ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 5, pp. 91-104
DOI:10.33168/JSMS.2022.0506

Typographic Error Identification and Correction in Chatbot Using N-gram Overlapping Approach

Kalaiarasi Sonai Muthu Anbananthen[1], Subramaniam Kannan[1], Mikail Muhammad Azman Busst[1], Saravanan Muthaiyah[2], Saravanan Nathan Lurudusamy[3]

[1] Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia

[2] Faculty of Management, Multimedia University, Selangor, Malaysia

[3] Division Consulting & Technology Services, Telekom Malaysia

kalaiarasi@mmu.edu.my (Corresponding author)

Abstract. The high demand in the business sector and Artificial Intelligence (AI) capability have led to the development of chat robots, or in short, chatbots. A chatbot interacts through instant messaging, artificially replicating the patterns of human interaction. It is a computer program or virtual agent that allows humans and machines to freely converse using Natural Language Processing (NLP). People may input their queries with various typographical errors when interacting with a chatbot. The typographical errors include misspelt words or using abbreviated words. The primary drawback of most existing chatbots is that they can only handle questions with correct sentences. Natural language processing alone is insufficient for detecting typographical problems in input queries. Although the typographical error checker has become one of the most commonly used features in many applications and programs, including web applications, crawlers, and web browsers, in chatbots, especially in "Manglish", it still does not exist. Therefore, this research aims to enable chatbots to respond to queries correctly even with typographical errors using an embedding model of the N-gram overlapping with a rule-based algorithm.

Keywords: chatbots, n-gram overlapping, Manglish, typographical-error.

1. Introduction

The majority of Malaysians use bilingualism or code-switching in their daily lives. Code-switching occurs when more than one language is utilized simultaneously or when several languages are combined and switched throughout the same communication process (Nil & Shamala, 2012). In Malaysia, especially online, the blending of Bahasa Malaysia and English, known as "Manglish" and "Bahasa rojak" (Ibrahim, 2013) is unavoidable. Furthermore, spelling is not given as much attention online, and practically all individuals commonly use abbreviations to speed up communication. Due to the style of speech and mix of languages, chatbots may not respond to users correctly.

Chat robots, or chatbots, were developed in response to strong demand in the commercial sector and Artificial Intelligence (AI) capabilities. A chatbot communicates with humans via instant messaging, artificially emulating human interaction patterns. It is a computer software or virtual agent that uses Natural Language Processing (NLP) to allow humans and machines to speak with one another freely. NLP is a branch of AI that deals with voice recognition and machine translation. It primarily concerns text parsing, signal processing, semantics, and pragmatics, allowing for natural language generation to produce meaningful dialogues depending on the analysis context (Linzen & Baroni, 2021; Hussain et al., 2019). NLP is the ability of a computer program to understand human speech as it is naturally spoken. Without NLP, humans would have to talk in a precise, unambiguous, highly structured programming language for computers to understand them.

A chatbot is programmed to function without the assistance of human operators. It can respond to its queries as if it were a real person. When conversing with a chatbot, people may make typographical errors in their requests. Typographical errors include misspelt words or the usage of abbreviated words. An abbreviation is a condensed version of a word or phrase used in communication. The primary drawback of most existing chatbots is that they can only handle correctly phrased queries. NLP alone is insufficient to detect typographical errors in input queries. Even though typographical error checkers have become one of the most widely used features in many programs and applications, including web applications, crawlers, and web browsers, chatbots still lack them, mainly when "Manglish" is used as a query. As a result, this research aims to assist chatbots in responding to questions accurately, even when they contain "Manglish" sentences and typographical errors, using an embedding model of the N-gram overlapping approach along with a rule-based algorithm.

Section 2 of the research begins with a background study and approaches followed by methodology in section 3. The results and discussion are in section 4, and the conclusion is in section 5.

2. Background Study

2.1. Typographic errors

Table 1: Types of typographical errors.

Types of Errors	Examples
Single-word error - one letter error: single letter insertion single letter deletion single letter substitution	e.g. "homee" → "home". e.g. "hme" →"home" e.g "gome" →"home"
Multi-word errors - more than one word is wrong and missing,	e.g. "allow propr sped" → "allow proper speed".

