

Research on the Construction Method of Tibetan Syllable Confusion Sets for Text Correction

Ga Zang Cai Rang, Dingguo Gao*, Jinyi Li, Liyuan Zhou

School of Information Science and Technology, Xizang University, Model Base for Training Innovative Talents in Tibetan Information Technology,
Lhasa, Xizang 850000, China

Keywords: Text correction; Real-word errors; Tibetan syllables; Tibetan grammar; Confusion set

Citation: Ga Zang C.R., Gao D.G., Li J.Y., et al.: Research on the Construction Method of Tibetan Syllable Confusion Sets for Text Correction. Data Intelligence, 2026, Vol. 8, Art. No.: 20250181. DOI: <https://doi.org/10.3724/2096-7004.di.2025.0181>

ABSTRACT

Error confusion sets refer to collections of words or characters that, within the same group, are similar in spelling and grammar to the corresponding words or characters in the correct text, thereby easily causing confusion. Such confusion sets are commonly used in automatic text correction systems, where the system replaces potential errors with other candidates from the confusion set to enhance the accuracy of the correction. This paper introduces a method for constructing Tibetan syllable confusion sets aimed at text correction, and validates the confusion sets in experiments on Tibetan text correction. The Tibetan syllable confusion sets presented in this paper are composed of three distinct subsets based on the characteristics of Tibetan text: a confusion subset based on Tibetan syllable component information and edit distance, a confusion subset based on input key position distance, and a confusion subset based on Tibetan grammatical knowledge. These subsets are merged sequentially to form a comprehensive Tibetan syllable confusion set for text correction, totaling 59.2M in size. Experiments on public data with this confusion set have shown that the accuracy of the BERT (Bidirectional Encoder Representations from Transformers) + confusion set correction method was 10.63% higher than that of a standalone BERT model, demonstrating the effectiveness of the Tibetan syllable confusion set in Tibetan text correction tasks.

1. INTRODUCTION

Confusion sets refer to collections of common words or characters that may be mistyped or misrecognized. Within the same group, these words or characters are similar in spelling and grammar to the correct words or characters in the original text, thereby easily causing confusion. As a fundamental

*Corresponding author: Dingguo Gao (E-mail: gdg@utibet.edu.cn; ORCID: 0009-0005-7995-6009).

data type, error confusion sets are often used in text proofreading for correcting real-word errors. Research on automatic Tibetan text proofreading started early, with Tashi Tsering [1] developing a DOS-based Tibetan spelling check system in the 1990s based on Tibetan spelling and grammar. With the evolution of technology, Tibetan automatic proofreading has gone through stages including rule-based and statistical methods, machine learning, and deep learning. However, most research has focused on error detection rather than correction, making the construction of Tibetan confusion sets for automatic text correction a relatively unexplored yet practically significant research direction.

Errors in text can be divided into real-word errors and non-real-word errors. Real-word errors refer to words in the text that are spelled correctly but do not fit the context, while non-real-word errors refer to characters that do not conform to the spelling rules of the language. To address different types of errors, distinct correction strategies are necessary, and error confusion sets serve as an effective solution for real-word errors. The confusion set design combined with a pointer network for Chinese text correction by Li Jiacheng et al. [2] achieved a 39.06% accuracy rate on the NLPCC2018 evaluation dataset, marking the best performance at the time and demonstrating the effectiveness of confusion sets in correction tasks. According to existing re-search, methods for constructing confusion sets can be categorized into three types: dictionary-based, language feature-based, and corpus-based approaches. Chen Keliang [3] used homophones and visually similar word dictionaries to search for similar characters and then arranged and combined them using a likelihood matching algorithm to construct a Chinese character confusion set aimed at correction tasks. Shi Hengli et al. [4] built a seed confusion set using linguistic knowledge of words and characters on top of a dictionary and then expanded it using a self-extension algorithm to generate a Chinese character confusion set. Pedler J and others [5] utilized large-scale corpora to count the number of mistakes between characters and constructed an English correction-oriented confusion set based on this data. While most research has focused on constructing confusion sets for languages like English and Chinese, there have been no publicly published results on the construction of Tibetan confusion sets. Although the confusion set construction methods for other languages may provide some reference for Tibetan, they cannot be directly applied. This paper, taking into account the grammatical features of Tibetan and using Tibetan syllables as the unit, designs a multi-information, multi-strategy Tibetan confusion set construction method for text correction tasks and validates it in correction experiments.

