Partial Greedy Algorithm to Extract a Minimum Phonetically-and-Prosodically Rich Sentence Set

Fahmi Alfiansyah[1]
School of Computing, Telkom University
Jl. Telekomunikasi No. 01, Terusan Buah Batu
Bandung, West Java, Indonesia 40257

Suyanto[2]
School of Computing, Telkom University
Jl. Telekomunikasi No. 01, Terusan Buah Batu
Bandung, West Java, Indonesia 40257

***Abstract*—A phonetically-and-prosodically rich sentence set is so important in collecting a read-speech corpus for developing phoneme-based speech recognition. The sentence set is usually searched from a huge text corpus of million sentences using the optimization methods. One of the commonly used optimization methods for this case is a Least-to-Most Greedy (LTMG) algorithm. It is effective in minimizing the number of phoneme-units. Unfortunately, it does not distribute their frequencies. In this paper, a new method called Partial LTMG algorithm (PLTMG) is proposed to search an optimum set containing triphones and prosodies those are distributed in a near-uniform fashion. Testing on an Indonesian text corpus of ten million sentences crawled from some websites of newspapers and novels shows that the proposed method is not only capable of minimizing both phoneme-units and prosodies but also effective in distributing their frequencies.**

Keywords*—Automatic speech recognition; minimum sentence set; prosody; speech corpus; triphone

I. Introduction

Before 2014, an Automatic Speech Recognition (ASR) or Computer Speech Recognition generally has three components, i.e. acoustic model, pronunciation or word lexical model, and language model. Most ASR systems use the statistical approaches with supervised learning. The Hidden Markov Model is a commonly used method to train both acoustic and language models.

In 2014, many researchers start to develop an End-to-End Automatic Speech Recognition (E2EASR), which trains those three components of ASR in a single model [1], [2], [3], and [4]. The E2EASR does not need both pronunciation and language models commonly used in the previous conventional ASR. Hence, it can be embedded into a microdevice since it does not consume a high memory.

The first effort to build an E2EASR system is conducted by some researchers in 2014 using a classification-based approach called Connectionist Temporal Classification (CTC) (Graves 2014). This system consists of a layer of CTC and a Recurrent Neural Networks (RNN), which is abbreviated CTC-RNN. This system learns two components of ASR, pronunciation and acoustic models. However, this system gives many spelling mistakes so that it needs an external language model separately. In [2], some researchers at Baidu Research build the Deep Speech 2, an E2EASR that is successfully applied to English and Mandarin in 2015.

In 2016, some researchers from CMU, Google Brain, and University of Montreal propose an attention-based ASR model. The model is called "Listen, Attend, and Spell" (LAS) [5], [6], and [7]. Unlike the CTC-based ASR, this LAS model is capable of learning all ASR components (acoustic models, pronunciation models, and language models) simultaneously. Hence, this LAS is the first fully E2EASR model with no external language model. In [8], the researchers consider that the LAS system is a more successful model than the CTC-based systems.

In 2017, the researchers from CMU, MIT, and Google Brain develop Latent Sequence Decompositions (LSD) that directly outputs the sub-word units, which are not only wider but also more natural than characters [9]. In early 2018, the researchers from Johns Hopkins University, Baltimore, USA, develop a new architecture called multi-modal data augmentation network (MMDA) that supports multi-modal inputs (acoustic and symbolic). The MMDA seeks to avoid the use of external language models with a much smaller combined text corpus and speech corpus to train the E2EASR [10]. Hence, a well-designed speech corpus is very important to train a high-performance E2EASR.

Many methods have been proposed to design a speech corpus, such as described in [11], [12], [13], [14]. The methods generally use a sub-word unit called triphone, i.e. a sequence of three contextual phonemes. A triphone is commonly written using a format L-X+R, where X is a target, L and R are a prefix and a postfix of the target respectively. Three samples of converting different types of sentences in Bahasa Indonesia (declarative, interrogative, and imperative/exclamatory) into cross-word triphone forms are listed in Table I: "Aku pergi." ("I go.") pronounced as /ku p@rgi./, "Apa kabar?" ("How are you?") pronounced as /p kbr?/, and "Ambil itu!" ("Take it!") pronounced as /mbi itu!/, where /sil/ is a silence. All Indonesian phonetic symbols described in [15], which are based on the International Phonetic Alphabet (IPA), are adopted in this paper.

