Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 22, No. 1, March 2022

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

Auto-Correction Model for Lip Reading System

Nada Hussain Ali[1], Matheel E. Abdulmunim[2], Akbas Ezaldeen Ali[3]

[1]*Department of Computer Technology Engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq*

[2,3]*Computer Science Department, University of Technology, Baghdad, Iraq*

[1]*nada.hussien@sadiq.edu.iq*, [2]*110104@uotechnology.edu.iq*, [3]*110034@uotechnology.edu.iq*

Abstract *-Auto-Correction is the process of correcting a misspelled word typed by the user as an application of automated translation process. lip-reading is the process of recognizing the words through processing and observing the visual lip movement of a speaker's talking without any audio input. Although visual information itself cannot be considered as enough resource to provide normal speech as intelligibility, it may succeed with several cases especially when the words to be recognized are limited. Auto-correction is a trail to diminish the number of errors that can be generated by lip reading systems and to improve their accuracy, many error-correction techniques were visualized. In this paper an auto-correction model is proposed to correct the misspelled words recognized by a lip reading system, the output of a lip reading system is subjected to auto-correction model to enhance the accuracy of the system. The auto-correction model is based on levenshtien distance and dictionary lookup with a proposed dataset. The proposed model achieved accuracy of more than 67% enhancing the lip reading system by almost 30%.*

Index Terms*— auto-correction, lip reading, levenshtien distance, dictionary lookup.*

I. INTRODUCTION

There are several ways to write the sentence with the same meaning. Sometimes the written text of some users might be unexpected by the readers, Creating apprehensive and clear text it's not that easy especially for people with a different mother language. An unusually written word in a sentence makes a spelling error.[1]

Auto-correction is a trail to diminish the number of errors that can be generated by lip reading systems and to improve their accuracy, many error-correction techniques were visualized, several of them are manual works by post-editing the recognized output reflex to correct misspellings, while other techniques works on different ways[2]. Spelling correction is a well-versed task in Natural Language Processing (NLP).Automatic spelling correction is important field for many applications in NLP like text summarization, web search engines, sentiment analysis and speech recognition[3].

Spellchecking is the task of knowing which word is misspelled and correcting it [4]. There are three sub-problems in spell checking task: non-word error detection, isolated-word error correction, and context-dependent error correction. Non-word error detection and correction techniques have headed to fall into two techniques n-gram and dictionary look up. The most used technique in optical character recognition is the n-gram analysis which work by finding the unusual sequences of characters and considering it as an indicator for error. More common used technique in spelling correction systems is dictionary lookup: this technique consider any word that does not appear in the dictionary as a misspelled word. Isolated-word spelling correction systems mostly use a form of minimum edit distance to rank or generate suggestions. Damerau discovered that over 80% of spelling errors involve of one of the following operations: a deleted

Received 30/June/2021; Accepted 5/September/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

letter, a letter substituted for another, an inserted letter, or two letters transposed or switched. Context-dependent error correction is used in cases where a word that is correctly spelled replaced with another word. Statistical language models are the techniques that are used to detect ill-formed sequences of words. Word spelling correction systems mostly generate a rank or suggestion list for the misspelled word by using some kind of minimum edit distance. The Damerau–Levenshtein edit distance is one of the most used technique for such calculations [5].

Spellchecking can be implemented on speech recognition as well as lip reading, in speech recognition there are three types of errors that occurs during the recognition process. The first type when a word sequence is translated as a different word the type is called substitution the second type when a word is missing from the reference, means that the word in the reference is completely missing from the translation this type is called deletion. The last type is insertion that occur when a word a pear in the translation that has no reference [6]

Lip reading system in the last few decades gained a lot of interest, it is an important fielded to help people with hiring problems or to make it a significant integrated process for surveillance system. In both cases lip reading system might miss read a word, to overcome this problem an auto-correction model is proposed to check the uttered word and to correct it in order to get the best results from such systems. The remaining sections of this paper are as follow: section 2 related work description is provided, section 3 the text similarity, dictionary lookup and similarity measure techniques are presented, section 4 the proposed model frame work is provided and the experimental results are discussed in section 5 conclusion and future work are discussed in section [6].