Combined word or Word concatenation error
- a series of interconnected word errors or two adjacent characters combined.
	e.g. "helpfulstaff" → "helpful staff";
	e.g. "cannotdetet" → "cannot detect";
	e.g. “thankstelekom” → “thanks telekom”.
	e.g. “wirelesrouter” → “wireless router”.

The task of expanding abbreviations is to recognize shorter word forms and expand them to their correlated words.	e.g. "abbr" or "abbrev." e.g: acronyms: "NATO" e.g. initialisms: "HTML" or "FBI"

Typographic errors can be divided into 'non-word errors' and' real-word errors’ (Nonghuloo & Krishnamurthi, 2017). When a term is not in the dictionary, it is referred to as a non-word error. People regularly make non-word mistakes when typing. Most of the time, the errors are caused by accidental typing and a lack of spelling understanding. The real-word errors deal with the words that exist in the dictionary but are wrongly used in the context. This study will focus on non-word errors, common among chatbot users.

Error detection and correction are the two primary steps in typographic error checking (Patrick et al., 2010; Wong et al., 2006). Error identification is the process of identifying the error in the query. Error correction is rectifying the error by generating a list of suggested words or phrases. For instance, detecting "cta" as a misspelling and suggesting that it be replaced with "cat," "act," or "tac." As a result, recommendations can only be given after a misspelt word has been discovered (Chan et al., 2005).

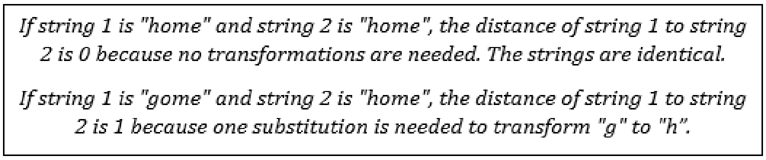
Table 2: Example of Manglish sentence errors

Manglish Sentence	Correct Meaning of the sentence
“bgmana sya nak add course”	Bagaimana saya hendak add course translated to English, "How can I add course.
"bgmana""nak"	Both are abbreviated words
"sya"	is a spelling error
“course”	English word

Table 1 shows several examples of single and multi-word typographical error problems. Table 2 displays some of the typographical error problems found in "Manglish" observed in this investigation.

2.2. Summary of error identification and correction approaches

Edit-distance, also known as Levenshtein Edit-distance, is a method of calculating the distance between two strings, token in research. "Distance" refers to the number of operations required to alter or transform one word or phrase into another. Damerau (1964) was the first to implement correcting spelling errors method based on edit distance. This method determines how many modifications (Levenshtein, 1966). There are three types of editing operations in this method. Modification or distance is the number of deletions, insertions, or modifications required to transform string 1 to string 2 (Bhaire et al., 2015), as shown in Fig. 1. This algorithm calculates the distance by comparing the source and targeted words. The definitive word to replace the error token matches the lowest distance value. The spelling correction solution developed by Petty et al. (2022) is an example of a spelling correction algorithm that implements the edit-distance method.



If string 1 is "home" and string 2 is "home", the distance of string 1 to string 2 is 0 because no transformations are needed. The strings are identical.

If string 1 is "gome" and string 2 is "home", the distance of string 1 to string 2 is 1 because one substitution is needed to transform "g" to "h".

Fig. 1: Converting string.

The second method, N-gram, is a probabilistic linguistic model for predicting the next item in a sequence. It is a set of consecutive characters taken from a string with a length of whatever n is set to be. N-gram tables can take on various forms: Unigram, when n is one, Bi-gram when n is two, and Tri-gram is used when n is three. N-grams can be used to determine the distance between two words. It can be used without a dictionary or together with one to determine the length. The N-gram model has two benefits: simplicity and scalability (Zhang, 2015). It does not require any prior understanding of the language (Atawy & ElGhany, 2018). The spelling correction solution developed by Sakuntharaj & Mahesan (2021) is an example of a spelling correction algorithm that implements the N-gram method.

The third method is the similarity keys technique. In this method, every string is to map into a key so that words with similar spellings have similar keys. It's called the SOUNDEX system (Nonghuloo & Krishnamurthi, 2017). Each dictionary word is assigned a key, and only dictionary keys are compared to the non-word key. The keys were computed for the non-word. The word with the most similar keys is chosen to replace the words. This method is fast because it only processes words that have comparable keys. With a suitable transformation algorithm, this method can manage typographic errors. The spelling correction solution developed by Aziz et al. (2021) is an example of a spelling correction algorithm that implements the similarity keys method.

