IAES International Journal of Robotics and Automation (IJRA)
Vol. 10, No. 4, December 2021, pp. 326~339
ISSN: 2722-2586, DOI: 10.11591/ijra.v10i4.pp326-339

Adaptive language processing unit for Malaysian sign language synthesizer

Haris Al Qodri Maarif[1], Teddy Surya Gunawan[2], Rini Akmeliawati[3]
[1]Department of Mechatronics Engineering, International Islamic University Malaysia, Malaysia
[2]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia
[2]School of Electrical Engineering and Telecommunications, University of New South Wales, Australia
[3]School of Mechanical Engineering, University of Adelaide, Australia

Article Info

Article history:

Received 24 Jun, 2021
Revised 29 Jul, 2021
Accepted 26 Aug, 2021

Keywords:

Classifier
Distance algorithm
Malaysian sign langauge
Natural language processing

ABSTRACT

Language processing unit (LPU) is a system built to process text-based data to comply with the rules of the sign language grammar. This system was developed as an important part of the sign language synthesizer system. Sign language (SL) uses different grammatical rules from the spoken/verbal language, which only involves the important words that hearing/impaired speech people can understand. Therefore, it needs word classification by LPU to determine grammatically processed sentences for the sign language synthesizer. However, the existing language processing unit in SL synthesizers suffers time lagging and complexity problems, resulting in high processing time. The two features, i.e., the computational time and success rate, become trade-offs which means the processing time becomes longer to achieve a higher success rate. This paper proposes an adaptive LPU that allows processing the words from spoken words to Malaysian SL grammatical rule that results in relatively fast processing time and a good success rate. It involves n-grams, natural language processing (NLP), and hidden Markov models (HMM)/Bayesian networks as the classifier to process the text-based input. As a result, the proposed LPU system has successfully provided an efficient (fast) processing time and a good success rate compared to LPU with other edit distances (mahalanobis, Levenshtein, and soundex). The system has been tested on 130 text-input sentences with several words ranging from 3 to 10 words. Results showed that the proposed LPU could achieve around 1.497ms processing time with an average success rate of 84.23% for a maximum of ten-word sentences.

Corresponding Author:

Haris Al Qodri Maarif
Department of Mechatronics Engineering
International Islamic University Malaysia
Jalan Gombak, Kuala Lumpur, 53100, Malaysia
Email: alqodri.maarif@gmail.com

1. INTRODUCTION

Sign language (SL) is the primary language and can be considered as the mother tongue for the HSI people. Many people who are born deaf learn sign language as their primary language, and it remains their preferred, or first, language. There is no written form of sign language, so deaf people communicate using reading and writing in their second or less preferred language. Therefore, a significant proportion of deaf people have a strong preference for accessing information in sign language rather than written text. Sign language only needs some important words compared to spoken language [1], [2]. Generally, sign language uses subject, verb, noun, and adverb. There are no other suffixes, prefixes, and particles. It is a non-verbal language that uses hand movement, hand orientation, face expression, head movement, posture, and body orientation [3]. Since sign language is a non-verbal language, the understanding of sign language has been compulsory for HSI people to communicate.

The awareness of sign language for non-HSI people is little, or many do not know sign language. Therefore, it provides an obstacle in communication in the community, especially if it needs interaction between non-HSI and HSI people. As obstacles arise in contact with the community, the communication bridge must fill the gap between them. The options are sign language translator and sign language synthesizer technology, translating spoken language to sign language [4].

Using a sign language translator to communicate between non-HSI and HSI has been limited since sign language translators are limited in Malaysia. As early as 2017, there are only less than 100 certified SL translators to cater to more than 30,000 persons of HSI (references). While in the world, the world federation of the deaf reported that there are about 70 million HSI people [4] and 138 living sign language, which is according to the ethnologue catalog [5].

Sign language synthesizer consists of three main modules, i.e., the voice recognition module, language processing unit module, and signing module. Each module has its components and algorithms which need a different approach to development. In this paper, the main focus is on the language processing module, which transforms the input language. The language processing module alters input language into output language that is suitable for output sign language. The input and output language are in the sequence of words (text), in which some methodology is required to do the transformation process properly.

The development of language processing units has been made and implemented in many different sign languages, for example, American sign language [6], British sign language [7], South African sign language [8], and Australian sign language [9]. However, in Malaysia, the language processing unit has not been implemented as an integral part of the sign language synthesizer. Furthermore, the language processing unit for bahasa isyarat Malaysia has not been implemented. A comprehensive review of the existing work and proposed work on the language processing unit is presented in this paper. In addition, various methods such as edit distance, natural language processing, HMM methods, and Bayesian network are discussed.

2. LITERATURE REVIEW

This section reviews the techniques of the language processing unit. The technique for the language processing unit provides a literature background for the language processing unit using natural language processing.

2.1. Natural language processing (NLP)

2.1.1. NLP basic processing

The necessary process of Natural language processing (NLP) that can be used for SL synthesizer is the most straightforward technique which has been implemented [10]. This technique only involves three basic operations, i.e., POS (part of speech) tagger, optimizer, and stemming. Figure 1 shows the step of this technique.

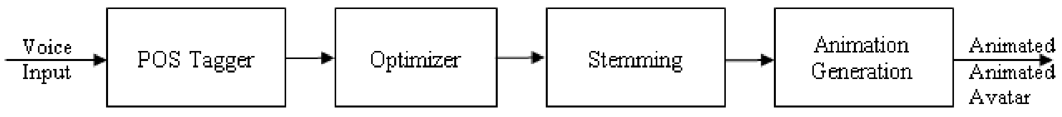


Figure 1. NLP basic processing [4]

The initial stage is the POS tagger, which involves morphological analysis. Then, as the POS tagger's output, the optimizer takes part in the following step to remove the unnecessary words. Finally, before the output is given to the animation step [11], the stemming process is involved in finding the words' primary form.

2.1.2. NLP with gloss-based approach

The gloss-based approach is a method that associates the words and their meanings through a dictionary [12]. The order of the language grammar defines the order of the glosses. In a report by Almeida *et al.* [13], the order of blocks (glosses) is calculated according to Portugal sign language (LGP) grammar. As the final step, the order of blocks was converted into the sign language order. Figure 2 shows the gloss-based approach technique by Almeida *et al*. [13]. The text was associated with the dictionary (database) to be processed in a natural language processing block. In addition, the output was translated to a sequence of glosses and actions.

A stemmer can be used to identify the stem and relevant suffixes (and prefixes), which allows inferring, for instance, the gender and the number of given words. A part-of-speech (POS) tagger can also contribute to the translation process, which couples with the stemmer in the identification of the different types of affixes. In addition, a POS tagger usually feeds further processing, for instance, named entity recognizers and syntactic analyzers. A named entity recognizer allows identifying persons' names and a syntactic analyzer to determine the sentence's syntactic components, such as subject and object.

