul. 2021, doi: 10.48078/publetters.v1i2.23.

[2] B. Ozfidan, D. A. S. El-Dakhs, and L. A. Alsalim, "The use of AI tools in English academic writing by Saudi undergraduates," *Contemp. Educ. Technol.*, vol. 16, no. 4, p. ep527, Oct. 2024, doi: 10.30935/cedtech/15013.

[3] P. Manorat, S. Tuarob, and S. Pongpaichet, "Artificial intelligence in computer programming education: A systematic literature review," *Comput. Educ. Artif. Intell.*, vol. 8, p. 100403, Jun. 2025, doi: 10.1016/j.caeai.2025.100403.

[4] M. Syahnaz and R. Fithriani, "Utilizing Artificial Intelligence-based Paraphrasing Tool in EFL Writing Class: A Focus on Indonesian University Students' Perceptions," *Scope J. English Lang. Teach.*, vol. 7, no. 2, p. 210, 2023, doi: 10.30998/scope.v7i2.14882.

[5] A. M. Gasaymeh and M. A. Beirat, "education sciences University Students ' Insights of Generative ArtificialGasaymeh, A. M., & Beirat, M. A. (2024). education sciences University Students ' Insights of Generative Artificial Intelligence (AI) Writing Tools. Intelligence (AI) Writing To," 2024.

[6] N. E. Mendoza Zaragoza, A. Téllez Tula, and L. Herrera Corona, "Artificial Intelligence in Thesis Writing: Exploring the Role of Advanced Grammar Checkers (Grammarly)," *Estud. y Perspect. Rev. Científica y Académica*, vol. 4, no. 2, pp. 649–683, May 2024, doi: 10.61384/r.c.a..v4i2.248.

[7] H. Balalle and S. Pannilage, "Reassessing academic integrity in the age of AI: A systematic literature review on AI and academic integrity," *Soc. Sci. Humanit. Open*, vol. 11, p. 101299, 2025, doi: 10.1016/j.ssaho.2025.101299.

[8] A. Misbullah, V. Mutiawani, J. Informatika, U. S. Kuala, and B. Aceh, "Deteksi Kata Tak Baku Dan Kesalahan Penulisan," vol. 6, pp. 130–141, 2022.

[9] T. M. Fahrudin, I. Sa'diyah, L. Latipah, I. Z. Atha Illah, C. C. Bey Lirna, and B. S. Acarya, "KEBI 1.0: Indonesian Spelling Error Detection System for Scientific Papers using Dictionary Lookup and Peter Norvig Spelling Corrector," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 12, no. 2, p. 78, Aug. 2021, doi: 10.24843/LKJITI.2021.v12.i02.p02.

[10] N. C. Dewi and A. Qoiriah, "Implementasi Algoritma Jaro-Winkler Distance dan N-Gram untuk Deteksi dan Prediksi Perbaikan Kesalahan Penulisan Kata Bahasa Indonesia pada Karya Tulis Ilmiah Mahasiswa," *J. Informatics Comput. Sci.*, vol. 2, no. 03, pp. 169–177, 2021, doi: 10.26740/jinacs.v2n03.p169-177.

[11] I. O. Suzanti, Y. Dwi Putra Negara, A. Radiah, A. Dwi Cahyani, and H. Husni, "Koreksi Kesalahan Ejaan Pada Pencarian Artikel Pariwisata Berbahasa Indonesia Menggunakan Levenshtein Distance," *J. Simantec*, vol. 13, no. 1, pp. 81–90, 2024, doi: 10.21107/simantec.v13i1.28186.

[12] I. M. A. Tresna, R. Dwiyansaputra, and H. Akhyar, "PERBAIKAN KESALAHAN KATA MENGGUNAKAN KOMBINASI JARO-WINKLER & JACCARD SIMILARITY," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 7, no. 1, pp. 25–36, Mar. 2025, doi: 10.29303/jtika.v7i1.435.