The fourth method is the rule-based approach which uses a set of rules to capture common typographic issues. This algorithm offers a correction for misspelt words (Li et al., 2020). A corpus and rules-based algorithms (KalaiarasiSMA et al., 2017) were constructed using social media data based on the guidelines set by Dewan Bahasa and Pustaka guidelines (Dewan Bahasa dan Pustaka, 2008). This rule-based algorithm will identify unknown words and, combined with abbreviation rules by Samsudin et al. (2013) and Park & Byrd (2001) work, can detect word errors. The spelling correction solution developed by Downs et al. (2020) is an example of a spelling correction algorithm that implements the rule-based method.

Typographical errors by the user will cause the Chatbot to respond incorrectly. Some misspelt words or phrases will cause the Chatbot to misinterpret the intent of the input question and react to the user with an incorrect answer. In light of this issue, an automated typographic error using an embedding model of the N-gram overlapping with a rule-based algorithm is proposed in the next section.

3. Methodology

The proposed typographic of error identification and correction framework has three main parts, as shown in Fig. 2: Pre-processing and Token checking, which includes error identification and correction. The last part is feedback to the user.

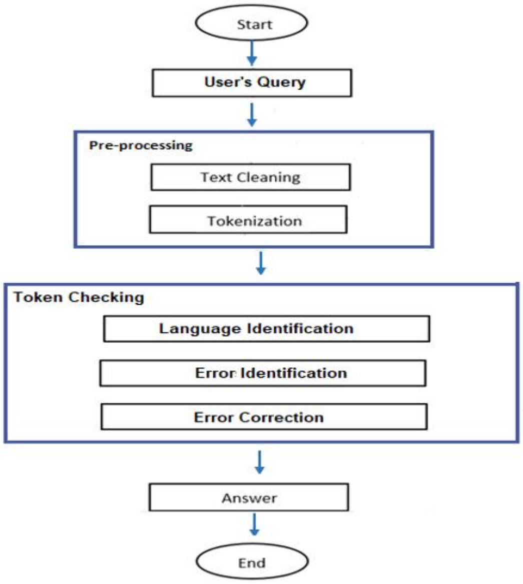


Fig. 2: Typographic framework.

Pre-Processing is the first phase which is done before error checking & correction of the user's queries. This phase is a crucial step in a chatbot. It includes cleaning and formatting the data before matching the chatbot answer. Text cleaning consists of many stages depending on the input of the query. The first step is removing all the emojis and emoticons, punctuation, special characters, and non-ASCII characters (Basri et al., 2013). The query is split into short sentences if the question is very long. The splitting is based on the dot (.), "and," "or," etc. Sentences will be tokenized after text cleaning. Tokenisation is a technique that breaks sentences into characters, words, phrases, or other meaningful elements and removes all the stop words before further processing. The text that has been pre-processed is then sent to the token-checking stage.

The token checking phase is divided into three parts: language identification, errors detection, and error correction. The token obtained from the pre-processing phase will be determined whether it is "English" or "Malay". Identifying the language of the tokens will ease the checking of the relevant corpus. After determining the language, the following step is to check for any errors in the tokens.

Error identification is a process that will check each word or phrase to detect any errors in the token. The method used to identify the error uses English and Malay corpus lookup. Corpus is a lexicon resource built from data, and in this research, the corpus was developed from data obtained from the online and education sector (KalaiarasiSMA et al., 2017). Each token will be compared to the data contained in the corpus. If the corpus is big, it will take longer to process. A hash map is used in this research to overcome this problem. This method overcomes searching and comparing the overall words in the corpus. For this, we compute the hash address and extract the word from the corpus. If the input word is not similar to the word in the address, it is treated as an error word.

Once the error word is detected, the next step is to correct this error. Error correction is finding the correct candidate term and providing a solution to correct the error in the token. The N-gram overlapping approach is used to find the right candidate term (Bosanac & Stefanec, 2011), as shown in Fig. 3. If the token is Malay words, then the rules base algorithm, together with the embedding method, will be used.