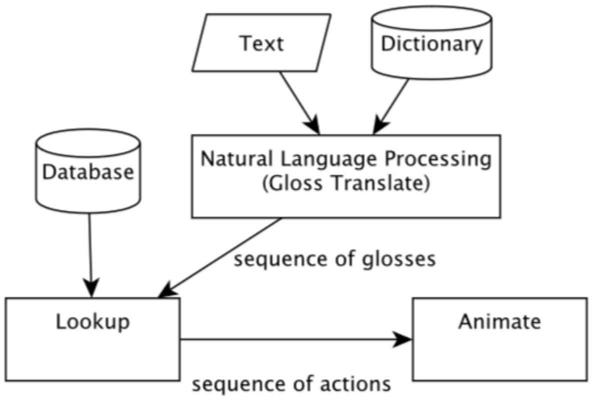


Figure 2. NLP with gloss based technique [13]

2.1.3. NLP with rule-based and statistical translation

The rule-based translation is a strategy that analyses the word's input until a group of words (sentence) [14]. The translation analysis finds specific combinations of words or signs (blocks) that generate a sign. The finding process starts from each word individually and extends the analysis to neighborhood context words or already-formed signs. There are two steps involved in the translation process. In the first one, every word is mapped to one or several syntactic pragmatic tags. The translation module then applies different rules that convert the tagged words into signs through grouping concepts or signs (blocks) and defining new signs. These rules can define short and extensive scope relationships between the concepts or signs. At the end of the process, the block sequence is expected to correspond to the sign sequence resulting from the translation process. The rule-based translation module provides the translation rules for the translation process. San-Segundo *et al*. [15] provided the evaluation tools for performance measures. Three available measurement tools have been considered: sign error rate (SER), position independent rate (PER), and bilingual evaluation understudy (BLEU).

The statistical translation method calculates the probability between the word sequence and sign sequence stored in a database as the reference [16]. One of the methods in statistical translation is phrase-based translation [15], [17]. Figure 3 shows the diagram of the phrase-based translation module used by San-Segundo *et al*. [15].

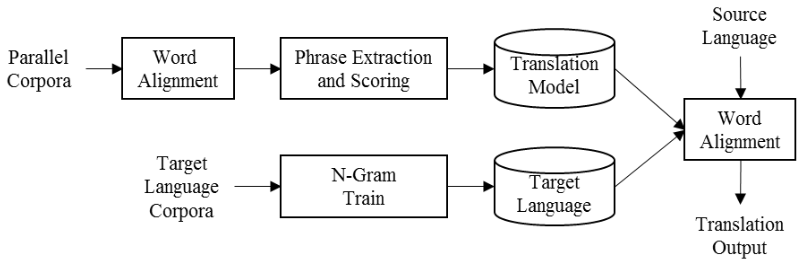


Figure 3. Diagram of phrase-based translation module [18]

The translation process uses a translation model based on phrases and a target language model as reported in San-Segundo *et al.* [19], [20], the GIZA++ software has been used to calculate the alignments between words and signs. San-Segundo *et al.* [15] reported that the statistical translation shows the worst outcome from the rule-based strategy. This condition is due to its restricted domain, and it has been possible to develop a complete set of rules with a reasonable effort.

2.2. Edit distance techniques

2.2.1. Levenshtein distance

Levenshtein distance (LD) is a technique for looking for the differences between two different strings and computing the two different phonetic strings' distance. The basic technique of LD involves three main processes, which are insertion, deletion, and substitutions. The LD is a method of aligning two phonetic segments. The enhancement was implemented in prior research, allowing only alignments of consonants with consonants and vowels with vowels [21].

The LD is a popular string metric used to evaluate strings on orthographic similarity in information theory. LD counts minimal substitutions, insertions, and deletions to edit one string into another of any length [22]. For word pairs with equal word length, LD produces only distances smaller or equal to the Hamming distance [23]. The Hamming distance counts the minimal number of substitutions needed to edit one string into another equal length [24].

The implementation of LD provides the distance calculation of two varieties of words [25]. The LD was used in the analysis of linguistic variations in many other languages, for example, German [26], Dutch [27], Frisian [28], and Bulgarian [29]. The other successful implementation was tested on 15 Norwegian dialects perceptually and acoustically [30].

2.2.2. Mahalanobis distance

Mahalanobis distance is the distance between two samples based on their mean feature vectors $\mu_a$and μ_b, and the covariance matrix Σ of the features across all samples in a database. The Mahalanobis distance is given as (3) [31].

$$D_M(\mu_a, \mu_b) = (\mu_a - \mu_b)^T \Sigma^{-1}(\mu_a - \mu_b) \tag{3}$$

The mahalanobis distance metric is scaled according to the precision matrix (the covariance matrix's inverse) [32]. It provides a way of reducing the influence of distances along dimensions irrelevant to the current descriptive word and normalizing distances across different feature spaces to create a single distance value for object classification. The mahalanobis distance metric can be seen as a feature weighting within dimensions of features and exclusive features. For example, the lightness dimension of color space varies more than the color dimensions for a given color word. Therefore, distance in the lightness dimension has a reduced effect on classification. Scaling features in this manner also allow us to combine disjoint features of varying dimensions and distributions, allowing greater flexibility for future features [33].

Mahalanobis distance is essentially a distance measure based on correlations between variables by which different patterns can be identified and analyzed. It is a useful way of determining the similarity of an unknown sample set to a known one. Distance-based approaches calculate the distance from a point to a particular point in the data set. Distance to the mean, the average distance between the query point and all points in the data set, the maximum distance between the query point and data set points are examples of the many options. Whether a data point is close to the data set depends on the user's threshold [34].

Mahalanobis distance is a distance between two points $x = (x_1, x_2, \dots, x_p)^t$ and $y = (y_1, y_2, .., y_p)^t$ in the p dimensional space R_p is defined as (4) [35].

$$d_s(x, y) = \sqrt{(x - y)^t S^{-1}(x - y)} \tag{4}$$

Where $d_s(x, 0) = \|x\|_s = \sqrt{x^t S^{-1} x}$ is the norm of x and S^{-1} is a positive semi-definite covariance matric. Points with the same distance of the origin $\|x\|_s = c$ satisfy $x^t S^{-1} x = c^2$ which is the general equations of an ellipsoid centered at the origin, and we are interested in the distance of an observation from its center $\bar{x}$ given by (5).

$$d_s(x, \bar{x}) = \sqrt{(x - \bar{x})^t S^{-1}(x - \bar{x})} \tag{5}$$

The mahalanobis distance's drawback is the equal adding up of the variance normalized squared distances of the features. In the case of noise-free signals, this leads to the best possible performance. But suppose the feature is distorted by noise due to the squaring of the distances. In that case, a single feature can have such a high value that it covers the other features' information and leads to a misclassification [35].

Therefore, to find classification procedures more robust to noise, we have to find a distance measure that gives less weight to the noisy features and more weight to the clean features. It is reached by comparing the different input features to decide which feature should be given less weight or excluded and have more weight [36], [37].