2. METHOD FOR CONSTRUCT TIBETAN SYLLABLE CONFUSION SETS

In order to better handle multi-scenario Tibetan text correction tasks, this paper separately constructs confusion subsets for Tibetan syllables based on near-homophonic characters, commonly mistyped characters, verbs, and function words, which are then combined to form the final Tibetan syllable confusion set. The subset of confusions between near-homophonic characters is constructed using the component differences and edit distance information between Tibetan syllables. The subset for commonly mistyped characters is constructed based on the input code coordinate distance of Tibetan syllables on a standard keyboard. The confusion set for verbs and function words is formed through statistical analysis of actual practice. The overall construction process is illustrated in Figure 1. During the design of the

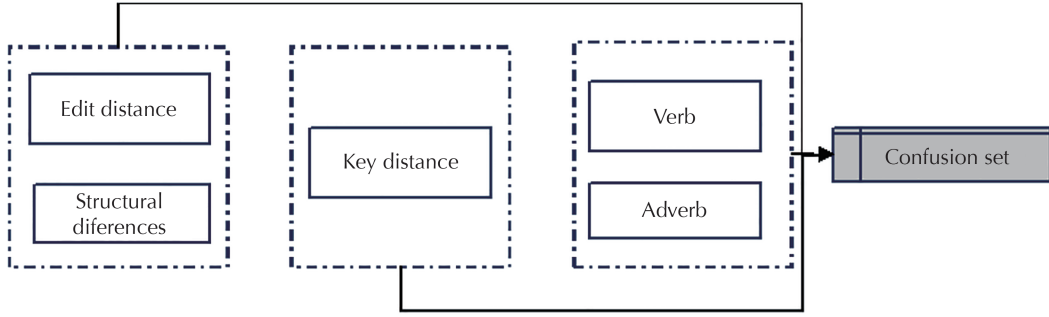


Figure 1. The construction method of Tibetan syllable confusion set for text error correction 2.1 Related Mathematical Concepts.

confusion set construction method, this paper employs some mathematical concepts, briefly introduced in conjunction with natural language processing as follows.

Levenshtein Edit Distance is used to calculate the similarity between strings by determining the minimum number of editing operations required to transform String A into String B. The editing operations include insertion, deletion, and substitution [6].

$$D(x, y) = \begin{cases} 0 & \text{if } i = 0 \text{ and } j = 0 \\ i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \begin{pmatrix} D[i-1][j] + 1 \\ D[i][j-1] + 1 \\ D[i-1][j-1] + \text{cost} \end{pmatrix} & \text{otherwise} \end{cases} \quad (1)$$

In Formula 1, the Levenshtein Edit Distance (referred to as edit distance hereafter) represents the distance between the first i characters of a string and the first j characters of another string. When both i and j are 0, the value is 0; when either i or j is 0, the value of D is the length of the other string. When both strings are non-empty, the value of D is calculated by selecting the minimum among $D[i-1][j] + 1$ (insertion), $D[i][j-1] + 1$ (deletion), and $D[i-1][j-1] + \text{cost}$ (substitution). Here, cost represents the cost of substitution. If the character at position i in string x and the character at position j in string y are identical, cost is 0; otherwise, cost is 1. Edit distance is widely used in fields such as spell checking and genetic sequence alignment.

The Hamming Distance is used to measure the bitwise difference between two strings of equal length. It quantifies the difference between them by counting the number of differing characters at corresponding positions in the two strings [7].

$$H(x, y) = \sum (x_i \neq y_i) \quad (2)$$

In Formula 2, $H(x, y)$ represents the Hamming Distance between strings x and y . Here, x and y are two strings of equal length, and x_i and y_i respectively denote their characters at the i th position. $H(x, y)$ equals the total number of differing characters at corresponding positions between x and y . Hamming Distance finds wide applications in fields such as error detection and error-correcting codes.

The Euclidean Distance is used to calculate the spatial distance between two vectors or coordinates. It measures the straight-line distance between two points in Euclidean space.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (3)$$

In Formula 3, $d(x, y)$ represents the Euclidean Distance between coordinates x and y . Here, x and y are two n -dimensional coordinate vectors, where $(x_1, x_2, \dots, x_n)$ and $(y_1, y_2, \dots, y_n)$ are the values of their respective dimensions. A smaller Euclidean Distance indicates that the two vectors are closer to each other. Euclidean Distance finds wide applications in fields such as machine learning and data mining.

