INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Tamil Speech Synthesizer App for Android: Text Processing Module Enhancement

A Arulprakash[1], M Synthiya[1], T Vijila[1], C Rajabhusanam[2]*

1 Assistant Professor, Department of Computer Science Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nādu, India
2 Professor, Department of Computer Science Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

OPEN ACCESS

Received: 10-11-2022

Accepted: 17-01-2023

Published: 21-02-2023

Citation: Arulprakash A, Synthiya M, Vijila T, Rajabhusanam C (2023) Tamil Speech Synthesizer App for Android: Text Processing Module Enhancement. Indian Journal of Science and Technology 16(7): 485-491. https://doi.org/10.17485/IJST/v16i7.2165

* **Corresponding author**.

rajabhushanamc.cse@bharathuniv.ac.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Designing dynamic computer systems that are effective, efficient, simple, and satisfying to use is becoming extremely important in this age of information and communication technology. Text to Speech or Speech Synthesis is one of the many methods being investigated by researchers to improve Human-Computer Interaction. The goal here is to improve the text processing component of the Tamil voice synthesizer by including a text normalizer and loan word identification that is efficient and reliable. **Methods:** Text normalization is conducted on unconstrained Tamil text to turn non-standard terms into common words to reduce confusing utterances during intermediate word processing. Loan/Native words in Tamil literature are detected to enhance the Tamil voice synthesizer system's pronunciation model. **Findings:** During normalization, non-standard Tamil words are replaced with standard ones to reduce ambiguous utterances during interim processing. A pronunciation model is built to improve the Tamil speech synthesizer system by identifying loan words in Tamil text. A syllable classifier is presented in this study, based on a decision list approach, which can handle various types of non-stationary sounds. **Novelty:** We also disclose a 'loan/native word classifier' based on multiple linear regressions that perform well even with small words of three syllables. Such sophisticated text processors are required in today's dominating Digital, Information-Communication Technology, and Human-Computer Interaction age.

Keywords: Mobile Communication Technology; HumanComputer Interaction; Speech Synthesis Affirms; Syllable Classifier; Prerecorded database

1 Introduction

This study aims to translate input Tamil sentences into equivalent spoken Tamil translations. This is to bring in conjunction with two major domains of NLP (Natural Language Processor), namely machine translation and Text-To-speech (TTS)[1]. Language interactions between computers and humans are part of the NLP, a branch of computer science that deals with NLP. Translating texts from one language to another is carried out using machine translation software. The blooming of Text-to-Speech (TTS) systems for Tamil languages has received extensive research[2,3]. An area of natural language processing stands at the point where languages and artificial intelligence meet. It focuses on the interplay between human and computer languages[4]. NLP is a subfield of human-computer interaction. An essential challenge in NLP is creating a medium through which computers can comprehend natural language and extract meaning. The field of computational linguistics is a subfield of natural language processing, focusing on language-related topics such as language modeling and representation[5].

Computer linguistics encompasses a variety of fields, including the development of a system for translating text translation; it was created as soon as written literature gained popularity. This was possible because the information in one language could easily be transmitted through translation to other languages and, therefore, to different cultures. The preponderance of literature has made it indispensable for modern translations to be automated, which would otherwise be tedious if done by hand[6]. The past decades have seen the development of various machine translation approaches. Universal Networking Language (UNL) has developed a system that can translate a Tamil sentence into Tamil speech without emotion. Prosody is added to the translated sentence for a humanized output voice using a Text-to-Speech system. Due to its final output of voice can be very useful for visually challenged or illiterate people[7].

2 Methodology

2.1 System Requirement

2.1.1 Hardware Requirements

The system should have at least 1 GB RAM Computer System (or) Android-based Mobile to run all the needed tools concurrently.

2.1.2 Software Requirements

- Operating System: Windows & Android
- Programming Languages
- Java and Android
- PHP, MySQL
- Tools
- Eclipse Luna
- XAMPP
- MATLAB

Applications only accessible on Windows and Linux domain analysis is necessary to clarify relationships when processing Tamil sentences that are complicated and compound. Android-based application for mobile application, using a syllable-based approach for complex and compound Tamil sentences[8].

An application that converts written text into a vocal version of the text is known as a Text to Speech scheme[9]. Pre-recorded database is used to produce the spoken voice by TTS. For the quality of the system, the following measures are used. There are two uses for speech intelligibility testing: one for evaluating hearing problems and the other for assessing speech synthesis and transmission[10].