2.2.3. Soundex distance

Phonetic encoding techniques consider a word phonetic transcription for classification and coding purposes, such as correcting eventual spelling mistakes and classifying phonetically digital libraries, dictionaries, and databases [38]. The phonetic representation has several applications. First, it allows to search concepts based on pronunciation rather than spelling [38]-[40].

The soundex phonetic technique was mainly used in applications involving searching people's names like air reservation systems, censuses, and other tasks presenting typing errors due to phonetic similarity [41]. Schütze *et al*. [42] reported that the soundex technique evaluates each letter in the input word and assigns a numeric value that converts each word into a code made up of four elements [43]. Thus, soundex uses numeric codes for each letter of the string to be codified, as shown in Table 1.

Table 1. Soundex phonetic codes for English alphabet

Numeric Code	Letter
0	a, i, u, e, o, y
1	b, p, f, v
2	c, g, j, k, q, s, x, z
3	d, t
4	L
5	m, n
6	r

2.2.4. N-grams distance

An N-gram is a sub-sequence of N items from a given sequence. N-grams are used in various areas of statistical natural language processing and genetic sequence analysis. The items in question can be characters, words, or base pairs according to the application. For example, this N-gram output can be used for statistical machine translation and spell checking [44].

Pattern extraction is the process of parsing a sequence of items to find or extract a particular pattern of items. Pattern length can be fixed, as in the n-gram model, or it can be variable. Variable-length patterns can be directives to specific rules, like regular expressions. However, they can also be random and depend on the context and pattern repetition in the patterns dictionary [45]-[47].

3. PROPOSED ADAPTIVE LANGUAGE PROCESSING UNIT

The proposed approach, shown in Figure 4, involves text classifiers, where they classify text input to its corresponding word tagging. The system implements NLP, HMM, and Bayesian as an adaptive combination module. The HMM and Bayesian Network are implemented together to cover the various lengths of the input text, in which the Bayesian Network handles longer sentences, while HMM handles shorter sentences. Such an adaptive selection of classifiers in the proposed system allows for longer sentences to be performed accurately.

The proposed system language processing unit contains word identification and tagging and an adaptive classifier, automatically selecting the classifier, either HMM or Bayesian Network, based on the number of words detected. The proposed system is evaluated based on its success rate and processing time. The success rate indicates whether the output from the language processing unit (LPU) contains all the important words with the correct structure (subject-predicate-object) or not. If all-important words detected from the speech are included in the output, then the system is considered a success. Otherwise, it is considered a failure. The processing time is defined as the LPU's measured time, starting from inputting data until obtaining the output from the LPU.

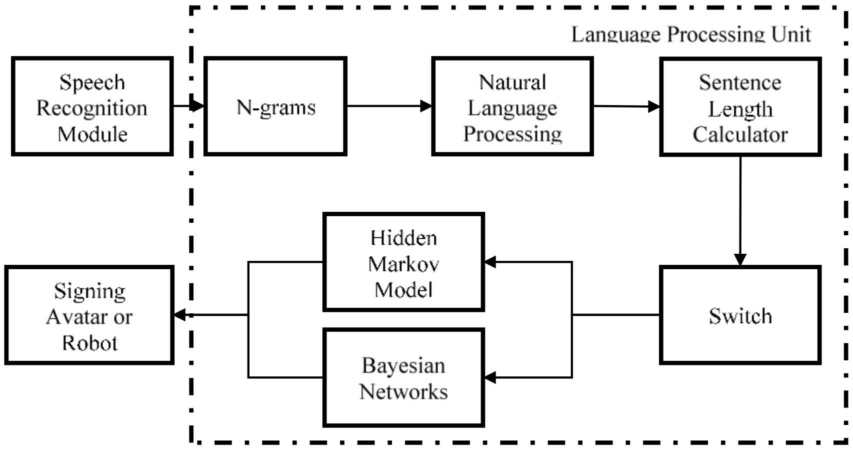


Figure 4. Block diagram of the proposed adaptive language processing unit

3.1. N-grams

In computational linguistics, a sequence of n contiguous words is called *N-gram* [46-48]. The system implements a combination of n-grams and NLP. The proposed system, LPU, is based on n-grams and NLP. Figure 5 shows that the output from N-grams is fed to the LPU. Let the input to the LPU be on the group of words ($n = 2$). NLP subsequently processes the processed text input to get the proper sentences represented by a particular sign language's signs. In an N-gram, sentences are truncated to the length $(n-1)$ and its truncation probability is defined as (6).

$$p(w_i|w_1 \dots, w_{i-1}) = p(w_i|w_{i-n+1} \dots, w_{i-1}) \tag{6}$$

Since $n = 2$, it has been called a bigram. In such case, the N-gram conditional probabilities $p(w_i|w_{i-1})$ can be estimated from raw text $C(w_{n-1}w_n)$ based on the relative frequency of word sequences $C(w_{n-1})$ as (7) and (8).

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{7}$$

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{8}$$

3.2. Natural language processing

The LPU performs the text-translated process according to Sign Language's grammatical rules, in this case, the Malaysian sign language (MSL). First, the proposed method identifies the "important" words that the proposed SL synthesizer synthesizes. Then, it utilizes the "tagging" process, which labels the input word into specific structure categories, i.e., subject, predicate, and object (S-P-O).

NLP is the essential process in the LPU. Figure 5 shows the detail of the NLP. It involves a tokenizer, POS tagger, named entity extraction, stemmer, and lexical transfer. The steps are required to enhance the translation and the output of the system.

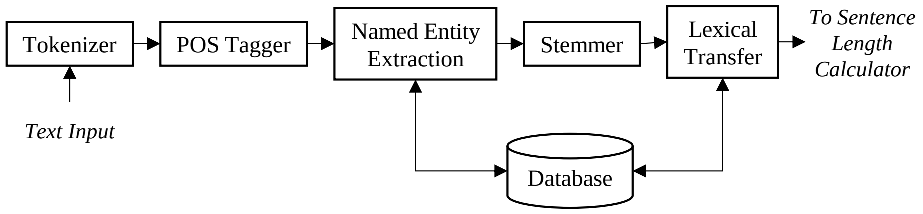


Figure 5. Word processing of the natural language processing

3.2.1. Tokenizer and POS tagger

Tokenising is a basic operation of NLP that is applied to an input text. It breaks up a stream of characters into words, punctuation marks, numbers, and other discrete items. Figure 6 shows the flowchart of the tokenizer. This NLP stage does not classify grammatical categories of the input text. It also does not consider any information on the syntactic structure of the text or the type of words in it (e.g., whether the words are verbs and nouns). The input of the tokenizer is the identified words (text input). In comparison, the output is the corresponding tokenized words. The last word is also tagged to indicate that it is the last token for the current input, and usually, the last word is a noun.