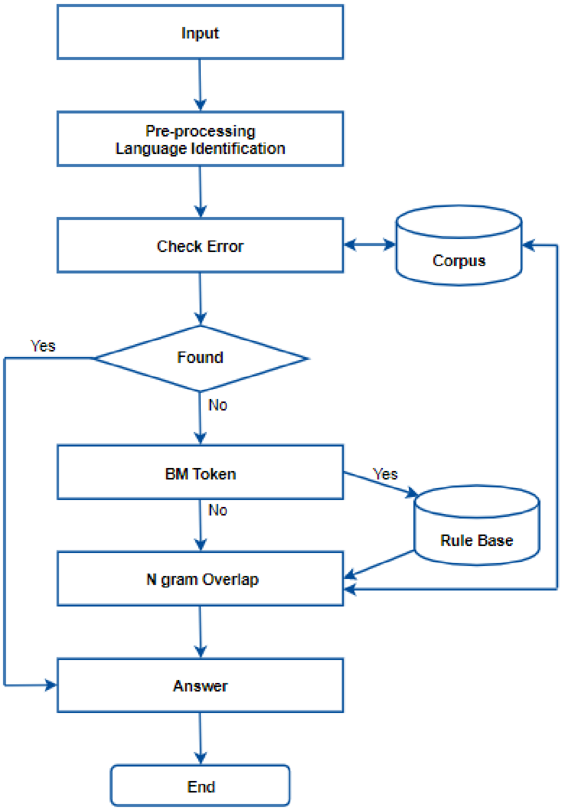


Fig. 3: Typographic error identification and correlation flowchart.

3.1. Proposed methodology

First, all possible candidate terms will be generated using N-gram based on the error words or phrases. Although N-gram distance produces a good list of candidate terms, too many candidates are considered. Furthermore, not all of them will be valid. Therefore, the candidate term generated will be checked and filtered based on the corpus. Even with the filtering, there still exist many candidate terms. The N-gram overlapping approach is used in this study to simplify the list of potential candidate words.

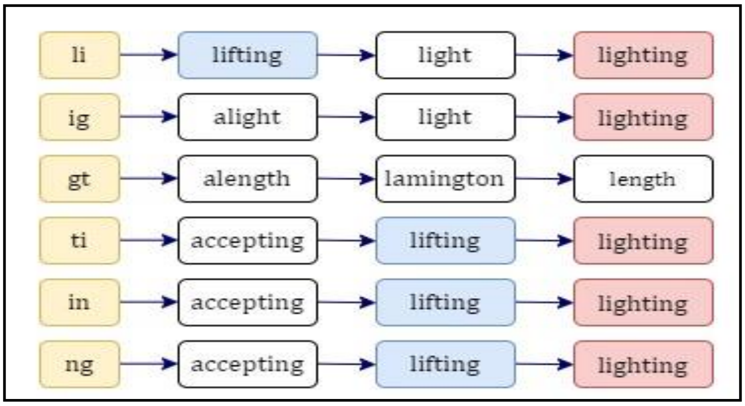
The proposed embedded approach doesn’t compare the entire corpus; it selects a "good set" of candidate terms and computes the distance only for this set of candidate terms. The N-gram overlap technique is utilized to choose the "good set" of candidates. This method will considerably minimize the number of candidate terms. For this, an N-gram index (Mahapatra & Biswas, 2011) was developed, and all the candidate terms matching (overlapping) with error words will be retrieved. The candidate with the highest probability value will replace the word error based on the Jaccard coefficient.

Table 3: Bi-gram Index for " Ligthing" error word

“Ligting”–	"li”	“ig”	“gt”	“ti”	“in”	“ng”

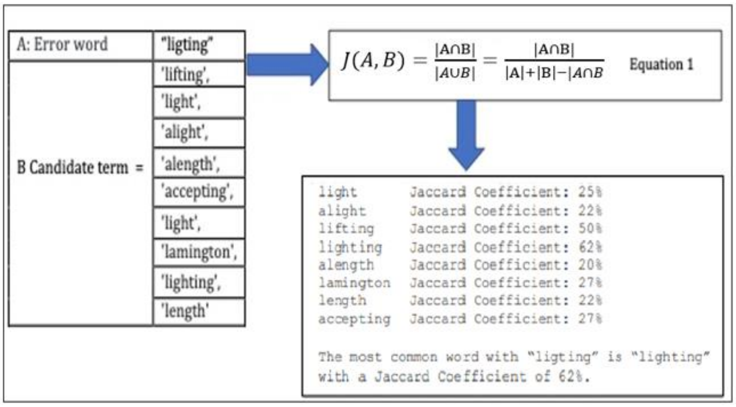
This part illustrates the proposed method. Let's assume N equals 2 and consider the error word is "Ligting". In this example, the N-gram index (Bi-gram) will contain all possible Bi-gram generated from the error token "Ligting", as shown in Table 3.