2.1 Construction of Tibetan Syllable Phonetically Similar Confusion Subsets

In phonetic scripts, the higher the similarity between the constituent letters of two words, the closer their pronunciations are. Tibetan, as a phonetic script, exhibits this characteristic prominently. For instance, the syllables “དགེ་ཉལ་མེད་” and “དགེ་ཉལ་མེད་” have similar written pronunciations due to the shared consonant letters “ད” and “ལ”. However, some Tibetan syllables utilize the same letters but have significantly different pronunciations. For example, “དགེ་ཉལ་མེད་” and “དགེ་ཉལ་མེད་” both contain the consonant letters “ད” and “ལ”, yet their pronunciations are entirely different. This disparity arises from the different positional components occupied by the consonant letters “ད” and “ལ” in the two syllables. A positional component refers to the position occupied by a Tibetan consonant letter within a syllable. Tibetan syllables are composed of seven consonant components (base, prefix, superscript, subscript, postfix, postfix postfix, and vowel sign) [8] and one vowel sign. The consonant letters have varying effects on the pronunciation of Tibetan syllables depending on their positional components. Generally, the base component determines the overall tone of the syllable, followed by the superscript and subscript components. Therefore, when constructing a confusion subset of phonetically similar Tibetan syllables, it is necessary to consider not only the number of shared letters between syllables but also the positional component information of the letters within the syllables. Hence, this paper proposes a method for constructing phonetically similar confusion subsets of Tibetan syllables that effectively reflects the aforementioned characteristics of Tibetan syllables, utilizing Levenshtein distance and Hamming distance. The specific framework is outlined in Figure 2, implemented in three steps.

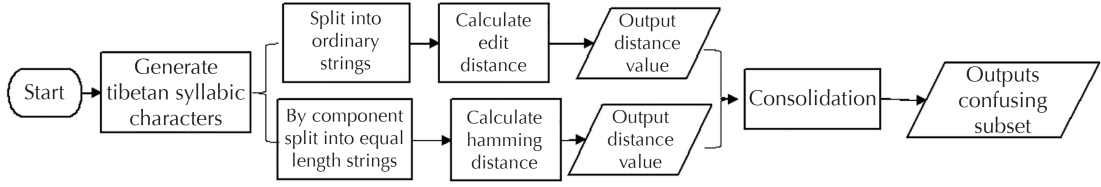


Figure 2. Framework for constructing a subset of Tibetan syllable sound proximity confusions.

Step 1: Calculate edit distance between Tibetan syllables.

For each Tibetan syllable, select the top N Tibetan syllables with the smallest Levenshtein distance. The specific process is as follows:

1. Data Preparation:

Generate a dataset of Tibetan syllables based on the method outlined in [9], resulting in a dataset containing 18,785 syllables.

2. Syllable Transformation:

Utilize the Wylie transliteration method to convert each syllable into its corresponding regular string form. For example, transform “�ྲ” to “kr”.

3. Calculate Levenshtein Distance:

For each Tibetan syllable, calculate its Levenshtein distance with every other syllable in the dataset using Formula 1.

4. Sorting:

Based on the computed Levenshtein distances, match each syllable with a confusion set. Sort the syllables in descending order of Levenshtein distance and select the top N syllables.

After completing Step 1, each syllable is paired with the N closest syllables in terms of Levenshtein distance, forming the confusion set for each syllable.

Step 2: Calculate the Hamming Distance between Tibetan syllables.

The Levenshtein distance calculated in the first step only reflects the similarity of letters between syllables and fails to capture the positional component information of letters within syllables. For instance, consider the confusion set with the key value of “ཁས”/gn/, where the candidate syllables ཁས/gu/ and ཁས/swoo/ have an equal Levenshtein distance of 2 from the key value འཁས/gn/. However, in actual pronunciation, ཁས/swoo/ differs more from ཁས/gn/ than ཁས/gu/ does. This is because in འཁས/gn/, “ཁ” serves as the base component and “ས” as the postfix, while in ཁས, “ཁ” is the prefix and “ས” is the base component. This illustrates that the

Levenshtein distance alone cannot capture the positional component information of Tibetan syllables. Therefore, this paper proposes a method based on Hamming distance to calculate the differences in components between Tibetan syllables. The calculation steps are largely similar to those of the first step, with the main difference lying in the syllable splitting and bitwise calculation of Hamming distance.