NLP components carry out text analysis. An automatic transcription system converts the text into phonemes. A syllabification is necessary for words that don't exist in the database. When a text stream is tokenized, it is broken into words and symbols called tokens[11]. Each sentence contains many words that are separated. It is possible to segment written text using whitespace as the delimiter ("!", "?", "."). Tokenization involves fragmenting the text using whitespace. In Net Beans, Text entered in Tamil is accepted as a string[12]. Arial Unicode MS is set as the font type to accept Tamil strings. Words are tokenized using the delimiter in the input sentence.

Figure 1 shows color codes used to identify modules developed from scratch or modified/unmodified versions of existing standard modules. This system aims at translating a Tamil sentence into Tamil Speech and then voicing it out with emotions ensuring that there is no loss in the meaning conveyed and that the emotions are elegantly heard in the output[13]. The Morphological analyzer and POS Tagger rules are taken from Tamil grammar rules. They are developed from scratch and are designed to split each input word into root word, inflection, tense suffix, count suffix, and corresponding POS. The rules regarding UNL conversion are devised based on UNL specifications and Tamil Grammar. The conversion module is designed to generate a UNL graph from the input[14].

Synthesizing speech should be possible with the same quality as that produced by humans. When speech is natural, it approaches the human level of speech reliably[15]. The fundamental user interface for Text-to-Speech systems for Natural Languages is shown in Figure 1. Text is converted into speech output according to given instructions[16]. In mapping text to speech, two phases are usually involved - low-level and high-level synthesis[17]. Using the grammar and rules of the target language, text analysis is a high-level process in the Recent Trends in Information Technology that transforms the input text into a phonetic representation. Analyzing phonemes at the lowest level is called the phonetic phase[18]. An algorithm for unit selection speech synthesis matches the Phoneme with the prerecorded speech during phonetic analysis[19].

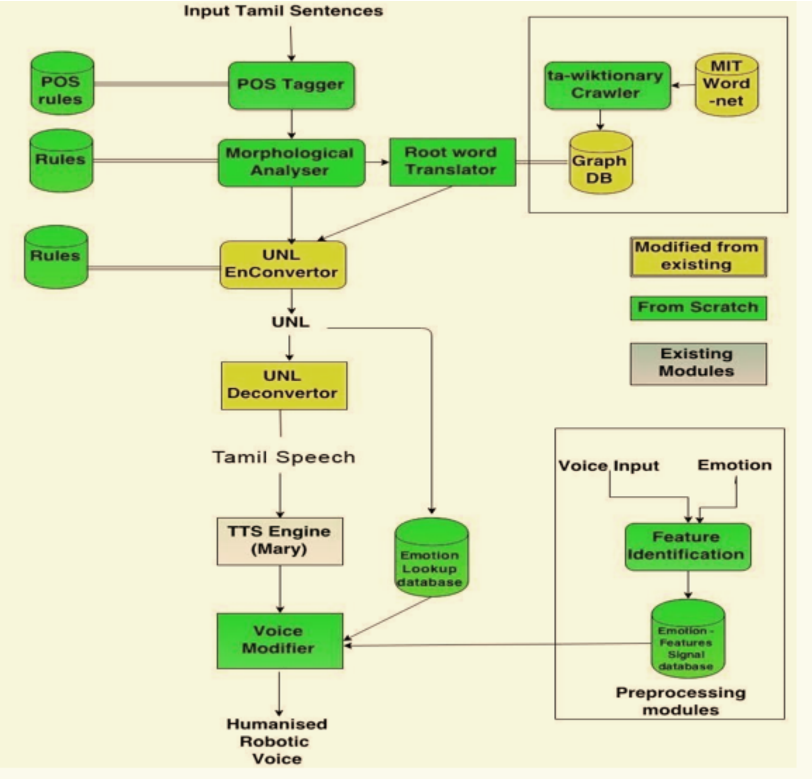

Fig 1. Block diagram for Tamil text to Tamil Emotional Speech Conversion Using UNL

3 Results and Discussion

Each feature identified in the voice samples has been analyzed and incorporated into this database. A total of three female speakers provided the voice samples[20,21]. Recording female speakers' voices is required since ”Mary” Text to Speech Synthesis needs a female voice as its output. Seven sentences are recorded in emotions such as angry, happy, sad, and neutral. The voices are recorded in a fairly quiet room to prevent any external noise from being recorded. Once the sentences have been read into Audacity, a software used for audio analysis, the word boundaries are differentiated, and the audio files for each word are divided. The noise removal feature of Audacity can be used to remove excess noise from an audio sample if excess noise has been recorded. Prosody and emotion are added to the text-to-speech system's output. The database of emotion features is consulted to determine the values for various parameters (such as fundamental frequency, intensity, duration, etc.). Then, to create the humanized voice, the robotic voice is altered based on the values of these parameters.