TABLE I. Conversion of Three Types of Sentences into Triphone Forms, where /sil/ is a Silence

Sentences	Triphone Forms
Aku pergi.	sil-+k -k+u k-u+p u-p+@ p-@+r e-r+g r-g+i g-i+. i-.+sil
Apa kabar?	sil-+p -p+ p-+k -k+ k-+b -b+ b-+r -r+? r-?+sil
Ambil itu!	sil-+m -m+b m-b+i b-i+ i-+i -i+t i-t+u t-u+! u-!+sil

Developing a read-speech corpus needs a well-designed text of transcription to be read by hundreds or even thousands

of varying speakers based on their ages, accents, dialects, and genders [14], [16], [17]. The text of transcription is commonly a minimum phonetically-and-prosodically rich sentence set searched from a huge text corpus. Why prosody? The prosody affects how a speech sentence is being interpreted [18], [13]. For example, two sentences "This is mine." and "This is mine?" have different prosodies (intonations) and consequently have different interpretations.

One of the effective optimization methods to find an optimum set is a Modified LTMG (MLTMG) that is proposed in [19]. Unfortunately, this algorithm just minimizes the phoneme-units but does not care to balance their frequencies. Hence, in this paper, a new method called Partial LTM Greedy algorithm (PLTMG) is proposed to search a phonetically-and-prosodically rich sentence set with balanced frequencies from an Indonesian text corpus.

II. Related Work

A speech corpus can be generally developed using either a phonetically-balanced or a phonetically-rich text corpus [20]. A phonetically-balanced corpus is a sentence set that follows Zipfian's law, where each triphone is represented proportionally to its frequency. This corpus is not good enough to build an ASR, a speech synthesizer, or a pronunciation quality assessment. In contrast, a phonetically-rich text corpus that is a uniform triphone representation gives more accurate results for those tasks. It has a high variety of triphones in a sentence that uniformly distributed regardless their appearances in a language.

Many optimization methods have been proposed to develop a phonetically-rich text corpus. They are commonly based on either a greedy approach, such as described in [21] and [22], or an evolutionary computation as described in [23]. However, the greedy-based approach is more widely used in practice since it provides a much faster processing time as well as a higher scalability.

In [21], the researchers show that an LTMG is capable of extracting smaller sentence sets and fewer computation costs than the other standard greedy algorithms. But, this algorithm has two problems. Firstly, it just selects a to-be-covered unit randomly when there are some units have the same frequencies. Secondly, it may produce redundant sentences as the covering score is computed based on a set of to-be-covered units updated by the previous selection. In [19], the researcher proposes an MLTMG to solve both problems by 1) collecting all sentences those contain the same frequencies to-be-covered units into a subset, then select the best one from it and 2) evaluating each sentence in the extracted minimum-so-far set to check its redundancy.

Unfortunately, the MLTMG also has two drawbacks. When some sentences containing a to-be-covered unit with the same scores, it cannot choose the best one. When some sentences have the same scores but different to-be-covered units or frequencies, it just randomly selects a sentence without other calculation nor consideration. The MLTMG extracts a optimum set by sequentially selecting the best sentences in a greedy way based on a ratio-based scoring formula. Besides, as explained in [19], it is just evaluated using a relatively small motherset of 500 k sentences without considering any prosody. Hence, in this research, the MLTMG is improved by proposing some new procedures to handle a much bigger motherset of 10 M sentences with considering the prosodies.