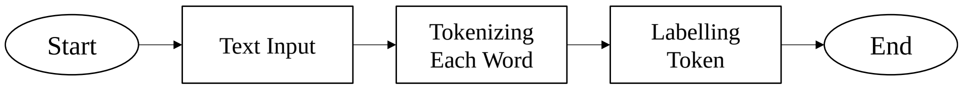


Figure 6. Flowchart of tokenizer

The POS tagger allows classifying the words into nine traditional word classes, i.e., noun, verb, adjective, adverb, preposition, article, interjection, pronoun, and conjunction. In addition, each word entering the POS tagging is labeled according to its status in a sentence. Thus, it contributes to the translation process downstream [49], [50].

Each word entering the POS Tagging block is labeled according to its status in a sentence. Figure 7 shows the tagging process where each word is tagged according to its particular label. Each token is tagged with its corresponding SPO category (i.e., as a subject, predicate, or object). The prepositions and other non-important words are not tagged and thus are discarded. For the same example as before, the tagged token as token 1 as "S" (subject), token 2 as "P" (predicate), token 3 as "O" (object), token 4 as "unknown," and token 5 as "O" (object).

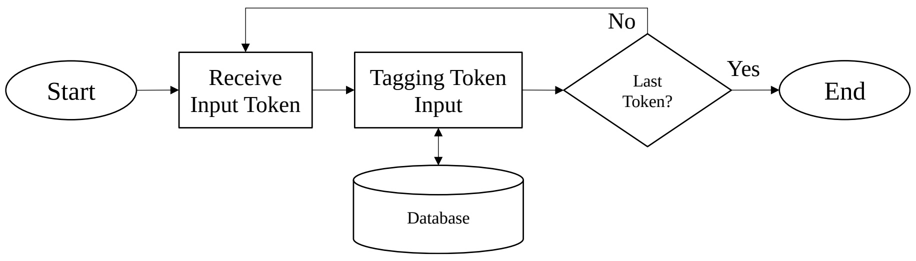


Figure 7. Flowchart of POS tagger

3.2.2. Named entity extraction

Named entity extraction identifies types of tagged words, such as names of persons, verbs, and removes words with unknown tagged tokens, as shown in Figure 8. The syntactic analysis allows the identification of the SPO syntactic components of the sentence. For example, if the tokenized word is a noun, the syntactic analysis validates the appropriateness of the tag given to the tokenized words.

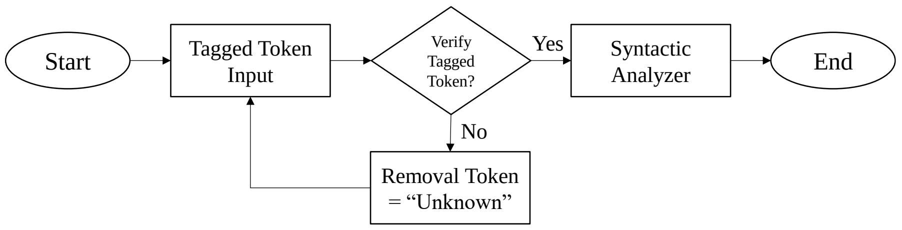


Figure 8. Flowchart of the named entity extraction and syntactic analyzer

For the same example, the processed token by NEE (after POS tagger) is resulted as token 1 as "S" (subject), token 2 as "P" (predicate), token 3 as "O" (object), and token 5 as "O" (object). Then, after passing the Syntactic Analyzer process, the resulted token is token 1 as "O" (object), token 2 as "P" (predicate), token 3 as "S" (subject), and token 5 as "S" (subject).

3.2.3. Stemmer and lexical transfer

The stemmer is used to identify basic forms (*stems*) of words allowing to infer gender information and the number of input words. It is aimed to map a speech (text) in a particular language to a corresponding sign in the target SL. It uses a language dictionary to perform an accurate reduction to root words. Figure 9 shows the flowchart for the stemming process. Stemming uses ordinary pattern matching to strip suffixes of tokens simply (e.g., remove "-s," and remove "-ing,", in the word endings), thus "stripping off" typical grammar. In the proposed system, the stemmer is used to identify verbs only, tagged as Predicate. This stage is less used in some languages (including Bahasa Melayu), where no tense-dependent changes of root words are needed. However, affixes may be used to give extra emphasis to the meaning of the root words. Also, they might be applied to derive new words (usually-verbs) that have different meanings though still relate to the root ones. For example, token 2 "dimakan" is tagged as "P." The tagging "P" refers to the verb, where it removes "di" as a prefix for the word "makan." token 2 changes into "makan" and with the same tag "P" by having this process.

The lexical transfer involves one-to-one mapping of the input sentences to their corresponding meaning. It requires referring to the dictionary and word database, see in Figure 10. Specifically, the stage allows distinguishing words having multiple meanings. If the words have two or more meanings, the meaning based on the SPO tagging information of the sentence is selected.

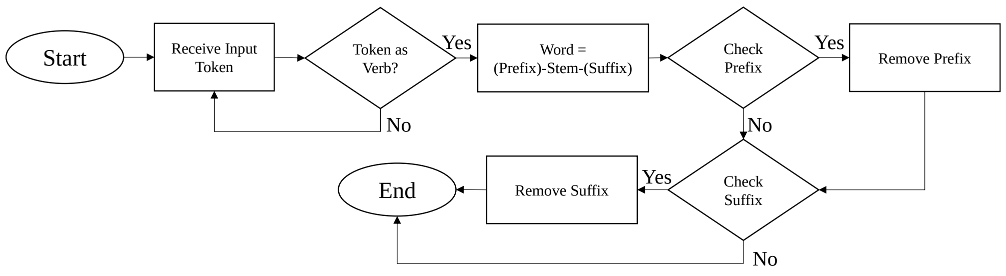


Figure 9. Flowchart of the stemming process

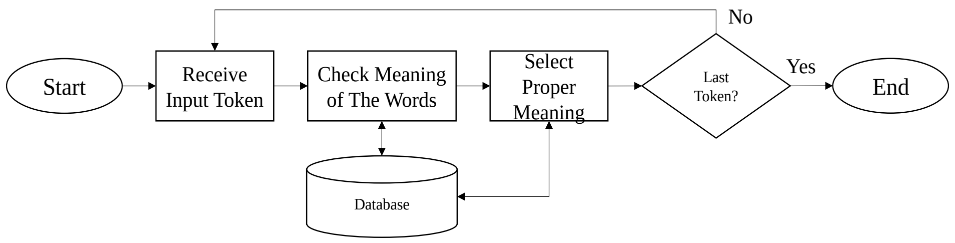


Figure 10. Flowchart of the lexical transfer

3.3. Sentence length calculator and adaptive selection of the classifier

The sentence length calculator (SLC) is a string processing that calculates the number of words in a sentence. In this research, any input given to the systems is limited to be one complete sentence, and the word count is the number of words in the input sentence. Figure 11 shows the flowchart for the SLC. In addition, the proposed system offers an adaptive selection of the classifier, allowing for switching between HMM and Bayesian networks automatically.