Hamming distance calculates the total number of differing characters at corresponding positions between two strings. It requires Tibetan syllables to be split into equally long component codes based on their component positions, ensuring that letters occupying the same component positions are placed at corresponding positions. The syllable splitting method used in this paper is the Tibetan component recognition method proposed by Gao Dingguo et al. [10]. After splitting the syllables, they are transcribed into binary component codes of length 8. The specific method is as follows:

1. Generate all Tibetan syllables.
2. Combine the Tibetan component recognition method to split Tibetan syllables into components and save them in a component information table.
3. Prepare a string of zeros of length 8, where each position in the string represents a component: prefix, superscript, base, subscript, postfix, postfix postfix, vowel, and post-postfix.
4. Query the component information for each syllable in the component information table: If the letter is present in the i -th component of the syllable, change the i -th position of the string of zeros to 1. Otherwise, leave the i -th position of the string of zeros as 0.
5. Generate binary component codes of length 8 for all Tibetan syllables. See Table 1 for partial details.

Table 1. Binary Component Codes of Tibetan Syllables.

	prefix	superscript	base	subscript	Suffixes of Suffixes	vowel sign	postfix	postfix of postfix	Binary Component Code
བསགས	1	0	1	0	0	0	1	1	10100011
སགས	0	0	1	0	0	0	1	1	00100011
བགས	1	0	1	0	0	0	1	0	10100010

Once the binary component codes for all Tibetan syllables are obtained, Hamming distance between the binary component codes of syllables is calculated using Formula 2. This reflects the differences in components between Tibetan syllables. When the Hamming distance is 0, it indicates that the component information of two syllables is identical, as seen in the example of “དཀ”/da:/ and “ཀླ”/ma:/. After completing the calculation of Hamming distance between syllables, to facilitate subsequent merging, each syllable is matched with the N (same as Step 1) syllables with the closest Hamming distance as the confusion subset.

Step 3: Adding Levenshtein Distance and Hamming Distance values between syllables

$$T(x, y) = D(x, y) + H(x, y) \quad (4)$$

In Formula 4, D represents the final distance between Tibetan syllables x and y , where E and H respectively represent the Levenshtein Distance and Hamming Distance. The sum of the distances between syllables is calculated, and the syllables in the confusion set are sorted accordingly.

Before performing the calculation, the confusion sets generated in the first and second steps are intersected, retaining only the overlapping parts for addition. In Figure 3, taking the confusion set of the Tibetan syllable “མན” /mɑ:n/ as an example, when sorting the confusion set based solely on the Levenshtein Distance values, “མནན” /nɑ:n/ appears before “དམན” /dmɑ:n/. However, both “མན” /mɑ:n/ and “དམན” /dmɑ:n/ share the same base letter “མ” and have similar pronunciations. On the other hand, “མནན” /nɑ:n/ has a base letter of “ན”, which has a significantly different pronunciation from the key syllable. This arrangement does not adhere to the principle of placing phonetically similar syllables first. After the calculation, it can be observed that the syllables with the base letter “མ” are placed before those with the base letter “ན”, resulting in a higher fidelity in retaining phonetic information between Tibetan syllables.

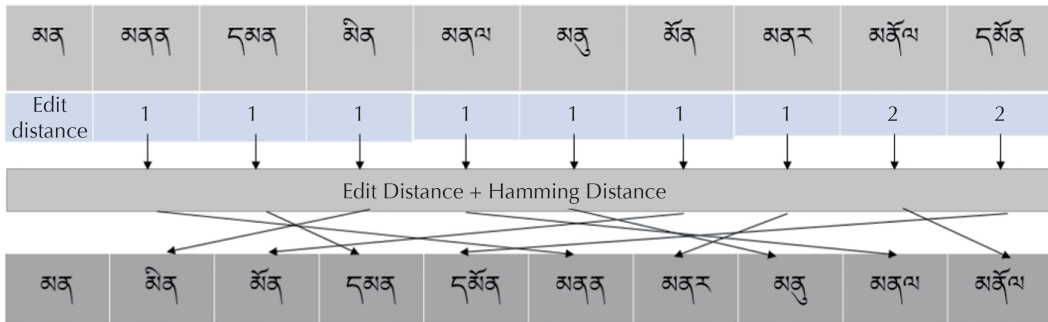


Figure 3. Example of confusion set $T(x, y)$ processing for syllable མན /mɑ:n/.