The process will further scan the corpus and extract all the potential candidate terms that match the Bi-gram generated on the error words. The list of all possible candidate terms matching (overlapping) for each Bi-gram will be extracted from the corpus. Fig. 4 shows the Bi-gram index maps of all the corresponding candidate terms list.



li → lifting → light → lighting
ig → alight → light → lighting
gt → alength → lamington → length
ti → accepting → lifting → lighting
in → accepting → lifting → lighting
ng → accepting → lifting → lighting

Fig. 4: The terms matching (overlapping) output.



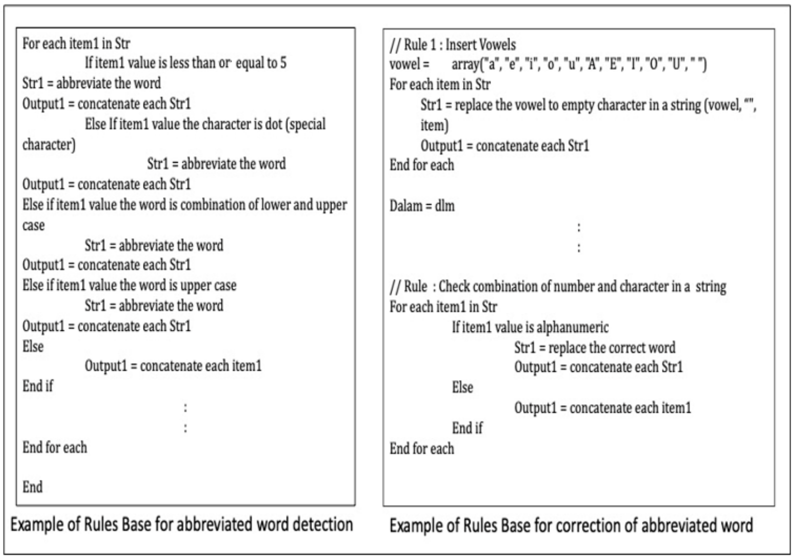
A: Error word	"ligting"
B Candidate term =	'lifting', 'light', 'alight', 'alength', 'accepting', 'light', 'lamington', 'lighting', 'length'

$$J(A,B) = \frac{|A\cap B|}{|A\cup B|} = \frac{|A\cap B|}{|A|+|B|-|A\cap B} \quad \text{Equation 1}$$

```
light      Jaccard Coefficient: 25%
alight     Jaccard Coefficient: 22%
lifting    Jaccard Coefficient: 50%
lighting   Jaccard Coefficient: 62%
alength    Jaccard Coefficient: 20%
lamington  Jaccard Coefficient: 27%
length     Jaccard Coefficient: 22%
accepting  Jaccard Coefficient: 27%

The most common word with “ligting” is “lighting”
with a Jaccard Coefficient of 62%.
```

Fig. 5: Jaccard coefficient results.



```
For each item1 in Str
          If item1 value is less than or equal to 5
Str1 = abbreviate the word
Output1 = concatenate each Str1
          Else If item1 value the character is dot (special
character)
                    Str1 = abbreviate the word
Output1 = concatenate each Str1
Else if item1 value the word is combination of lower and upper
case
          Str1 = abbreviate the word
Output1 = concatenate each Str1
Else if item1 value the word is upper case
          Str1 = abbreviate the word
Output1 = concatenate each Str1
Else
          Output1 = concatenate each item1
End if
                              :
                              :
End for each

End
```