The HMM and Bayesian Network identify the sentence and process it such that it follows the pre-assigned order, i.e., subject, predicate, and object (SPO), based on the number of words in the sentence. Based on the experimental result, HMM works for short sentences (threshold = 7 words), and Bayesian networks applies for longer sentences (more than 7 words). Once the resulting sentence has been arranged into the SPO order, the sentence is fed into the animation part, which allows the animated avatar.

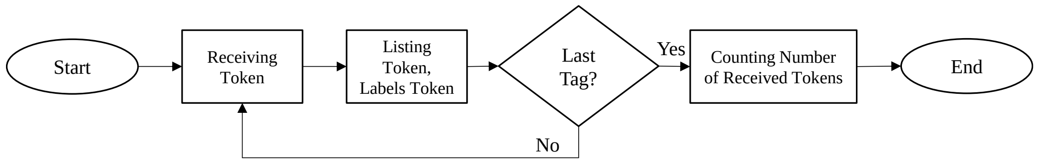

Start
Receiving Token
Listing Token, Labels Token
Last Tag?
Yes
No
Counting Number of Received Tokens
End

Figure 11. Flowchart of the sentence length calculator

HMM states and transition matrix are designed to organize results into Subject, Predicate, and Object patterns. HMM involves three states and a single output. It has three observation probability distributions B and three state transition probabilities A. for each state, B emits a single output V. The mathematical model is described as follows:

a) $\lambda = \{A, B, \pi\}$
b) $A = \{a_{11}, a_{12}, a_{22}, a_{23}, a_{33}\}$
c) $B = \{subject, predicate, object\}$

The initial condition π is defined as the basic words, which, in our case, has three words, i.e., subject (S), predicate (P), and object (O). "subject" is defined as a person or something which does the action. "Predicate" is classified as an active verb that indicates the activity which is done by the subject. Finally, "object" is classified as a person or object, which is the subject's goal. The SPO structure is proposed to ease the synthesizer process, where the complete sentence should consist of these three basic elements SPO.

The Bayesian network is applied to a condition where the number of words is more than 7. Then, the joint probability distribution represents the implementation of the Bayesian network. In this case, the probabilistic distribution is the relationship among subject (S), predicate (P), and object (O). The joint probability distribution can be written as (9).

$$p(S, P, O) = p(S)p(P/S)p(O|S, P) \quad (9)$$

The CPD is calculated from the joint probability, and the Bayesian network consists of class variables and feature variables that are readily applicable to the classification task. The S (subject) is selected as the class variable, and the calculation for the probability of $S = T$ given any observed value set (P, O) as (10).

$$p(S = T|P, O) = \frac{p(S=T,P,O)}{p(S=T,P,O)+ p(S=F,P,O)} \quad (10)$$

Where $p(S = T|P, O)$ and $p(S = F, P, O)$ can be computed efficiently using (9). Similarly, It can be applied to calculate $p(S = F, P, O)$. Then the value of S is determined by computing $p(S = F, P, O)$ and $p(S = T, P, O)$. Once the resulting sentence is arranged into the SPO order, it is fed into the animation avatar (or robotic manipulator), outputting it in the chosen SL lexis.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Experimental setup

The data used is text-based input, which is considered a simple Malay sentence. However, combining the number of words in one sentence, from three to ten, indicates input sentences from simple sentences into more complex sentences. Table 2 shows 13 sample data selected from the 130 data used in this experiment. The simple structure consists of only SPO structure, whereas the more complex sentences have SPO structure and random order of sentence structures. The latter is introduced to evaluate the proposed SL synthesizer technique based on the processing time and success rate.

The proposed system has been developed using Matlab 2018a and running at Intel i5 5200U processor and 4 GB RAM to process the output from speech data. To evaluate the performance of the system, we performed four types of comparisons. The first comparison involves evaluating four different edit distance methods, namely Levenshtein, soundex, N-grams, and mahalanobis. The second comparison consists of the selection of the number of parameters used for the selected edit distance. The next comparison is made to evaluate the NLP and HMM's performance with NLP and Bayesian Network. Then, we can determine the threshold number of words in the input sentences for the adaptive classification. Finally, the last evaluation is on the edit distances with the adaptive system of the classifier.

Table 2. Sample input selected for language processing unit

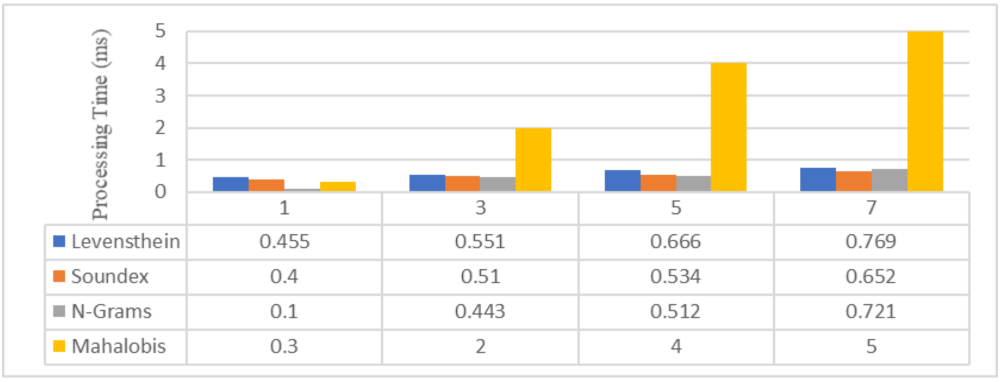
No	Selected Input	Number of Words
1	Saya Makan Nasi *(I eat rice)*	3
2	Nasi Saya Makan *(Rice I eat)*	3
3	Dia Makan Nasi *(He/she eats rice)*	3
4	Dia Pergi ke Pasar *(He/she goes to Market)*	4
5	Ke Pasar Saya Pergi *(To market I go)*	4
6	Dia Makan Nasi di Pasar *(He/she eats rice at market)*	5
7	Saya dan Dia Makan Nasi *(I and He eat rice)*	5
8	Nasi Saya dan Dia Makan *(Rice I and he/she eat)*	5
9	Saya Dia Pergi Makan Pasar Nasi *(I, he go eat market rice)*	6
10	Saya Makan Nasi dan Pergi ke Pasar *(I eat rice and go to market)*	7
11	Saya Makan Nasi Dan Membeli Buah Di Pasar (*I Eat Rice and Buy a Fruit at The Market*)	8
12	Saya dan Dia Makan Nasi dan Buah Di Pasar (*I and He/She Eat Rice and Fruit in The Market*)	9
13	Saya Makan Nasi dan Pergi ke Pasar di Pagi Hari *(I Eat Rice and Go to the Market in The Morning)*	10

4.2. Selection of edit distance and its characteristics

Each edit distance is used in the proposed system, and the performance of each is compared based on its success rate and processing time. The selection is performed to determine the most effective edit distance and its characteristics value. The edit distance, which provides the highest success rate and the shortest processing time, is selected. The evaluation is carried out by comparing the processing time of each edit distance. It is necessary for getting a fast response for the whole system for the SL synthesizer. The comparison is performed by varying the number of words in a sentence. The comparison aims to find the significant edit distance to support the SL synthesizer's proposed language processing unit. In this research, determining the most suitable edit distance technique is key for the proposed system to provide the fastest processing time and the highest success rate.