II. RELATED WORK

In this section, a brief description is presented for the most relevant works that are relates to the proposed model in this paper as follow:

In [7] the authors presented a system that can detect and semantic errors in Arabic text by using four methods that are contextual based on linguistic information and statistics information, the system implemented on Arabic corpus by employing multi agent system. The system was able to detect the semantic validity of the words in sentences, the system achieved results of 90% precision and 83% recall, the system still have some limitation like the size of the corpus.

In [8] the authors proposed a system that correct real word errors based on contextual information implemented on the confused words that belongs the brown corpus that have a set of confused words that are fed into the system, the authors presented two phase to correct the words one uses the trigram algorithm and the other uses Bayesian approach and achieving accuracy of 89%, the limitation of this work lies in the corpus size bigger corpus means more confused set of words that leads to better detection and correction.

In [9] the authors presented a proposal for Russian language using morphological and semantic information, the proposed system used SpellRuEval contest (Sorokin et al., 2016) dataset to correct errors on sentence level by employing a hypothesis of a noisy channel model and feature reranking and achieving accuracy of 78%.

In [10] the authors presented a system that check the spilling and detect errors of Bangali words and suggest a suitable word to replace the misspelled word using Levenshtein distance and unigram

Received 30/June/2021; Accepted 5/September/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

strategy, the system were implemented on added Bangali corpus, the system achieved accuracy of 78%, the author concluded that the accuracy achieved is due to the huge amount of data used in the corpus, which also was the reason for the time taken by the system to suggest a suitable word.

In [11] the authors tried to generate errors form a small seed of errors (misspelled words) based on annotated corpus in order to have enough data to use deep neural networks for error detection and correction in typed words, the presented work achieved accuracy of around 90%.

III. METHODOLOGY

The rapid data growth in recent years caused some information problem, text similarity approach was one of the solutions in many areas. Document, paragraph and sentence similarity are based on finding similarities between words which leads to text mining by finding the relevant information between words. Text mining approach is very important in several systems like text classification, document clustering, text summarization, machine translation, auto correction. [12] Words usually similar either lexically or semantically. If the Words have the same character sequence they are considered as lexically similar, while if the words have the same meaning they are considered as semantically similar but if the words doesn't have the same meaning but used in the same context then the words are not related. Lexical similarity can be measured using string based algorithms while semantic similarity can be measured using knowledge based and corpus based algorithms[13]. Many similarity measures have been proposed and applied in a wide range in literature such as Jaccard correlation coefficient, cosine similarity and levenshtein distance. Similarity is often captured in terms of similarity and dissimilarity by distance measure[14] and Euclidean distance is considered one of the best metrics in finding the best neighbors to the misspelled word[15].

Dictionary lookup method are very popular in spellcheckers systems, but a robust dictionary lookup needs complex calculations especially with large size dictionaries. Levenshtien distance measure is an effective technique with dictionary lookup that is able to reduce the complexity of the lookup process[16].

Levenshtein distance introduced by Vladimir Levenshtein is a metric to check similarity between two strings, the main interest of levenshtien is to extend Hamming's error correction to include the insertion and deletion of single letter. Levenshtien metric is the minimum number needed of edits like substitutions, insertion and deletion of single letter to change X string into Y[17]. Levenshtien distance algorithm is considered as editor algorithm that use dynamic programming string for operation. The algorithm works by giving a Wight of 1 for every edit operation (substitution, insertion, deletion) for example the levenshtien distance between dog and cat is 3 one for substituting d for c, one for substituting o for a and one for substituting g for t. The algorithm can also be used for defining the number of adjustment and modification like insertion and deletion in a string s1 to be the same as s2, it counts the number of operation on the strings[18]. Assuming two strings S and T with length m and n respectively, construct matrix LD [n + 1, m + 1],

In order for the algorithm to calculating the value of each cell LD (i,j) in the matrix LD, the formula is as follows[19]:

Received 30/June/2021; Accepted 5/September/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

$$\text{LD}\ (i,j) = \begin{cases} 0, i = 0, j = 0 \\ j, i = 0, j > 0 \\ i, i > 0, j = 0 \\ min. i > 0, j > 0 \end{cases} \quad \ldots(1)$$

Min=min{LD(i-1,j)+1,LD(i,j-1)+1,LD(i-1,j-1)+f(i,j)},where f(i,j) =1 if the ith word of S is not equal to the jth word of T, otherwise f (i, j) = 0. Finally, the LD (n, m) of the rightmost corner of the matrix is the size of the desired edit distance. The similarity of the two strings can be represented by the LD matrix, intuitively, the greater the LD, the smaller the similarity. Assuming that there are two strings S and T of length m and n respectively, using LD to express their edit distance, using Sim (S, T) to show their similarity [19]:

$$\text{Sim(S,T)} = 1 - \frac{LD}{\max(m,n)} \quad \ldots.(2)$$

IV. RESEARCH METHOD

The proposed model is divided into several stages as illustrated in the flowchart of this model *Fig. 1* in the subsections below a full description of the model is presented.

4.1 Dataset

The dataset is proposed by the authors collected and processed, each video has the pronunciation of the targeted word, and these words is processed and recognized by a proposed system for lip reading called Constructed Model for Lip reading System (CMFLRS)[20]. The output of the CMFLRS is word that might be misapplied corrected by the propose auto-correction model presented in this paper. The data set contain almost 400 video for different words recorded for different speakers from different ages and genders. These videos were splitted into two sets, the first set contain almost 300 videos were tested and the uttered word was entered to the dictionary as a misspelled word and the correct word associated with it, the other part was used as a testing samples for the model.

4.2 proposed model

The proposed model is an auto-correction model for the output of the lip reading system, it is based on dictionary lookup and levenshtien distance measure, and the steps of the model are illustrated below:

Step one dictionary creation: the first step is creating dictionary that include the misapplied word and the right word associated with it, the right words are placed in the value of the dictionary, the dictionary is divided into three types (three letter word, four letter word, five letter word).

Step two searching for a match: in this step a search for a match is conducted, the word of interest (the output from the CMFLRS) is searched in the dictionary keywords if found than the word is right spilled if not than the word is misspelled.

Step three calculation levenshtien distance: in this step the word of interest if wasn't found in the second step than it will be subjected to levenshtien distance measure. The levenshtien distance will be calculated for the misspelled word with the whole dictionary the word with the smallest value will be chosen as the correct word, since the levenshtien distance works by finding how many changes are needed to make the two words similar then the smallest value means the most similar word to the word of interest.

Fig. 1 shows the model flowchart with a detailed explanation in the preceding section.

Received 30/June/2021; Accepted 5/September/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

The first step creating the dictionary: a three types of dictionary were created 3 letter word , 4-letter word and 5 letter word by choosing the number of letter when executing the model, the purpose of dividing the dictionaries into three types is to speeding the matching process. Each element in the dictionary has a value and associated keyword, the value is the misspelled word of the keyword, the cases of misspelled were gathered from different speakers uttering the words correctly but recognized with a one or two letter wrong like (fan recognized as lan, man recognized as mas) these misspelled words were entered as the value to the correct word {mas:man , lan:fan}. The dictionary can be expanded as more cases and misspelled words are entered leading to more global dictionary for the purpose of auto-correction for lip reading.