With its Java Speech API (JSAPI) release, Sun Microsystems completed The Java Speech API Markup Language (JSML), a specification for speech markup, which the company is developing. JSAPI defines an abstract Application Programming Interface (API) as a set of abstract classes and interfaces representing a Java programmer's view of a speech engine. A companion specification to JSAPI is JSML, which defines the Java Speech API Grammar Format (JSGF).

This system accepts ordinary (plain) text as input, as shown in Figure 2. The linguistic rules are an important aspect of text-to-speech synthesis is the conversion of printed or typed input into synthesized sounds. During the process of converting text

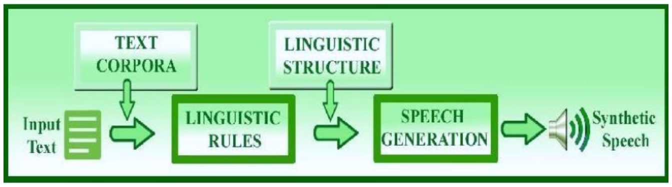

Fig 2. Text-to-Speech Conversion Techniques

into sounds, to send the words and intent of the text message in a way that human perception can interpret in real-time, a set of linguistic rules must be used to establish the proper collection of sounds (perhaps incorporating emphasis, pauses, rates of speaking, etc.). An efficient text-to-speech system relies heavily on the production of voice and text. This procedure provides a suitable order of phonemic units from the input text. By synthesizing parameters or selecting an entire speech from a large volume, the speech generation component generates these phonemic units. To build a computer system capable of speaking, the first thing aims to make a system capable of conveying a message, and the second thing seeks to make this message sound human. This is referred to as intelligibility and naturalness within the research community.

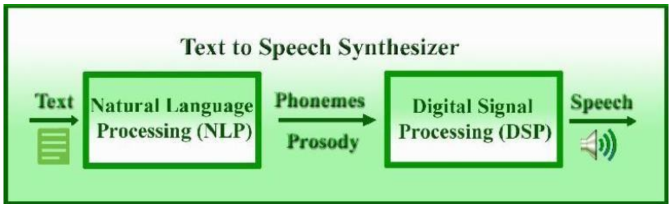
Text to Speech Synthesizer
Text
Natural Language Processing (NLP)
Phonemes
Prosody
Digital Signal Processing (DSP)
Speech

Fig 3. Text-to-Speech Synthesizer

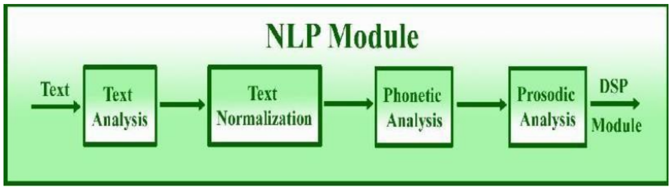

Fig 4. The NLP module

The science of Natural Language Processing (NLP) directly deals with human (natural) language processing. Computers or any other processing unit are the target devices used to accomplish such processing, which derives from computer science. According to NLP, this description corresponds to the ”Processing” particle. In contrast to other processing activities, NLP is unique in applying to human languages in Figure 3 above. Text processors need to support learning capabilities as they deal with more knowledge-related issues. There is a high level of linguistic dependability when it comes to transforming plain text into its linguistic representation and prosodic information. The change is relatively straightforward since some languages, like Indian languages, encode speech sounds orthographically.