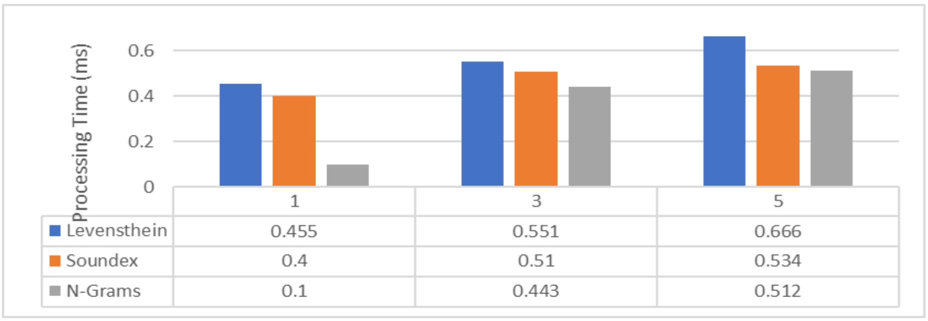
Figure 12 shows the performance of each edit distance with the various number of words in a sentence. It indicates that Levenshtein, soundex, and N-grams provided similar processing times. The similarity of processing time resulting from three edit distances is expected due to those similar characteristics. As n-grams have provided the fastest processing time than mahalanobis and soundex distance, N-grams have been selected to be implemented in the proposed system's language processing unit.

Figure 13 shows the time comparison for Levenshtein, soundex, and n-grams. Levenshtein and soundex distance resulted in a longer processing time than n-grams distance. It is demonstrated that Levenstein, soundex, and mahalanobis have a longer processing time than n-grams. N-grams, at n = 2, provide efficient processing time compared to other edit distances. Therefore, N-grams are implemented for the language processing unit.



	1	3	5	7
🟦 Levensthein	0.455	0.551	0.666	0.769
🟧 Soundex	0.4	0.51	0.534	0.652
⬜ N-Grams	0.1	0.443	0.512	0.721
🟨 Mahalobis	0.3	2	4	5

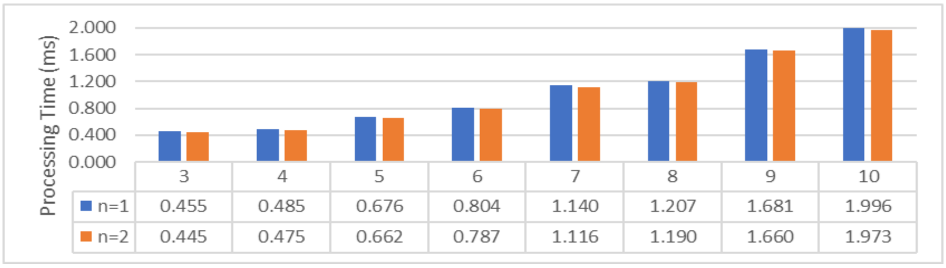
Figure 12. Time comparison for four edit distance in various number of words



	1	3	5
■ Levensthein	0.455	0.551	0.666
■ Soundex	0.4	0.51	0.534
■ N-Grams	0.1	0.443	0.512

Figure 13. Time comparison for Levenshtein, soundex, and N-grams edit distance

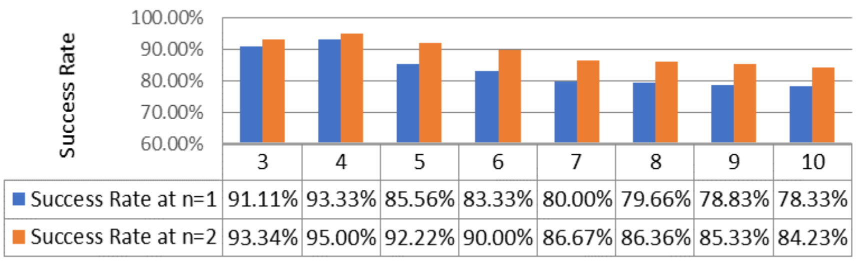
N-grams distance is set by the number *n*, which shows the grouping of words from a set of sentences. The comparison of the processing time for n-grams at various numbers *n* provides different processing times from LPU. Figure 14 shows the comparison of the processing time for different values of n-grams over the number of words. It can be seen that n-grams at *n* = 2 provide the fastest processing time, which is better than n-grams at *n* = 1.



	3	4	5	6	7	8	9	10
■ n=1	0.455	0.485	0.676	0.804	1.140	1.207	1.681	1.996
■ n=2	0.445	0.475	0.662	0.787	1.116	1.190	1.660	1.973

Figure 14. Processing time for N-grams at n = 1, and n = 2 over various number of words

The success rate over the various number of words in a sentence is shown in Figure 15. The different number of n produces multiple success rates. N-grams at $n = 2$ provide a higher success rate with more than 80% success rate than n-grams at $n = 1$ (less than 80% for 10-word input sentences). The success rate at $n = 2$ decreases 9.11% when the number of words increases from 3 to 10.



	3	4	5	6	7	8	9	10
■ Success Rate at n=1	91.11%	93.33%	85.56%	83.33%	80.00%	79.66%	78.83%	78.33%
■ Success Rate at n=2	93.34%	95.00%	92.22%	90.00%	86.67%	86.36%	85.33%	84.23%

Figure 15. Success rate for N-grams at n = 1, and n = 2 over various number of words

4.3. Performance Evaluation

The implementation of n-grams and HMM provides the fastest processing time for the proposed LPU. Figure 16 provides the processing time average success rate for LPU using HMM and Bayesian network for various numbers of words. The processing time required for both techniques are varied depending on the number of words in the input sentence. The adaptive implementation of HMM and

Bayesian network is in place to achieve the fastest processing time depending on the number of words in the input sentence.

The processing time increases over various words, and the success rate decreases over multiple words. The processing time for LPU using HMM increased by 4.5 times when there was an increase in the number of words from 3 to 10. The processing time for LPU using Bayesian network increased by 2.2 times longer when there was an increase in the number of words from 3 to 10. For the number of words between 3 to 7 words, LPU using HMM provides an average of 69.89% processing time than NLP with Bayesian network. On the other hand, when LPU uses Bayesian network, the number of words is 7 to 10 words average 88.35% processing time than NLP with HMM. In addition, the average success rate starts at (number of words = 3) 93.75% and slightly decreases up to 84.49% at 10 words in a sentence.