2.2 Construction of Tibetan Syllable Confusion Subsets for Error-Prone Entries.

With the advancement of computer technology, electronic Tibetan texts such as news and blogs are being directly input via keyboards, making it essential to construct confusion sets for error-prone syllables tailored to electronic text correction. Common Tibetan input methods include the Sambhota Tibetan Input Method, the Wylie Transliteration Input Method, and the Himalaya Tibetan Input Method, among others. Initially, Tibetan web pages primarily used the Sambhota font, but the release of Windows Vista by Microsoft in 2007, which included a built-in Tibetan font, significantly impacted the digitalization of Tibetan information. This font, being in line with the international Unicode standard, led many Tibetan websites in China to adopt it. By 2011, web pages using the Microsoft Himalaya font had surpassed those using the Sambhota Tibetan font, accounting for 52.17% of all Tibetan web pages [11]. Furthermore, its integrated Microsoft Himalaya Tibetan input method replaced the various Tibetan keyboard input methods,

becoming the mainstream choice. This paper focuses on the construction of a confusion set for commonly mistyped Tibetan syllables, based on the Euclidean distance between the input codes of Tibetan syllables on a standard keyboard layout as measured by the Himalaya input method. The process is illustrated in Figure 4 and is implemented in three steps:

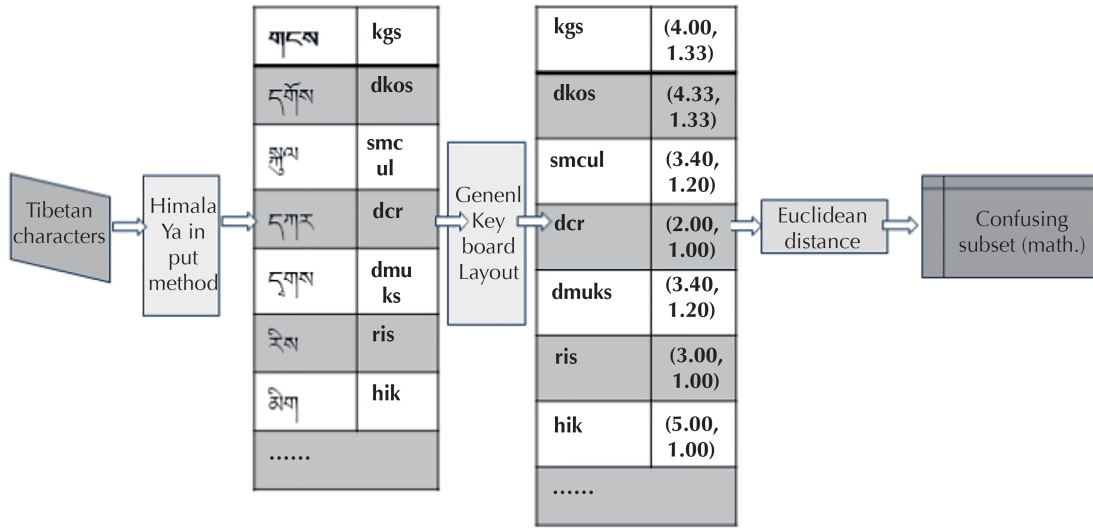


Figure 4. Construction of error-prone confusing subset of Tibetan syllables.

Step 1: Generating Key Codes for Syllables.

The Himalaya input method utilizes OpenType processing technology, using the “M” key to input stacked Tibetan syllables, making input simple and user-friendly, and aligning well with Tibetan characteristics. In this paper, following the Himalaya input method, key input codes were generated for each syllable (see Table 2).

Table 2. Common Input Codes for Tibetan Syllables.

Syllable	input code	Syllable	input code	Syllable	input code	Syllable	input code
ཀ	k	ཀགས	fkgs	ཀམས	fsmkmrg	སྐུལ	smkmrg
ཀད	kg	ཀམས	fsmkg	ཀམས་སྐུལ	fsmkmrgs	ཀམས་སྐུལ་སྐུལ	fsmkmregs

Step 2: Calculating the Key Coordinates of Syllables on a Standard Keyboard.

The coordinates are represented using a two-dimensional coordinate system, where (x, y) denotes the key’s position in the x-th row and y-th column. For instance, Q(2, 2) represents the key “Q” located in the 2nd row and 2nd column. Similarly, the coordinates of some keys on a standard keyboard are shown in Table 3.