The process of producing human speech artificially is known as speech synthesis. Typically, this process is carried out on a computer using a speech synthesizer or computer that can produce speech; this device may be hardware or software-based. Speech is produced from text using text-to-speech systems; phonetic transcriptions are rendered into speech using symbolic linguistic representations (Figure 4). A database can combine recorded pieces of speech to create synthesized speech. There are differences in the dimensions of speech units that are kept by the systems; the system with the greatest output range but the potential for poor clarity stores phones or diphones. High-quality output is possible when entire words or sentences are stored for specific usage domains. The vocal tract can also be modeled in a synthesizer to produce a truly ”synthetic” voice.

It is possible to measure the quality of a speech synthesizer by the degree to which it resembles the human voice and by the degree to which it is understandable. Listening to the act of writing on a computer is possible for an intelligible text-to-speech program to be helpful for disabled readers or people with visual impairments. It has been common for computer operating systems in the early 1990s to include speech synthesizers. An application that converts text into speech consists of a front and a backend. There are two major tasks that the front end performs. As the first step, symbols such as numbers and abbreviations must be converted into written words. It is commonly called normalization, preprocessing, or Tokenization of text. The second step divides the text into prosodic units (phrases, clauses, and sentences) and assigns a phonetic transcription. The assignment of phonetic ascription words resulted in conversion from text to phoneme or phoneme to text. The symbolic linguistic representation produced by content is composed of phonetic transcriptions and prosody information. A synthesizer, also known as the back end, is then used to translate the representation of language in sound through symbolic language. Depending on the target prosody (pitch contour and phoneme durations), this part's output speech is then imposed and computed.

A multilingual TTS system may read texts in more than one language and generate synthesized voices in that language. A polyglot TTS system, on the other hand, may transition between languages as needed, even when reading a single language's text input content. Language switching is a must for reading multilingual electronic materials in today's environment. A language identification model is required to execute this language change within a document. A Polyglot TTS system is a monolingual TTS system with the ability to transition between languages when it detects the presence of a loan word in the input text document[(22)]. The capacity of language identification and language switching is entirely reliant on the training of the language identification model on the individual languages for successful loan word identification[(23)].

Several computer approaches for language identification based on various characteristics have already been employed. These language recognition algorithms may be used with modest modifications to identify loan words. In the case of a TTS system, the text sequence is a word that might range from very few to numerous characters. As a result, the difficulty here is determining the optimal approach and optimizing the parameters for recognizing the borrowed word from brief text inputs[(24)].

Several machine learning methods have been investigated for this task, including Bayesian classification, Relative entropy-based classification, Decision trees, Neural Networks, Centroid based classification, Multiple Linear Regression Classification, Support Vector Machines, Letter Weighting Approach using Neural Networks, and Artificial Ants and K-means algorithms[(25)]. The Multiple Linear Regression (MLR) models has proven effective for identifying Indian languages.

The language identification approach was used in the current investigation to determine whether a borrowed word belongs to the Tamil language[(26)]. The procedure of determining the language of origin of the borrowing word still needs to be completed. The MLR model, used for language identification, has been tweaked and is now utilized for Tamil loan word recognition. The n-gram syllable-based language model is used to train an MLR model. As previously stated, English, Sanskrit, Hindi, and Telugu loanwords are more prevalent in Tamil[(27)].

A text processing module for a Tamil Text to Speech System was studied, as well as text normalization and loan word recognition. The 'semiotic classifier'-based decision list technique for text normalization can handle many types of NSW. However, there are notable exceptions. These outliers are attributable to the fascinating facts of the morphological richness of the Tamil language. It is difficult enough to verbalize an NSW by examining the semiotic class; if the NSW is a compound NSW, the work becomes even more difficult[(28)]. Furthermore, due to the extremely agglutinative and inflectional structure of the Tamil language, the processing becomes highly difficult, and the processing efficiency of the semiotic classifier is discerned. Next, with an average accuracy of 91%, the 'loan/native word classifier' based on multiple linear regression performs well even on shorter terms of 3 syllables[(29)]. The difficulty in categorizing loan words is related to inter-lingual homophones and homographs, which are difficult to identify from the text. The loan word classifier's performance is sensitive to changeable test samples and can be improved further if we can design a method to deal with inter-lingual homophones and homographs. The syllable feature set can be used as an introduction to the future modules of a TTS[(30)].

We can create a pronunciation model by mapping graphemes to syllables; work on segmenting syllables has previously been done to construct a syllable-based Tamil TTS. This comparable type of syllable feature set may be used to investigate further length and intonation modeling for boosting naturalness in a syllable-based TTS27. As a result, this syllable-level feature will excite the development of a thriving syllable-based Tamil TTS.