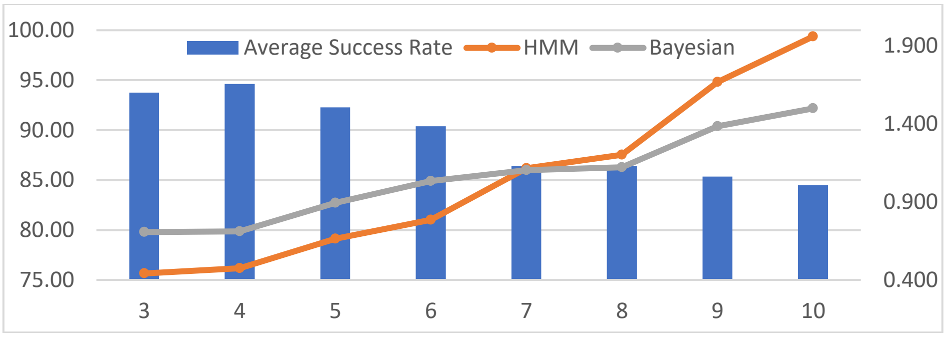


Figure 16. Processing time for the LPU using HMM and Bayesian network

5. CONCLUSION

This paper has investigated the utilization of the n-grams, NLP, HMM, and the Bayesian network for the classifier. It involved measuring the processing time and success rate compared to other edit distances. The proposed system has adapted n-gram as the edit distance and combined it to HMM and Bayesian network to obtain efficient (fast) processing time and a high success rate. Evaluation and selection of appropriate edit distance were achieved by comparing the processing time of each selected edit distance. As a result, n-grams have been selected to be implemented on the LPU. Furthermore, the adaptive language processing Unit has implemented n-grams with NLP, HMM, and Bayesian network for the proposed algorithm, and it has been enabled by implementing SLC to decide either HMM or Bayesian network is used to adjust the output to match the SPO sequence. The proposed system's performance was reached over various sentence lengths. The processing time and success rate reveal that the proposed system outperforms the other edit distances. As a result, the proposed LPU could achieve around 1.449 ms processing time with an average success rate of 84.49% for a maximum of ten-word sentences.

ACKNOWLEDGMENT

The authors gratefully acknowledge the use of service and facilities of the Intelligent Mechatronics System Research Unit laboratory at International Islamic University Malaysia. This study receives funding from the International Islamic University Malaysia, under the Research Grants-Endowment Fund B - IIUM.

REFERENCES

[1] M. M. Islam, S. Siddiqua and J. Afnan, "Real time Hand Gesture Recognition using different algorithms based on American Sign Language," *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 2017, pp. 1-6, doi: 10.1109/ICIVPR.2017.7890854

[2] Z. Kang, "Spoken Language to Sign Language Translation System Based on HamNoSys," *SSPS 2019: Proceedings of the 2019 International Symposium on Signal Processing Systemss*, 2019, pp. 159-164, doi: 10.1145/3364908.3365300.

[3] S.C. Ong, and S. Ranganath, "Automatic Sign Language Analysis: A Survey and The Future Beyond Lexical Meaning," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 6, no. 6, pp. 873-891, 2005, doi: 10.1109/TPAMI.2005.112.

[4] J. Joy and K. Balakrishnan, "A Prototype Malayalam to Sign Language Automatic Translator," ArXiv Preprint:1412.7415, 2014. [Online]. Available: https://arxiv.org/abs/1412.7415.

[5] A. Karpov, I. Kipyatkova, and M. Zelezny, "Automatic Technologies for Processing Spoken Sign Languages," *Procedia Computer Science*, vol. 81, pp. 201-207, 2016, doi: 10.1016/j.procs.2016.04.050.

[6] R. Wolfe , P. Cook, J. C. McDonald and J. Schnepp, "Linguistics as Structure in Computer Animation: Toward a More Effective Synthesis of Brow Motion in American Sign Language," *Sign Language & Linguistics*, vol. 14, no. 1, pp. 179-199, 2011, doi: 10.1075/sll.14.1.09wol.

[7] D. Murph, *IBM's SiSi Virtually Translates Speech to Sign Language*, 2007. [Online]. Available: https://www.engadget.com/2007-09-13-ibms-sisi-virtually-translates-speech-to-sign-language.html.

[8] L. V. Zijl and G. Olivrin, "South African Sign Language Assistive Translation," *IASTED - International Association of Science and Technology for Development*, 2008.

[9] S. Yeates, E.-J. Holden, and R. Owens, "An Animated Auslan Tuition System," *Machine Graphics and Vision*, vol. 12, no. 2, pp. 203-214, 2003.

[10] R. Bharti, S. Yadav, and S. Gupta, "Automated Speech to Sign language Conversion using Google API and NLP," *Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAEEC) 2019*, doi: 10.2139/ssrn.3575439.

[11] R. Elliott, J. R. W. Glauert, J. R. Kennaway, and I. Marshall, "The Development of Language Processing Support for The ViSiCAST Project," *Assets '00: Proceedings of the fourth international ACM conference on Assistive technologies*, 2000, pp. 101-108, doi: 10.1145/354324.354349.

[12] F. Robertson, *Word sense disambiguation for Finnish with an application to language learning.* 2020.

[13] I. Almeida, L. Coheur, and S. Candeias, "Coupling Natural Language Processing and Animation Synthesis in Portuguese Sign Language Translation," *Proceedings of the Fourth Workshop on Vision and Language*, 2015, pp. 94-103, doi: 10.18653/v1/W15-2815.

[14] U. Inurrieta, I. Aduriz, A. D. Ilarraza, G. Labaka and K. Saraso, "Rule-based translation of Spanish Verb-Noun combinations into Basque," *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, 2017, pp. 149-154, doi: 10.18653/v1/W17-1720.

[15] R. San-Segundo, *et al.*, "Speech to Sign Language Translation System for Spanish," *Speech Communication*, vol. 50, no. 11-12, pp. 1009-1020, 2008, doi: 10.1016/j.specom.2008.02.001.

[16] M. Davydov and O. Lozynska, "Information system for translation into ukrainian sign language on mobile devices," *2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 2017, pp. 48-51, doi: 10.1109/STC-CSIT.2017.8098734.

[17] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. pp. 5039–5049, doi: 10.18653/v1/D18-1549.

[18] R. San-Segundo, *et al.*, "Design, development and field evaluation of a Spanish into sign language translation system," *Pattern Analysis and Applications*, vol. 15, no. 2, pp. 203-224, 2012, doi: 10.1007/s10044-011-0243-9.

[19] R. San-Segundo, R. Barra, L. F. D'Haro, J. M. Montero, R. Córdoba and J. Ferreiros, "A Spanish Speech to Sign Language Translation System for Assisting Deaf-Mute People," in *Ninth International Conference on Spoken Language Processing*, 2006, pp. 1399-1402, doi: 10.21437/Interspeech.2006-420.

[20] R. San-Segundo, A. Pérez, D. Ortiz, L. F. D'Haro, M. Inés Torres and F. Casacuberta, "Evaluation of Alternatives on Speech to Sign Language Translation," *Eighth Annual Conference of the International Speech Communication Association*, 2007, pp. 2529-2532, doi: 10.21437/Interspeech.2007-672.