Table 3. Partial Key Coordinates.

keypad	coordinates	keypad	coordinates	keypad	coordinates
Q/q	(2, 2)	W/w	(2, 3)	E/e	(2, 4)
A/a	(3, 2)	S/s	(3, 3)	D/d	(3, 4)
Z/z	(4, 2)	X/x	(4, 3)	C/c	(4, 4)

The input code for a Tibetan syllable is generally composed of multiple keys. Using the following formula (Formula 5), we can calculate the overall coordinates of the syllable using the coordinates of individual keys in the input code.

$$K(X, Y) = \begin{cases} (x_1 + x_2 + \dots x_n) / n \\ (y_1 + y_2 + \dots y_n) / n \end{cases} \quad (5)$$

In Formula 5, $K(X, Y)$ represents the coordinates of the keys in the syllable, where n represents the number of keys in the syllable input code. $x_1 \dots x_n$ represents the coordinates of each key in the input code. For example, if the input code for the syllable “ཅེས” is “des”, then the key coordinates would be $K(2.66, 3.66) = \begin{cases} (3 + 2 + 3) / 3 \\ (4 + 4 + 3) / 3 \end{cases}$.

Step 3: Calculating the Key Distance Between Syllables using Euclidean Distance

After the above two steps, with the coordinates of each syllable obtained, we can utilize Formula 3 to compute the Euclidean distance between syllable coordinates, and select the top N syllables with the smallest distances to form a confusion set for easily mistyped syllables.

2.3 Construction of Confusion Subset Based on Grammatical Knowledge

In addition to the aforementioned errors, Tibetan real-word errors also include errors in verb tenses and collocations of particles [12]. Errors in verb tenses refer to the misuse of verb tenses in the context of a sentence. For instance, in “སང་ཉིན་ལྷ་སར་སོང་།” (going to Lhasa tomorrow), the verb “སོང་” (going) should use the future tense “འོང་” (will go). Tibetan linguists of past generations have refined and summarized the changes in verb tenses, which can be directly utilized. This paper uses the “Table of Tenses (བྱུང་ལུགས་ཀྱི་ཁྱུ་ལྟུ་ལྟུ་དང་བཅས་པའི་དཔེ་བཞུང་)” compiled by Nase Doje Zangbu, obtaining 892 different forms of monosyllabic verbs as the confusion set, with the number of confusion words for a single syllable not exceeding 4.

Errors in particles can be divided into two types:

(1) Incorrect usage of particles that do not fit the context.

Example sentences:

Sentence 1: “བཀྲ་ཤིས་ཀྱིས་དཔེ་ཆར་བརྟམས།” (Zaxi is reading.)

Sentence 2: “བཀྲ་ཤིས་ཀྱི་དཔེ་ཆར་བསྟན།” (Reading Zaxi’s book.)

In sentences 1 and 2, the other components remain unchanged, but the change from the ergative particle (ཟླ་ལྟོ་སྟེང་།) to the genitive particle (འབྲེལ་སྟེང་།) has caused a change in meaning.

(2) Incorrect collocation of non-freely used particles.

The “Grammar of Thirty Verses” has strict regulations on the collocation of non-freely used particles. Incorrect collocations of particles that do not conform to grammar result in incorrect sentences that cannot convey meaning.

Example sentence:

“བཀྲ་ཤིས་གིས་དཔེ་ཆར་བསྟན།”

In sentence 3, the second syllable of the name “བཀྲ་ཤིས་” (Zaxi) has the postfix “ས”. If the ergative particle is used according to the grammar, “ཀྱིས” should be used. However, “གིས” in the example sentence is a collocation error.

The frequency of confusion between particles obtained from real corpus statistics more accurately reflects actual misuse scenarios. This paper collected 96 Tibetan compositions from primary and secondary schools in Maqu County, Gansu Province, and Aba County, Sichuan Province, and manually counted 234 cases of particle collocation errors. A confusion matrix of particle collocations was established by restoring the misuse frequencies between each particle, as shown in Table 4.

Table 4. Confusion Matrix of Particle Collocations (Partial).

Number of misuse	null	གིས	ཀྱིས	ཡིས	ན
null	0	1	3	2	0
གིས	1	0	4	0	0
ཀྱིས	0	5	0	4	0
ཀྱིས་བཀྲ་ཤིས་སྟེང་།	2	2	3	0	6
ཡིས	2	0	0	5	0

In Table 4, rows represent correct particles, while columns denote the objects misused in the compositions. For instance, the entry in the first row and the second column (blank, གིས) = 1 indicates one instance of the redundant insertion of the particle གིས in the compositions. Similar interpretations apply to other entries. Utilizing the aforementioned confusion matrix, errors in the usage of 70 particles in the corpus were identified. Based on the frequency of confusion, confusion sets ranging from 2 to 13 in length were constructed for the 70 particles.