Example of Rules Base for abbreviated word detection

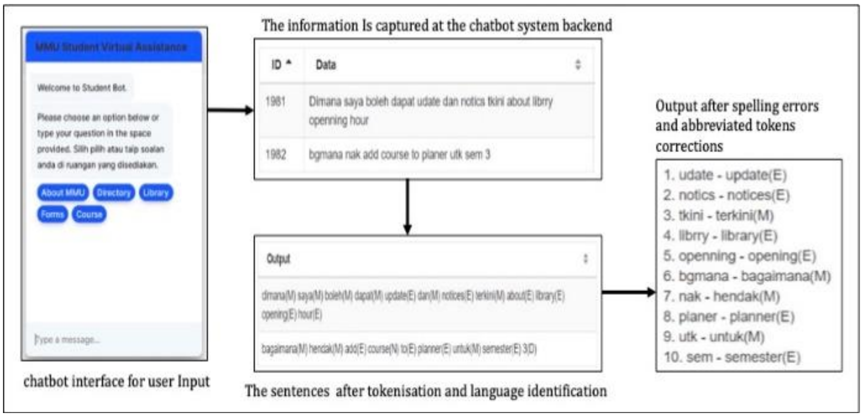
```
// Rule 1 : Insert Vowels
vowel =     array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U", " ")
For each item in Str
     Str1 = replace the vowel to empty character in a string (vowel, "",
     item)
     Output1 = concatenate each Str1
End for each

Dalam = dlm
                              :
                              :

// Rule  : Check combination of number and character in a  string
For each item1 in Str
          If item1 value is alphanumeric
                    Str1 = replace the correct word
                    Output1 = concatenate each Str1
          Else
                    Output1 = concatenate each item1
          End if
End for each
```

Example of Rules Base for correction of abbreviated word

Fig. 6: Example of Rules Base for abbreviated word detection and correction.


MMU Student Virtual Assistance
Welcome to Student Bot.
Please choose an option below or type your question in the space provided. Sila pilih atau taip soalan anda di ruangan yang disediakan.
About MMU
Directory
Library
Forms
Course
Type a message...
chatbot interface for user Input
The information Is captured at the chatbot system backend
The sentences after tokenisation and language identification
Output after spelling errors and abbreviated tokens corrections

ID	Data
1981	Dimana saya boleh dapat udate dan notics tkini about librry openning hour
1982	bgmana nak add course to planer utk sem 3

Output
dmana(M) saya(M) boleh(M) dapat(M) update(E) dan(M) notices(E) terkini(M) about(E) library(E) opening(E) hour(E)
bagaimana(M) hendak(M) add(E) course(N) to(E) planner(E) untuk(M) semester(E) 3(D)

1. udate - update(E)
2. notics - notices(E)
3. tkini - terkini(M)
4. librry - library(E)
5. openning - opening(E)
6. bgmana - bagaimana(M)
7. nak - hendak(M)
8. planer - planner(E)
9. utk - untuk(M)
10. sem - semester(E)

Fig. 7: Chatbot input data typographical error detection and correction process.

The final selection of the correct word is made by scanning the number of occurrences where the candidate terms appear in the lists for all Bi-grams of "ligting". For example, "lifting" appears four times in Fig. 4, whereas "lighting" appears five times. The correct selection of the candidate term is assured by applying the Jaccard coefficient (Fig. 5: Equation 1) to measure the N-gram overlap between two candidate terms. Let's assume A stands for the error word, and B is the potential candidates' term to replace the error words. Based on Fig. 5, the right candidate word to replace “ligting." is, 'lighting.'

If the token is identified as a Malay token, it will be subjected to rule-based algorithm analysis (Fig. 6) and the embedding approach. The rule is developed based on DBP guidelines and the work of Kalaiarasi et al. (2017), in combination with the abbreviation rules algorithm by Park & Byrd (2001) and Samsudin et al. (2013).

The chatbot interface is shown in Fig. 7. Once the user keys the questions to Chatbot, the information is captured at the backend. The data will be processed. The sentences will be tokenized, and the token language will be tagged. The spelling errors and abbreviated tokens will be changed, expanded, and passed chatbot engine to process and answer the question. The Chatbot is trained on English and Malay tokens.

3.2. Dataset

A corpus of 150,000 words was used in this research; out of this, 50000 were Malay. The system captured a total of 1982 questions in the Chatbot, with 80 sentences representing the "Manglish" language extracted for this study. These language structures include tokens, name entities, and abbreviations in Malay and English. The typographic errors using an embedding algorithm were applied to these queries. The accuracy of identifying and correcting the errors was measured.