III. Proposed Partial LTM Greedy

Here, the MLTMG is improved by taking into account the number of to-be-covered triphones as well as their frequencies before selecting a sentence so that this method is called a PLTMG. This new method is simply implemented by replacing the step 5 in the MLTMG in [19] with four new steps to become:

1) Let A be a mother sentence set, B be an empty set, and U be a list of all to-be-covered unique triphone tokens sorted by their frequency in ascending order;
2) Select all infrequent triphones (those have the least frequency) from U and then store them in a subset U_{sub};
3) Select all sentences from A those contain at least one triphone in U_{sub} and put them in a subset A_{sub};
4) For each sentence in A_{sub} calculate its score using a formula in Eq. (1)

$$S_i = \frac{V_i}{T_i}, \tag{1}$$

where V_i and T_i are the number of to-be-covered triphones and the total triphone tokens in the *i*th sentence respectively;
5) Sort the scores of sentences in ascending order;
6) Define P, a small number between 0 and 1, that states a percentage of sentence scores selected to compete;
7) Take the top P percent of sentences having scores bigger than $(1 - P) \times thebestscore$ and then store them in a subset C;
8) From C select a sentence with the highest score. If there are two or more sentences with the same highest scores then select one containing the most to-be-covered triphones. If there are two or more sentences having the most to-be-covered triphones then choose a sentence containing the least frequent triphones in B. Delete all triphones appear in the selected sentence from both U and U_{sub}. Remove all sentences from C.
9) Do step 3 to 8 until U_{sub} is empty;
10) Do step 2 to 9 until U is empty.

Step 8 in the proposed PLTMG can be easily explained using two illustrations in Fig. 1 and Fig. 2. The Fig. 1 illustrates a case where there are three sentences with the same highest scores of 1.00, i.e. the sentence index of 7995, 577, and 10000000. Since the 10000000th sentence has the maximum number of to-be-covered triphones of 40 and B is initially empty, the sentence is selected as the best one.

Meanwhile, Fig. 2 illustrates a case where there are two sentences with the same highest scores of 1.00 as well as the highest number of to-be-covered triphones of 27. In this case, let the 7995th sentence contains "***Pergi*** *jauh dariku, katanya*." (Go away from me, he said) and the 577th sentence consists of "*Kami mendapatkan ijazahnya*!" (We get the certificate!). Since B contains a sentence "*Sudah lama ia tidak* ***pergi*** *ke rumah mertua di desa*." (For a long time he does not go to

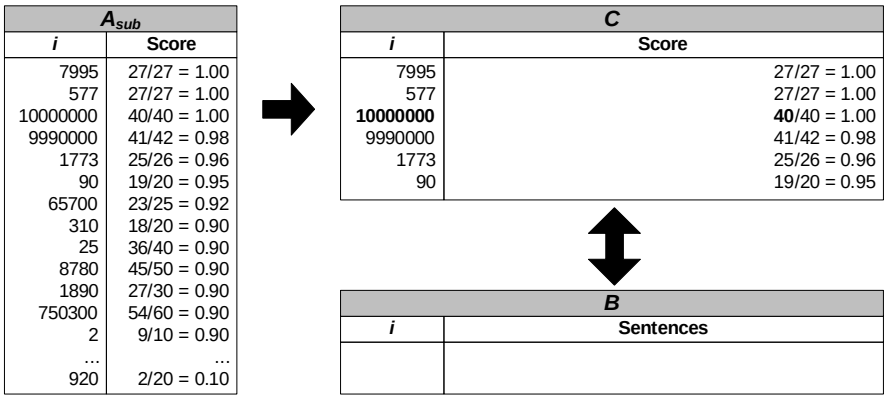

Fig. 1. PLTMG for a case where there are two or more sentences with the same highest scores but there is only one sentence has the highest number of to-be-covered triphones