4 Conclusion

With UNL serving as the Interlingua, this technology translates Tamil text into the emotional voices of the Tamil text. The usage of an Interlingua is preferable for translation in the case of a morphologically complex language like Tamil. The Tamil text is processed using a specially created morphological analyzer to extract nouns, tenses, and gender. A POS Tagger was also developed to categorize the words into nouns, verbs, objects, adverbs, and adjectives. Based on POS tags and morphological structures, a UNL converter is created. The sentence's emotion is then determined from the UNL, and the TTS output for the translated Tamil Speech is adjusted to include prosody based on the determined emotion. The built program is trustworthy and user-friendly, and effective communication is carried out. This system could provide a solution to the issues that many people face in their busy lives, especially those who have low vision or reading disabilities, because it would enable them to listen to eBooks, study for examinations by listening to notes, and relax by listening to their emails while doing so.

Acknowledgment

Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India is gratefully acknowledged for providing the laboratory facilities to carry out the Project.

References

1) Joshi S, Bairag V. Recent trends in text to speech synthesis of Indian Languages. *Helix*. 2019;9:4931–4936. Available from: https://doi.org/10.29042/2019-4931-4936.
2) Tebbi H, Hamadouche M, Azzoune H. A new hybrid approach for speech synthesis: application to the Arabic language. *International Journal of Speech Technology*. 2019;22(3):629–637. Available from: https://doi.org/10.1007/s10772-018-9499-4.
3) Singh A, Kaur N, Kukreja V, Kadyan V, Kumar M. Computational intelligence in processing of speech acoustics: a survey. *Complex & Intelligent Systems*. 2022;8(3):2623–2661. Available from: https://doi.org/10.1007/s40747-022-00665-1.
4) Joshi MM, Agarwal S, Shaikh S, Pitale P. Text to speech synthesis for Hindi language using festival framework. *International Research Journal of Engineering and Technology*. 2019;6:630–632. Available from: https://www.irjet.net/archives/V6/i4/IRJET-V6I4142.pdf.
5) Rajendran V, Kumar GB. A Robust Syllable Centric Pronunciation Model for Tamil Text To Speech Synthesizer. *IETE Journal of Research*. 2019;65(5):601–612. Available from: https://doi.org/10.1080/03772063.2018.1452642.
6) Nadig PP, Pooja G, Kavya D, Chaithra R, Radhika AD. Survey on text-to-speech Kannada using Neural Networks. *International Journal of Advance Research*. 2019;5:128. Available from: https://www.ijariit.com/manuscripts/v5i6/V5I6-1159.pdf.
7) Kewley-Port D, Nearey TM. Speech synthesizer produced voices for disabled, including Stephen Hawking. *The Journal of the Acoustical Society of America*. 2020;148(1):1–2. Available from: https://doi.org/10.1121/10.0001490.
8) Koc WW, Chang YT, Yu JY, İk TU. Text-to-Speech with Model Compression on Edge Devices.. 2021. Available from: https://doi.org/10.23919/APNOMS52696.2021.9562651.
9) Manoharan S. A smart image processing algorithm for text recognition, information extraction and vocalization for the visually challenged. *Journal of Innovative Image Processing*. 2019;1(01):31–38. Available from: https://doi.org/10.36548/jiip.2019.1.004.
10) Balyan A. An Overview on Resources for Development of Hindi Speech Synthesis System. *New Ideas Concerning Science and Technology Vol 11*. 2021;16:57–63. Available from: https://doi.org/10.9734/bpi/nicst/v11/5977D.
11) Jayakumari J, Jalin AF. An Improved Text to Speech Technique for Tamil Language Using Hidden Markov Model. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. 2019;28:1–5. Available from: https://doi.org/10.1109/ICSCC.2019.8843683.
12) Herbert B, Wigley G, Ens B, Billinghurst M. Cognitive load considerations for Augmented Reality in network security training. *Computers & Graphics*. 2022;102:566–591. Available from: https://dx.doi.org/10.1016/j.cag.2021.09.001.
13) Gujarathi PV, Patil SR. Gaussian Filter-Based Speech Segmentation Algorithm for Gujarati Language. *Smart Computing Techniques and Applications*. 2021;2021:747–756. Available from: https://doi.org/10.1007/978-981-16-1502-3_74.