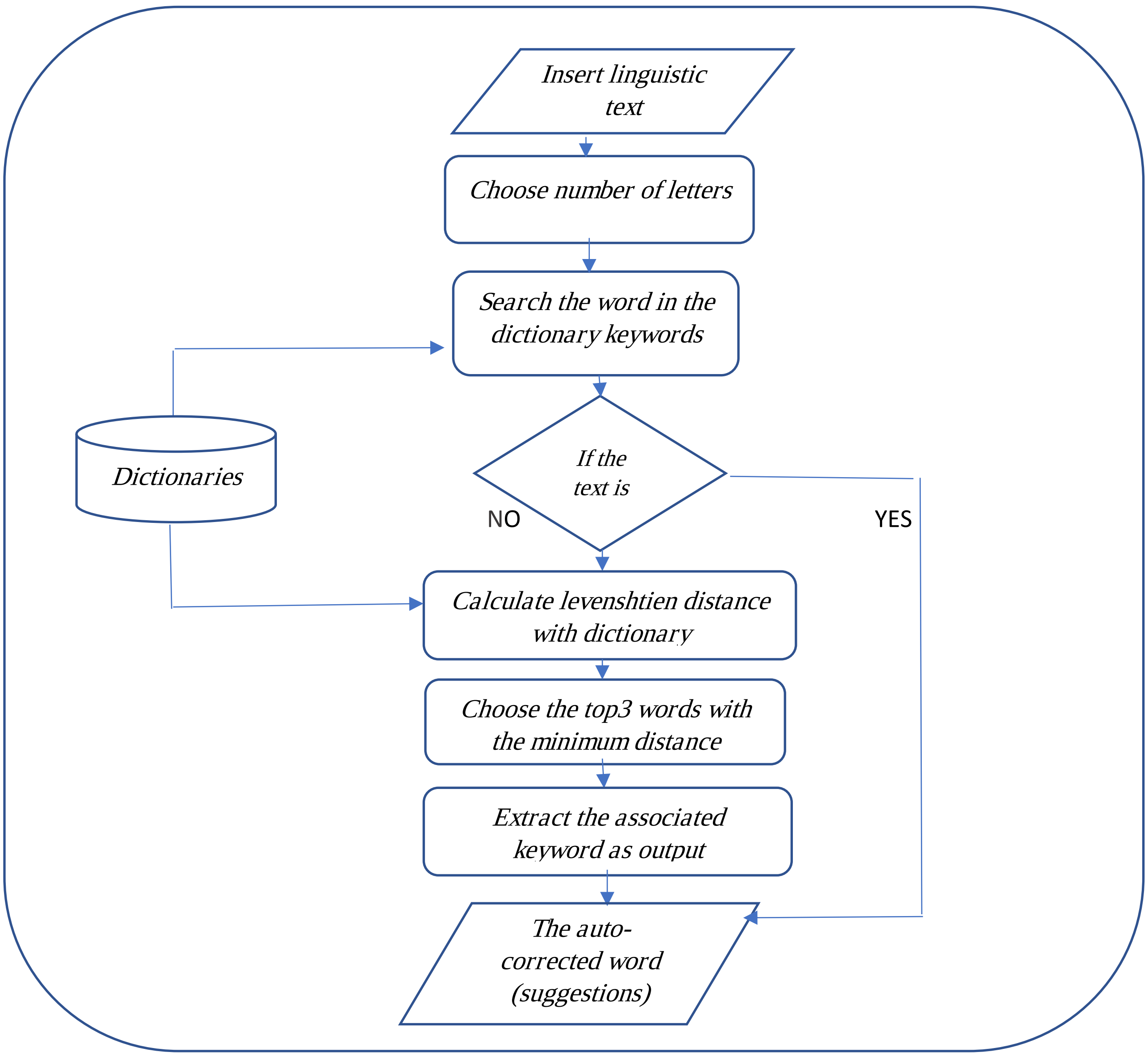

FIG.1. MODEL FLOWCHART

Received 30/June/2021; Accepted 5/September/2021
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

The second step searching for a match: in this step a search for a match is conducted, first a search for a match between the word of interest and the keywords of the dictionary if the word is found that it is not misspelled, if the word was not found it means it's a misspelled word. The search process will be faster when choosing the number of utterance letters in the word (silent letters not included) for instance , instead of searching in a dictionary that contain a 1000 words of different lengths searching in a dictionary that has 200 words only from a specific length.

The third step levenshtien distance calculation: when the search in step two fails to find a match it means the word is misspelled. The misspelled word is subjected to levenshtien distance calculation with all the values of the dictionary, the reason why the search is implemented with the values of the dictionary instead of the keywords is that during the testing of the model almost all speakers have the same misspelling (a missing letter or incorrectly recognized letter) for that reason the search with the value was more useful, faster and more accurate than calculating the levenshtien distance with the keywords. The levenshtien distance is calculated according to equation 1 and 2 the top 3 closest words with the minimum value are considered as suggestion to the word of interest, once the closest words are found the associated keyword is presented as the utterance words after it were subjected to auto-correction.