2.4 Construction of Tibetan Syllable Confusion Set Based on Hybrid Strategies

After establishing confusion subsets for Tibetan syllables based on the three methods mentioned above—namely, for phonetic similarity, ease of mistyping, and errors in particle and verb usage—the final confusion set was generated by merging duplicates based on the same key values. N was set to 100 to limit the size of the confusion set, representing the top 100 syllables closest in terms of edit distance, Hamming distance, and Euclidean distance. The format of the confusion set constructed in this paper is JSON, with the final confusion set totaling 59.2 MB in size. For further details, please refer to Table 5.

Table 5. Confusion Set Information.

Item	Size/word count
Average Confusion Set Length	192.7
shortest note	125
longest note	200
total number of words	241390

The confusion set generated in this paper follows the conventional format of confusion sets [13], where the queried syllable serves as the key, and the corresponding confusing syllables serve as the values, thereby forming pairs of syllable confusions. One key corresponds to several values (refer to Table 6 for details).

Table 6. Example of Confusion Set.

Target syllable	Candidate syllable
ཀལ	ཀོལ བཀལ ལྟལ ལྟལ
ཁལ	མཁལ ཁོལ འཁལ
གལ	མགལ དགལ ལྟལ
ངལ	ང ལྟལ མངལ

3. EXPERIMENTS AND ANALYSIS

3.1 Dataset

The original training set serves as the experimental training set for this paper, while the original development set is used to create the test set for this study. The Tibetan error statistics used in the experiments are sourced from the laboratory, obtained through the comparison against correct texts by six Bachelor's degree students and two Master's degree students, by simulating various real-world writing and keyboard input errors. Based on the statistics of Tibetan script errors and their corresponding frequencies, operations such as insertion, deletion, or replacement of Tibetan components are randomly applied to the

syllables identified in the correct text according to weighted sampling, thereby generating incorrect Tibetan syllables (true word errors) that are then substituted into the correct text.

Before inputting into the model, each line of text in the training and test sets undergoes preprocessing, which mainly includes: (1) Removing characters other than Tibetan script from the text. (2) Replacing Tibetan numeral characters with separators. (3) Converting multiple separators into a single separator. (4) Consolidating multiple standalone Tseg symbols into one. (5) Excluding lines with fewer than four syllables. (6) Adding separators before standalone Tseg symbols if not already preceded by a separator. Finally, the text is segmented into tokens (syllables) using separators as de-limiters and transformed into input sequences. During training, the text of the training set is randomly masked with the Tibetan syllable as “[MASK]” according to the original BERT [14] masking probability for the Masked Language Model (MLM) task. The erroneous parts of the test set are also replaced with the masked token “[MASK]”. The detailed data specifications are shown in Table 7, with examples from the test set provided in Table 8.

Table 7. Dataset Statistics.

Train set	29890
Test set	4979

Table 8. Examples from the Test Set.

Word Entry	Correct Element (Syllable)	Incorrect Element (Syllable)
ཟམ་ [MASK] འཕྲོ་བཞིན་ཡོད།	པར	པ

3.2 Experimental Parameters and Evaluation Metrics

To mitigate overfitting, this study utilized a smaller version of DistilBERT [15], referred to as BERT-small (a distilled version with half the number of layers compared to the original BERTbase network), as the experimental model [15]. The parameters are listed in Table 9.

Table 9. BERT Model Parameters

hidden_size	768
token_max_length	300
num_hidden_layers	6
weight_decay	0.01
learning_rate	2e-6

The role of the confusion set proposed in this paper mainly manifests in the error correction part after detecting errors, as indicated by the metric in Formula 6.

$$\text{Correction Accuracy} = \frac{\text{Corrections correct}}{\text{All errors in the text}} \times 100\% \quad (6)$$

3.3 Results and Analysis

This paper compares the text correction performance of the BERT model used alone versus the BERT model supplemented with the confusion set proposed in this paper as a selection criterion. When correcting texts using only the BERT model, the token (syllable) ranked highest by the model's scoring is considered the correct token (syllable). After incorporating the confusion set, the process begins by retrieving the potentially correct token (syllable) from the confusion set dictionary based on the erroneous token (syllable) as a candidate list. If the score of the BERT model's top-ranked token is below a confidence threshold of 0.7, empirically determined based on validation experiments, and the predicted correct token is not in the candidate list, the model skips this token and selects the next-highest-scoring token, repeating the aforementioned assessment until a correct token is identified.