[13] A. I. Fahma, I. Cholissodin, and R. S. Perdana, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 53–62, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/690

[14] D. Darmanto, K. Kharisma, A.-R. Muhammad, and E. Wahyudi, "Perbandingan Kinerja Algoritma Jaro-Winkler dan Levenshtein Distance untuk Deteksi Kesalahan Penulisan Bahasa Indonesia pada Karya Ilmiah Mahasiswa," *J. Rekayasa Teknol. Inf.*, vol. 9, no. 2, p. 112, 2025, doi: 10.30872/jurti.v9i2.19134.

[15] Yuslena, H. Khatimi, and R. A. Fajrin, "Deteksi Plagiarisme Menggunakan Algoritma Levenshtein Distance," *J. Teknol. Inf. Univ. Lambung Mangkurat*, vol. 6, no. 1, pp. 31–38, 2021, doi: 10.20527/jtiulm.v6i1.66.

[16] M. F. Azhri, D. Swanjaya, and R. K. Niswatin, "Penerapan Algoritma Levenshtein Distance pada Aplikasi Asisten Guru Bahasa Inggris," *Semin. Nas. Inov. Teknol.*, pp. 155–160, 2019, [Online]. Available: file:///G:/My Drive/Strata 1 UNUSIDA/Bimbingan TIF Gasal 2024-2025/Aisya/Referensi Bacaan/530-Article Text-1300-1-10-20201102.pdf

[17] M. Lihawa, A. D. Hartanto, N. Norhikmah, D. Prabowo, I. N. Fajri, and W. Widayani, "Implementation of the Levenshtein Distance Algorithm and the Regular Search Expression Method for Detecting Typors in Javascript," *Sistemasi*, vol. 12, no. 2, p. 452, 2023, doi: 10.32520/stmsi.v12i2.2795.

[18] M. M. Yulianto, R. Arifudin, and A. Alamsyah, "Autocomplete and Spell Checking Levenshtein Distance Algorithm To Getting Text Suggest Error Data Searching In Library," *Sci. J. Informatics*, vol. 5, no. 1, p. 75, 2018, doi: 10.15294/sji.v5i1.14148.

[19] N. N. Syarafina and J. F. Palandi, "Designing a word recommendation application using the Levenshtein Distance algorithm," *Matrix J. Manaj. Teknol. dan Inform.*, vol. 11, no. 2, pp. 63–70, 2021, doi: 10.31940/matrix.v11i2.2419.

[20] Wikidepia, "indonesian_datasets." GitHub, Feb. 2026.

[21] N. A. Sekartaji and R. Arifudin, "Implementation of Raita Algorithm in Manado-Indonesia Translation Application with Text Suggestion Using Levenshtein Distance Algorithm," *Recursive J. Informatics*, vol. 2, no. 2, pp. 88–96, Sep. 2024, doi: 10.15294/rji.v2i2.73651.

[22] A. Febrianti Azzaroh, A. Surya Editya, and A. Khoir Al-Haq, "Sistem Penilaian Esai Otomatis Berbasis Web Menggunakan Model BERT dan Levenshtein Distance," *J. Pustaka AI (Pusat Akses Kaji. Teknol. Artif. Intell.*, vol. 5, no. 2, pp. 142–148, Aug. 2025, doi: 10.55382/jurnalpustakaai.v5i2.1022.

[23] "Introduction to Levenshtein distance," Geeks For Geeks.

[24] L. Yujian and L. Bo, "A Normalized Levenshtein Distance Metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007, doi: 10.1109/TPAMI.2007.1078.

[25] H. Gong, Y. Li, S. Bhat, and P. Viswanath, "Context-Sensitive Malicious Spelling Error Correction," in *The World Wide Web Conference*, New York, NY, USA: ACM, May 2019, pp. 2771–2777. doi: 10.1145/3308558.3313431.

[26] R. Alfred and R. W. Teoh, "Improving Topical Social Media Sentiment Analysis by Correcting Unknown Words Automatically," 2019, pp. 299–308. doi: 10.1007/978-981-13-3441-2_23.

[27] S. J. Greenhill, "Levenshtein Distances Fail to Identify Language Relationships Accurately," *Comput. Linguist.*, vol. 37, no. 4, pp. 689–698, Dec. 2011, doi: 10.1162/COLI_a_00073.