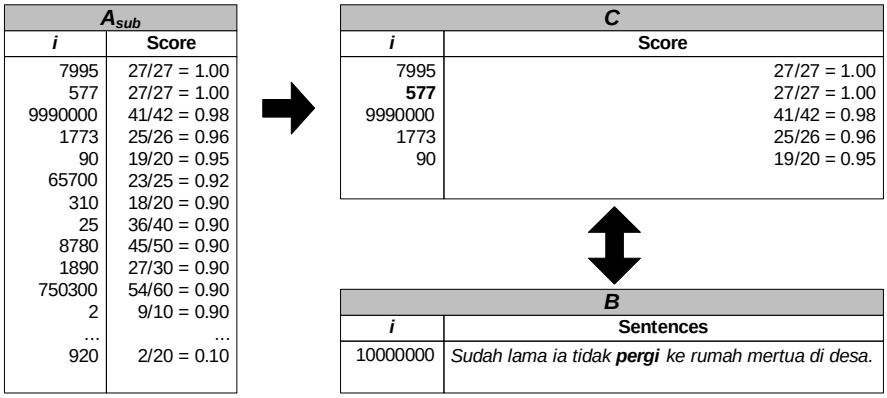

Fig. 2. PLTMG for a case where there are two or more sentences with the same highest scores as well as highest number of to-be-covered triphones

the home of his father in law at the village.) with a word "*pergi*", then the 7995th sentence is not selected. Instead, the PLTMG chooses the 577th sentence as it has the least frequent triphones covered in B.

Using those steps, the PLTMG should produce a sentence set containing slightly more to-be-covered triphones but lower frequencies than the MLTMG. Besides, this algorithm is also expected to be capable of avoiding some sentences with the same scores in a competition so that the random selection used in the MLTMG can be removed.

IV. Result and Discussion

The text corpus used here is a set of 10 M sentences that is collected by crawling some Indonesian websites of news and novels as describes in [24]. The corpus covers three types of sentence: declarative (ended by "."), interrogative (ended by "?"), and imperative/exclamatory (ended by "!"). Based on the corpus, a phonemic dictionary of 128,779 words is generated by an automatic Indonesian G2P system described in [15]. Phonetizing each sentence in the mother set using the dictionary, and then converting the phonemic sequences into triphones, produce 289,096,873 triphone tokens and 18,909 unique triphones as listed in Table II. It means the ratio of unique triphone and the tokens is very low, only 0.000065.

Some experiments are performed using a personal computer of an i7 processor and 4 GB RAM to get the runtime

TABLE II. Statistics of the Mother Sentence Set

Total number of sentences appear	10,000,643
Number of declarative sentences	9,938,093
Number of interrogative sentences	50,314
Number of imperative/exclamatory sentences	12,236
Total number of words appear	47,590,317
Number of distinct words	128,779
Number of triphone tokens	289,096,873
Number of unique triphones	18,909
Average number of triphones per sentence	28.91

of 5 hours per experiment. In the PLTMG, the variable P functions to select some sentences to compete. For example, if the best to-be-covered unit score on the iteration is 1 and P = 0.05 then the minimum score to compete will be 0.95. The PLTMG is tested using P = 0.05, 0.1, and 0.2 to see its behavior in extracting the mother set.

The experimental results in Table III proves that the proposed PLTMG is effective to decrease the standard deviation of triphone frequencies, where the standard deviation decreases by around 0.34 on each specified P. However, the number of triphones are higher than those produced by the Modified LTM Greedy. The PLTMG with P = 0.20 produces much more triphone tokens (up to 170,108) than the MLTMG. The PLTMG with P = 0.10 reaches an optimum sentence set. It produces a lower standard deviation than the PLTMG with P = 0.05 and fewer triphone tokens than the PLTMG with P = 0.20. In addition, the triphone frequencies on the PLTMG decrease

TABLE III. STATISTICS OF THE OPTIMUM SENTENCE SETS EXTRACTED BY MLTMG AND PLTMG ALGORITHMS

Algorithm	#triphones	#sentences	Avg. triph. freq.	Std. triph. freq.
MLTMG	165,673	7,334	8.76	30.42
PLTMG, $P = 0.05$	166,527	7,286	8.80	30.08
PLTMG, $P = 0.10$	167,604	7,263	8.86	29.74
PLTMG, $P = 0.20$	170,108	7,206	8.99	29.39