4. Result and Discussion

4.1. Analysis

From the 80 Manglish sentences, a total of 602 tokens were tokenized. After removing the stop words (282 tokens), there were 320 tokens. Fig. 8 shows that 50.3% were correct tokens, whereas 49.7% were error tokens, with 37.2% spelling errors and 12.5% abbreviated words. The algorithm correctly identifies 86.2% of the error words and corrects 84.3% of them, as illustrated in Fig. 9. Only 1.9% of words were correctly detected; however, they were incorrectly corrected. The algorithm could not detect 13.8% of error tokens, with abbreviation errors accounting for 9.4%.

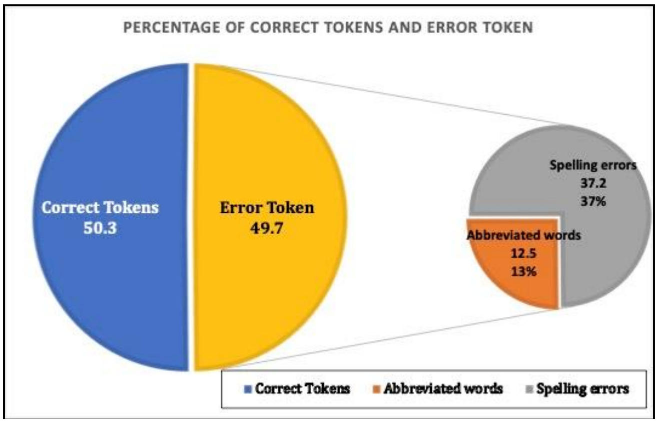
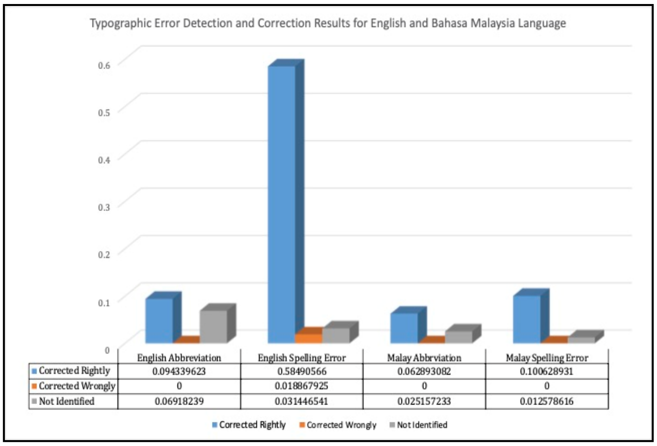


Fig. 8: Percentage of correct tokens and error token detection.



	English Abbreviation	English Spelling Error	Malay Abbrviation	Malay Spelling Error
Corrected Rightly	0.094339623	0.58490566	0.062893082	0.100628931
Corrected Wrongly	0	0.018867925	0	0
Not Identified	0.06918239	0.031446541	0.025157233	0.012578616

Fig. 9: Typographic error results for english and malay language.

4.2. Discussion

This study aimed to detect and rectify typographical errors in "Manglish" data input for chatbots. The N-gram overlapping using the Jaccard coefficient analysis method has been used for detecting typographical errors. The current findings show that identifying the correct term for non-word errors in "Manglish" is adequate. Based on our corpus and a rule-based guideline, the chatbot system could locate 86.2% and correct 84.3% of identified " Manglish" words.

One of the research's significant limitations is that the Malay corpus used in this study only contains around 50000 words. In addition, new words or abbreviated words are always coined. For this, new words gathered are added to a temporary table for future processing. These terms will be manually updated in the corpus once it reaches a specified threshold. This application sets the threshold to 10 occurrences before a new word can be included.

5. Conclusions

Based on this research, Chatbot's primary yet crucial element in handling typographical error queries is discussed. Most of the existing chatbots are only able to handle proper input queries. Suppose there are typographical words or phrase errors in the query, especially in Manglish language; Chatbot may misinterpret the question's intent and provide an inappropriate reply to the user. Therefore, in this paper, we presented N-gram overlapping embedding with rule base methodology to handle typographical errors in Chatbot. This embedding method can detect 86.2% of “Manglish" words and correct 84.3% of them, resulting in a better Chatbot response to the user.

Acknowledgments

This work was supported by TMRD grants (grant number MMUE/210038).

References

Aziz, R., Anwar, M. W., Jamal, M. H., & Bajwa, U. I. (2021). A hybrid model for spelling error detection and correction for Urdu language. *Neural Computing and Applications,* 33, 14707-14721. DOI: https://doi.org/10.1007/s00521-021-06110-7.