The following algorithm, represent the steps of the model.

Algorithm (1), Auto-correction for lip reading algorithm

Input : linguistic text

Output: auto-corrected word
Begin
Step1: create dictionaries based on number of letters in the input text
    Step1.1 create dictionary of 3 letters (dic3)
    Step1.2 create dictionary of 4 letters (dic4)
    Step1.3 create dictionary of 5 letters (dic5)
Step2: enter number of letters in the input text

Step3: search the input text in the dictionary(dic3 , dic4 or dic5 based on step2) key words

  If the text word is found in the dictionary key words then / the dictionary according to the value entered in step2*
    Return the same word(the word is not misspelled)
  Else
    Step4: calculate levenshtien distance between the input text and the dictionary Values(dic3 , dic4 or dic5 based on step2) using eq.1 and eq.2.

    Step5: choose the value from the dictionary(dic3 , dic4 or dic5 based on step2) with the minimum distance from the input text
    Step6: extract the associated keyword with the value from the determined dictionary

Step7: return the auto-corrected word

Received 30/June/2021; Accepted 5/September/2021

 ISSN (Print) 1811-9212   *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

V. RESULTS

The model achieved an excellent result in regard to the scope of the implementation. The model correct the words that read from a lip reading system based on reading each letter of the word individually. The lip reading model succeeded to recognize same letters correctly but didn't succeeded to read a whole word with high accuracy some examples (study cases) are illustrated in Table I.

TABLE I. STUDY CASES.

The correct words	Read by the lip reading system
fan	aan
mice	mii
wall	ooll
see	sii
nine	lan
left	lifi
meet	mel
lime	lnn
soup	som

As shown in Table I, the lip reading system misplaced more than one letter in some words and in other it only misplaced one letter, for that reason the necessity has arisen to integrate the system with an auto-correction model.

The auto correction model which is considered as an integration part of the lip reading system, which succeeded to boost the accuracy almost 30%. The accuracy of the proposed auto-correction model for lip reading system achieved more than 67% tested on more than 100 video of different words. The model were able to recognize the word Fat from the word Fate and Fan from the word Man, but the model were unable to recognize words like Door or Cold. The levenshiten distance algorithm produced the closest distance values between the word of interest and the dictionary, the top3 closest words were considered as output (suggestions to the user). Table II illustrate the results of the model.

TABLE II. MODEL RESULTS.

Total Number of Samples (videos) in dictionary	Number of testing Samples (videos)	Number of corrected words in testing samples	Number of words uncorrected words in testing samples
300	104	70	34
Accuracy		67.3%	

Received 30/June/2021; Accepted 5/September/2021

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

VI. CONCLUSION

A proposed model for Auto-correction for lip reading a system based on levenshtien distance algorithm, the model takes input as misspelled words from a lip reading system proposed by the authors based on deep learning and micro-content algorithm. The model was tested on a proposed dataset that contain more than 300 video of different words uttered by different speakers from different ages and genders. The model enhanced the accuracy of the lip reading system by almost 30%, a few points were concluded as follows:

1. The levenshtien distance algorithm was suitable for a letter to letter comparison between the word of interest and the words in the dictionary.
2. By splitting the dictionary in to several sub dictionaries according to the number of uttered letters the word has the execution time was less and the accuracy was better.
3. The model were able to give really good recognition rate to a very close words as fat and fate but failed to correct other words like door or cold based on the output from the lip reading system.
4. The overall accuracy also effected by the speakers in the videos each person might utter the word in a different way.

For future work to enhance the accuracy of the model a corpus must be built for thousands of misspelled words based on uttered words for different speakers and different accents.

REFERENCES

[1] D. Hládek, J. Staš, M. Pleva, Survey of Automatic Spelling Correction, Electronics, 2020, doi:10.3390/electronics9101670.