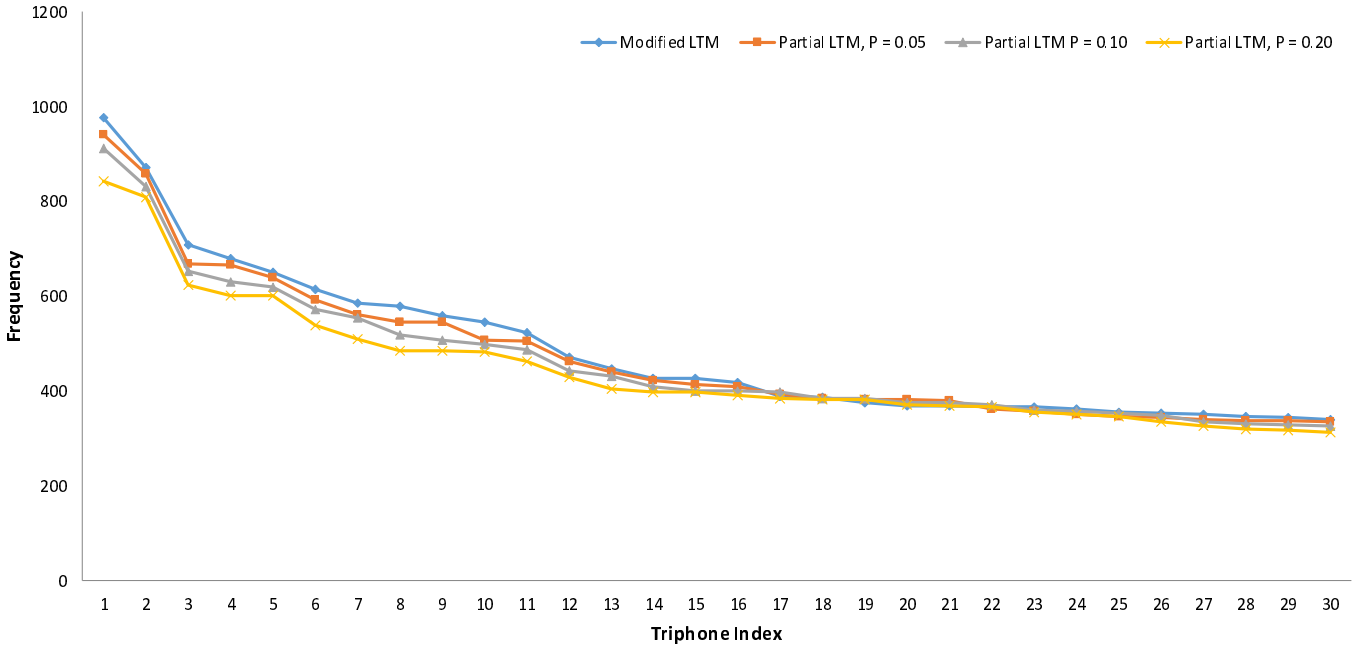

Fig. 3. The first thirty most-frequent triphones

as illustrated in Fig. 3. It shows that there are differences in frequencies at the beginning of the largest triphones since the PLTMG takes into account the number of triphone frequencies in a sentence to find the lowest frequency in the sentence.

V. CONCLUSION

The proposed PLTMG is effective to produce a sentence set that contains more uniformly distributed triphones than the previous MLTMG. The value of P affects the number of triphones as well as their standard deviations. The greater P the lower standard deviation. Unfortunately, the bigger P the more triphones selected. However, the PLTMG enables a user to make any adjustment to get the optimum extracted sentence set. In the future, the user can also apply the PLTMG to a much bigger motherset of hundreds of millions or even billions of sentences to get much more unique triphones.

ACKNOWLEDGMENT

We would like to thank all colleagues in the School of Computing, Telkom University, for the great support and suggestions.