Basri, S. B., Alfred, R., On, C. K., & Razali, M. N. (2013). An automatic spell checker framework for malay language blogs. *In Social Media Retrieval and Mining,* Springer, Berlin, Heidelberg, 55-64.

Bhaire, V. V., Jadhav, A. A., Pashte, P. A., & Magdum, P. G. (2015). Spell checker. *International Journal of Scientific and Research Publications*, 5(4).

Bosanac, S. & Stefanec, V. (2011). N-gram overlap in automatic detection of document derivation. *INFuture2011: Information Sciences and e-Society*. 2011 in Belgrade, Serbia.

Chan, S., He, B., & Ounis, I. (2005). An in-depth survey on the automatic detection and correction of spelling mistakes. In *Proceedings of the 5th Australasian Data Mining Conference*.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM,* 7(4), 171-176.

Dewan Bahasa dan Pustaka (2008). Singkatan khidmat pesanan ringkas bahasa melayu, “khidmat pesanan kandungan, *Dewan Bahasa dan Pustaka,* 65.

Downs, B., Anuyah, O., Shukla, A., Fails, J. A., Pera, M. S., Wright, K., & Kennington, C. (2020). KidSpell: A child-oriented, rule-based, phonetic spellchecker. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020),* 12, 6937-6946.

El Atawy, S. M., & Abd ElGhany, A. (2018). automatic spelling correction based on n-gram model, 182, 5-9.

Hussain, S., Sianaki, O., & Ababneh, N. (2019). A survey on chatbots/chatbots classification and design techniques.

Ibrahim, N. (2013). Code-switching in advertising: an exploratory study on " Manglish" and " Bahasa Rojak". In newspapers and advertisements. Diss. Universiti Teknologi MARA.

KalaiarasiSMA, Surendren, JayaKumar, K (2017). Generation of Malay Lexicon. *American Journal of Applied Science,* 503-510.

KalaiarasiSMA, JayaKumar, K, Shohel, M.S and Praviny, M. (2017). Comparison of stochastic and rule-based POS tagging on malay online text. *American Journal of Applied Science*.

Levenshtein, V. (1966). 'Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), 707–710.

Linzen, T. & Baroni, M. (2021). Syntactic structure from deep learning. *Annual Review of Linguistics*.

Li, X., Liu, H., & Huang, L. (2020). Context-aware stand-alone neural spelling correction. *arXiv preprint arXiv*:2011.06642.

Mahapatra, A.K, Biswas, S. (2011). Inverted indexes: Types and techniques. *IJCSI International Journal of Computer Science Issues*, 8(4), No 1, July.

Nil, Z. M. & Shamala, P. (2012). Code-switching in Gol & Gincu. *Procedia - Social and Behavioral Sciences,* 66, 169–75.

Nonghuloo, M. S. & Krishnamurthi, K. (2017). Spell checker for Khasi language. *Int. J. Softw. Eng,* 7(1).

Park, Y. & Byrd, R. J. (2001). Hybrid text mining for finding abbreviations and their definitions. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Patrick, J, Sabbagh, M., Jain, S., & Zheng, H. (2010). Spelling correction in clinical notes with emphasis on first suggestion accuracy. *2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining*, 1-8.

Petty, T., Hannig, J., Huszar, T. I., & Iyer, H. (2022). A new string edit distance and applications. arXiv. DOI:https://doi.org/10.48550/arXiv.2203.06138.

Sakuntharaj, R. & Mahesan, S. (2021). Missing word detection and correction based on context of tamil sentences using n-grams. *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, 10, 42-47. DOI: https://doi.org/10.1109/ICIAfS52090.2021.9606025.

Samsudin, N, Puteh, M, Hamdan, A. R. and Nazri, M. Z. A. (2013). Normalization of noisy texts in Malaysian online reviews. *J. Inf. Commun. Technol.*, 12(1),147–159.

Wong, W., Liu, W., & Bennamoun, M. (2006). Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In *Proceedings of the fifth conference on Data mining and analytics*, 61, 83-89.

Zhang, W. (2015). Comparing the effect of smoothing and n-gram order: Finding the best way to combine the smoothing and order of n-gram. [PhD dissertation]. Florida Institute of Technology, Melbourne, Florida, 1–6.