14) Daba E. Improving Afaan Oromo Question Answering System: Definition, List and Description Question Types for Non-factoid Questions. St. Mary's University. 2022. Available from: http://repository.smuc.edu.et/handle/123456789/6240.
15) Kim C, Gowda D, Lee D, Kim J, Kumar A, Kim S, et al. A Review of On-Device Fully Neural End-to-End Automatic Speech Recognition Algorithms. *2020 54th Asilomar Conference on Signals, Systems, and Computers*. 2020;1:277–283. Available from: https://doi.org/10.1109/IEEECONF51394.2020.9443456.
16) Kodhai SD. Textaloud Assistant App Development for Multilanguage. *International Journal of Innovative Technology and Exploring Engineering*. 2019;8:1–5. Available from: https://www.ijitee.org/wp-content/uploads/papers/v8i7s/G10010587S19.pdf.
17) Narvani V, Arolkar H. Information and Communication Technology for Competitive Strategies. *Lecture Notes in Networks and Systems*. 2021;190. Available from: https://doi.org/10.1007/978-981-16-0882-7_84.
18) Madhfar MAH, Qamar AM. Effective Deep Learning Models for Automatic Diacritization of Arabic Text. *IEEE Access*. 2021;9:273–288. Available from: https://dx.doi.org/10.1109/access.2020.3041676.
19) Changrampadi MH, Shahina A, Narayanan MB, Khan AN. End-to-End Speech Recognition of Tamil Language. 2022. Available from: https://doi.org/10.32604/iasc.2022.022021.
20) Araya M, Alehegn M. Text to Speech Synthesizer for Tigrigna Linguistic using Concatenative Based approach with LSTM model. *Indian Journal of Science and Technology*. 2022;15(1):19–27. Available from: https://doi.org/10.17485/IJST/v15i1.1935.
21) Divyasri K, Gayathri GL, Swaminathan K, Durairaj T, Bharathi B. PANDAS@ TamilNLP-ACL2022: Emotion Analysis in Tamil Text using Language Agnostic Embeddings. *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*;2022:105–111. Available from: https://doi.org/10.18653/v1/2022.dravidianlangtech-1.17.
22) Gamallo P, Pichel JR, Alegria I. From language identification to language distance. *Physica A: Statistical Mechanics and its Applications*. 2017;484:152–162. Available from: https://doi.org/10.1016/j.physa.2017.05.011.
23) Romsdorfer H, Pfister B. Text analysis and language identification for polyglot text-to-speech synthesis. *Speech Communication*. 2007;49(9):697–724. Available from: https://doi.org/10.1016/j.specom.2007.04.006.
24) Hakkinen J, Tian J. n-gram and decision tree based language identification for written words. *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001 ASRU '01*. 2001;9:335–338. Available from: https://doi.org/10.1109/ASRU.2001.1034655.
25) Tian J, Suontausta J. Scalable neural network based language identification from written text. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 Proceedings (ICASSP '03)*. 2003;1:1–48. Available from: https://doi.org/10.1109/ICASSP.2003.1198713.
26) Kruengkrai C, Srichaivattana P, Sornlertlamvanich V, Isahara H. Language identification based on string kernels. *IEEE International Symposium on Communications and Information Technology, 2005 ISCIT 2005*. 2005;2:926–929. Available from: https://doi.org/10.1109/ISCIT.2005.1567018.
27) Murthy KN, Kumar GB. Language identification from small text samples*. *Journal of Quantitative Linguistics*. 2006;13(1):57–80. Available from: https://doi.org/10.1080/09296170500500694.

28) Bhargava A, Kondrak G. Language identification of names with SVMs. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2010;p. 693–696. Available from: https://aclanthology.org/N10-1102.pdf.

29) Ng CCC, Selamat A. Improved Letter Weighting Feature Selection on Arabic Script Language Identification. *2009 First Asian Conference on Intelligent Information and Database Systems*. 2009;p. 150–154. Available from: https://doi.org/10.1109/ACIIDS.2009.33.

30) Amine A, Elberrichi Z, Simonet M. Automatic Language Identification: An Alternative Unsupervised Approach Using a New Hybrid Algorithm. *International Journal Computer Science Applications*. 2010;7:94–107. Available from: https://www.researchgate.net/publication/42387505.