REFERENCES

[1] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," pp. 1–12, 2014. [Online]. Available: http://arxiv.org/abs/1412.5567

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," pp. 1–28, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[3] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-Free Conversational Speech Recognition with Neural Networks," *Proceedings the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015. [Online]. Available: http://deeplearning.stanford.edu/lexfree/

[4] A. L. Maas, P. Qi, Z. Xie, A. Y. Hannun, C. T. Lengerich, D. Jurafsky, and A. Y. Ng, "Building DNN acoustic models for large vocabulary speech recognition," *Computer Speech & Language*, vol. 41, pp. 195–213, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230816301930

[5] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, Attend and Spell," pp. 1–16, 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4945–4949.

[8] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," 2016.

[9] W. Chan and Y. Zhang, "Latent Sequence Decompositions," pp. 1–12, 2017.

[10] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-Modal Data Augmentation for End-to-end ASR," in *Interspeech*, 2018. [Online]. Available: http://arxiv.org/abs/1803.10299

[11] M. Pinnis, A. Salimbajevs, and I. Auziņa, "Designing a Speech Corpus for the Development and Evaluation of Dictation Systems in Latvian," in *The Tenth International Conference on Language Resources and Evaluation (LREC)*, 2016, pp. 775–780.

[12] D. Koržinek, K. Marasek, Ł. Brocki, and K. Wołk, "Polish Read Speech Corpus for Speech Tools and Services," in *CLARIN*, 2017, pp. 54–62.

[13] S. M. Hosseini and H. Sameti, "Creating a corpus for automatic punctuation prediction in Persian texts," in *2017 Iranian Conference on Electrical Engineering (ICEE)*, may 2017, pp. 1537–1542.

[14] H. Abera and S. H/Mariam, "Design of a Tigrinya Language Speech Corpus for Speech Recognition," in *Workshop on Linguistic Resources for Natural Language Processing*, vol. 9, 2018, pp. 78–82.

[15] S. Suyanto, S. Hartati, and A. Harjoko, "Modified Grapheme Encoding and Phonemic Rule to Improve PNNR-Based Indonesian G2P," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 3, pp. 430–435, 2016.

[16] C. Kurian, "Development of Speech corpora for different Speech Recognition tasks in Malayalam language," in *International Conference on Natural Language Processing*, no. December, 2015, pp. 229–236.

[17] D. Arnold, F. Tomaschek, K. Sering, F. Lopez, and R. H. Baayen, "Words from spontaneous conversational speech can be recognized with human-like accuracy by an error-driven learning algorithm that discriminates between meanings straight from smart acoustic features, bypassing the phoneme as recognition unit," *PLOS ONE*, vol. 12, no. 4, pp. 1–16, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174623

[18] N. Moore, "What ' s the point ? The role of punctuation in realising information structure in written English," *Functional Linguistics*, 2016. [Online]. Available: http://dx.doi.org/10.1186/s40554-016-0029-x

[19] Suyanto, "Modified Least-to-Most Greedy Algorithm to Search a Minimum Sentence Set," in *IEEE TENCON*, 2006.

[20] G. Mendonça, S. Candeias, F. Perdigão, C. Shulby, R. Toniazzo, A. Klautau, and S. Aluísio, "A method for the extraction of phonetically-rich triphone sentences," in *2014 International Telecommunications Symposium (ITS)*, 2014, pp. 1–5.

[21] J.-s. Zhang and S. Nakamura, "An Efficient Algorithm to Search For A Minimum Sentence Set For Collecting Speech Database," in *ICPhS*, 2003, pp. 3145–3148.

[22] K. Arora, S. Arora, K. Verma, and S. S. Agrawal, "Automatic Extraction of Phonetically Rich Sentences from Large Text Corpus of Indian Languages," in *INTERSPEECH*, 2004.

[23] M. Nicodem, I. Seara, R. Seara, D. Anjos, and R. Seara-Jr, "Selecao automatica de corpus de texto para sistemas de sıntese de fala," in *XXV Simposio Brasileiro de Telecomunicacoes (SBrT)*, 2007.

[24] B. Nugroho and B. Nurtomo, "Greedy Algorithms to Optimize a Sentence Set Near-Uniformly Distributed on Syllable Units and Punctuation Marks," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 10, pp. 291–296, 2018.