[2] Y. Bassil, M. Alwani, Post-Editing Error Correction Algorithm For Speech Recognition using Bing Spelling Suggestion, (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 3, no.2, 2012.

[3] P. Etoori, M. Chinnakotla, R. Mamidi, Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning, Proceedings of ACL, pp. 146–152 Melbourne, Australia, July 2018.

[4] S. M. El Atawy, A. Abd ElGhany, Automatic Spelling Correction based on n-Gram Model, International Journal of Computer Applications, vol.182 , no. 11, pp. 0975 – 8887, August 2018.

[5] K. H. Lai , M. Topaz , F. R. Goss , L. Zhou, Automated misspelling detection and correction in clinical free-text records, Journal of Biomedical Informatics 55 , pp. 188–195,2015.

[6] R. Errattahi, A. EL Hannani , H. Ouahmane, Automatic Speech Recognition Errors Detection and Correction: A Review , International Conference on Natural Language and Speech Processing(ICNLSP), Elsevier , pp. 32–37, 2018.

[7] Ch. O. Zribi ,M. Ahmed, Detection of semantic errors in Arabic texts, Artificial Intelligence 195, Elsevier, pp. 249–264, 2013.

[8] S. Sharma, S. Gupta, A correction model for real-word errors, 4th International Conference on Eco-friendly Computing and Communication Systems, Elsevier, 2015.

[9] A. Sorokin, Spelling Correction for Morphologically Rich Language: a Case Study of Russian, Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, Valencia, Spain, pp. 45–53, April 2017.

[10] M. Hossain, F. Labib, A. S. Rifat, A. K. Das, M. Mukta, Auto-correction of English to Bengali Transliteration System using Levenshtein Distance, 7th International Conference on Smart Computing & Communications (ICSCC), 2019.

[11] K. Shah, G. de Melo, Correcting the Autocorrect: Context-Aware Typographical Error Correction via Training Data Augmentation, Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), Marseille, pp. 6930–6936, May 2020.

[12] D. D. Prasetya, A. P. Wibawa, T. Hirashima, The performance of text similarity algorithms, International Journal of Advances in Intelligent Informatics vol. 4, no. 1, pp. 63-69, March 2018.

[13] M. K.Vijaymeena, K. Kavitha, A Survey On Similarity Measures In Text Mining, Machine Learning and Applications: An International Journal (MLAIJ) vol.3, no.1, March 2016.

Received 30/June/2021; Accepted 5/September/2021

DOI: https://doi.org/10.33103/uot.ijccce.22.1.7

[14] K. Maher, M. S. Joshi , Effectiveness of Different Similarity Measures for Text Classification and Clustering, International Journal of Computer Science and Information Technologies(IJCSIT), vol. 7 no. 4, pp. 1715-1720, 2016.

[15] C. A. Boiangiu, M. Zaharescu, O. Ferche, and A. Danescu, Automatic Correction of OCR Results Using Similarity Detection for Words and Fonts, International Journal Of Applied Mathematics And Informatics, vol. 10, 2016.

[16] R. Haldar, D. Mukhopadhyay, Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach, ArXiv,2011.

[17] B. Berger, M. S. Waterman, Y. W. Yu, Levenshtein Distance, Sequence Comparison and Biological Database Search, in IEEE Transactions on Information Theory, vol. 67, no. 6, Pages 3287-3294, June 2021, doi: 10.1109/TIT.2020.2996543.

[18] M. M. Yulianto, R. A. Alamsyah, Autocomplete and Spell Checking Levenshtein Distance Algorithm to Getting Text Suggest Error Data Searching in Library, Scientific Journal of Informatics vol. 5, no. 1, May 2018.

[19] Sh. Zhang ,Y.Hu , G. Bian, Research on String Similarity Algorithm based on Levenshtein Distance, IEEE, 2017.

[20] N. H. Ali , M. E. Abdulmunim, A. E. Ali, Constructed Model for Micro Content Recognition in lip reading Based Deep Learning, Bulletin of Electrical Engineering and Informatics, vol. 10, no. 5, pp. 2557-2565, October 2021.

Received 30/June/2021; Accepted 5/September/2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352