The experimental results, as shown in Table 10, indicate that using the confusion set proposed in this paper as a filter for selecting the correct tokens (syllables) can improve the BERT model's correction accuracy by 10.63%, demonstrating that the construction of this confusion set can indeed enhance the performance of text correction tasks.

Table 10. Text Correction Experiment.

BERT	BERT + Confusion Set
78.81%	89.44%

4. CONCLUSION

To better study the correction of Tibetan syllables, this paper combines existing methods for constructing confusion sets in other languages with the unique linguistic rules of Tibetan, designing a mixed-strategy, multi-method approach to constructing Tibetan syllable confusion sets. Comparative correction experiments show that the confusion set constructed in this study significantly improves the accuracy of Tibetan text correction tasks. This indicates that researching confusion sets and correction methods for more complex Tibetan correction tasks at the word, sentence, and paragraph levels is both meaningful and valuable.

AUTHOR CONTRIBUTIONS

GaZangCaiRang was responsible for the overall conception and writing of the paper. GaoDingGuo oversaw the overall control, content review, and editing of the manuscript. LiJinYi conducted the

experimental section. ZhouLiYuan handled data organization and collection. Each author's specific contributions were crucial in the development of this high-quality research paper.

ACKNOWLEDGEMENTS

We sincerely thank Maqu Tibetan Middle School in Gannan Prefecture and Aba Middle School for providing the samples of student compositions.

REFERENCES

- [1] Z. Ciren, *Design of a Tibetan Spell Checking System*. Beijing, China: Tsinghua University Press, 1998.
- [2] J. C. Li, J. Y. Shen, C. Gong, et al., "Chinese grammar correction based on pointer network integrating confusion set knowledge," *J. Chinese Inf. Process.*, vol. 36, no. 4, pp. 29–38, 2022.
- [3] K. L. Chen, "Research on key technologies of intelligent processing of data resources for digital publishing," Ph.D. dissertation, Beijing Univ. Posts Telecommun., Beijing, China, 2021.
- [4] H. L. Shi, L. L. Liu, S. Wang, et al., "Research on the construction method of Chinese character seed confusion set," *Comput. Sci.*, vol. 41, no. 8, pp. 229–232, 2014.
- [5] J. Pedler and R. Mitton, "A large list of confusion sets for spellchecking assessed against a corpus of real-word errors," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, 2010.
- [6] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Dokl.*, vol. 10, no. 8, pp. 707–710, 1966.
- [7] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.
- [8] D. G. Gao and Y. C. Gong, "Research on attribute statistics of modern Tibetan character set," *J. Chinese Inf. Process.*, vol. 19, no. 1, pp. 71–75, 2005.
- [9] D. G. Gao, J. Y. Li, and J. W. Wang, *Automatic Analysis of Modern Tibetan syllable*. Chengdu, China: Southwest Jiaotong Univ. Press, 2022.
- [10] B. Wu and D. Jiang, "Development and technical application of Tibetan international encoding," *J. Southwest Minzu Univ. (Humanities Social Sci.)*, vol. 32, no. 8, pp. 46–51, 2011.
- [11] S. M. Cuo, Z. J. Cai, R. Z. Cai, et al., "Analysis of Tibetan text spelling error types," *J. Northwest Minzu Univ. (Natural Sci.)*, vol. 43, no. 1, pp. 26–31, 2022.
- [12] J. J. Xu, G. D. Huang, and C. X. Ma, "Chinese text proofreading model based on RoBERTa-wwm-ext and confusion set," *J. Hubei Univ. (Natural Sci.)*, vol. 45, no. 5, pp. 712–718, 2023.
- [13] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
- [14] V. Sanh, L. Debut, J. Chaumond, et al., "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.
- [15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

AUTHOR BIOGRAPHY

Dingguo Gao (1972–), Corresponding Author, Male, Ph.D. Candidate, Professor, Senior Member of CCF (27174S), research interests are natural language processing.

Ga Zang Cai Rang (1993–), first author, male, PhD student, CCF member (R4016M), research interests are Tibetan Computational Linguistics.