[21] E. Valls, J. Nerbonne, J. Prokic, M. Wieling, E. Clua, M.-R. Lloret, "Applying The Levenshtein Distance to Catalan Dialects: A Brief Comparison of Two Dialectometric Approaches," *Verba: Anuario Galego de Filoloxía*, vol. 39, pp. 35-61, 2012.

[22] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions," *Insertions, and Reversals, Soviet Physics Doklady*, vol. 10, no. 8, pp. 707-710, 1966.

[23] J. J. Schepens, T. Dijkstra and F. Grootjen, "Distributions of Cognates in Europe Based On The Levenshtein Distance," *Bilingualism: Language and Cognition*, vol. 15, pp. 157-166, 2012, doi: 10.1017/S1366728910000623.

[24] D. Bazeia, J. Menezes, B. F. de Oliveira and J. G. G. S. Ramos, "Hamming distance and mobility behavior in generalized rock-paper-scissors models," *EPL (Europhysics Letters)*, vol. 119, no. 5, p. 58003, 2017, doi: 10.1209/0295-5075/119/58003.

[25] B. Kessler, "Computational Dialectology in Irish Gaelic," *EACL '95: Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, 1995, pp 60-66, doi: 10.3115/976973.976983.

[26] N. Korchagina, "Normalizing Medieval German Texts: from rules to deep learning," *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language,* 2017, pp. 12-17.[Online]. available: https://aclanthology.org/W17-0504/.

[27] E. Lefever, S. Labat, and P. Singh, "Identifying Cognates in English-Dutch and French-Dutch by means of Orthographic Information and Cross-lingual Word Embeddings," *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 4096–4101. [Online]. Available: https://aclanthology.org/2020.lrec-1.504.

[28] E. Bosma and N. Nota, "Cognate facilitation in Frisian–Dutch bilingual children's sentence reading: An eye-tracking study," *Journal of experimental child psychology*, vol. 189, p. 104699, 2020, doi: 10.1016/j.jecp.2019.104699.

[29] A. Georgiev and K. Stefanov, *Bulgarian Open Science Digital Library-First Prototype.* 2019. [Online]. Available: https://dipp.math.bas.bg/images/2019/251-258_8_3.8_sDiPP2019-66_f_v.1.F_20190908.pdf.

[30] W. Heeringa, and C. Gooskens, "Norwegian Dialects Examined Perceptually and Acoustically," *Computers and the Humanities*, vol. 37, no. 3, pp. 293-315, 2003, doi: 10.1023/A:1025087115665.

[31] M. I. Mandel and D. P. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005, pp. 594-599, doi: 10.5281/zenodo.1415024.

[32] P. C. Mahalanobis, "On The Generalized Distance in Statistics," *National Institute of Science of India*, 1936.

[33] I. Perera and J. Allen, "SALL-E: Situated Agent for Language Learning," *AAAI Conference on Artificial Intelligence*, 2013, pp. 1241-1247.

[34] J. Jaworska, N. Nikolova-Jeliazkova, and T. Aldenberg, "QSAR Applicability Domain Estimation by Projection of The Training Set in Descriptor Space: A Review," *Alternatives to Laboratory Animals*, vol. 33, no. 5, pp. 445-459, 2005, doi: 10.1177/026119290503300508.

[35] S. Kapoor, S. Khanna, and R. Bhatia, "Facial Gesture Recognition Using Correlation and Mahalanobis Distance," AarXiv Preprint:1003.1819, 2010. [Online]. Available: https://arxiv.org/abs/1003.1819.

[36] M. Wölfel and H. K. Ekenel, "Feature weighted mahalanobis distance: Improved robustness for Gaussian classifiers," 2005 13th European Signal Processing Conference, 2005, pp. 1-4.

[37] T. Reitmaier and B. Sick, "*The responsibility weighted Mahalanobis kernel for semi-supervised training of support vector machines for classification*," Information Sciences, vol. 323, pp. 179-198, 2015, doi: 10.1016/j.ins.2015.06.027.

[38] M. Z. Alksasbeh, B. A. Y. Alqaralleh, T. Abukhalil, A. Abukaraki, T. A. Rawashdeh and M. Al-Jaafreh, "Smart detection of offensive words in social media using the soundex algorithm and permuterm index," *International Journal of Electrical & Computer Engineering*, vol. 11, no. 5, pp. 4431-4438, 2021, doi: 10.11591/ijece.v11i5.pp4431-4438.

[39] D. Pinto, D. Vilariño, Y. Alemán, H. Gómez, N. Loya and H. Jiménez-Salazar, "The Soundex Phonetic Algorithm Revisited for SMS Text Representation," *International Conference on Text, Speech and Dialogue*, vol. 7499, 2012, doi: 10.1007/978-3-642-32790-2_5.

[40] R. Anand, R. Mahajan, N. Verma, P. Singh, "Soundex Algorithm for Hindi Language Names," *Advances in Data Sciences, Security and Applications*, pp. 285-293, 2020, doi: 10.1007/978-981-15-0372-6_22.

[41] D. E . Knuth, *Art of Computer Programming*, Addison-Wesley Professional, 2014.

[42] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*, vol. 39, Cambridge University Press Cambridge, 2008.

[43] V. Gautam, A. Pipal, and M. Arora, "SoundEx Algorithm Revisited for Indian Language," *International Conference on Innovative Computing and Communications*, vol. 56, pp. 47-55, 2019, doi: 10.1007/978-981-13-2354-6_6.

[44] S. Wankerl, E. Nöth, and S. Evert, "An N-Gram Based Approach to the Automatic Diagnosis of Alzheimer's Disease from Spoken Language," in *INTERSPEECH* 2017, pp. 3162-3166, 2017, doi: 10.21437/Interspeech.2017-1572.

[45] H. Mostafa, "N-gram and Fast Pattern Extraction Algorithm," 2007, [Online]; Avilable: http://www.codeproject.com/Articles/20423/N-gram-and-Fast-Pattern-Extraction-Algorithm.

[46] Al-Sarem, M. and A.-H. Emara, "The effect of training set size in authorship attribution: application on short Arabic texts," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 652-659, 2019, doi: 10.11591/ijece.v9i1.pp652-659.

[47] M. D. R. Rahman, M. D. T. Habib, M. D. S. Rahman, G. Z. Islam and M. D. A. A. Khan, "An exploratory research on grammar checking of Bangla sentences using statistical language models," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, ppp. 3244-3252, 2020, doi: 10.11591/ijece.v10i3.pp3244-3252.

[48] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh and L. Chanona-Hernández, "Syntactic n-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853-860, 2014, doi: 10.1016/j.eswa.2013.08.015.

[49] K. K. Purnamasari and I. Suwardi, "Rule-based Part of Speech Tagger for Indonesian Language," *IOP Conference Series: Materials Science and Engineering*, vol. 407, p. 012151, 2018, doi: 10.1088/1757-899X/407/1/012151.

[50] D. L. Cing and K. M. Soe, *Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar Language*, *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 2023-2030, 2020, doi: 10.11591/ijece.v10i2.pp2023-2030.