Evidence of correspondence

Data Augmentation Methods for Low-Resource Orthographic Syllabification

1. **First submission (19 July 2020)**
2. LoA with Minor Revision (06 Augustus 2020)
3. Responses to Reviewers, Final submission (08 August 2020)
4. Final paper of source latex submitted (08 August 2020)
5. Creative Commons Attribution License (08 August 2020)

Telkom University

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

IEEE Access - Manuscript ID Access-2020-35994

IEEE Access <onbehalfof@manuscriptcentral.com> Sun, Jul 19, 2020 at 9:32 PM
Reply-To: ieeeaccess2@ieee.org
To: suyanto@telkomuniversity.ac.id, kemas.muslim@gmail.com, arifbijaksana@telkomuniversity.ac.id, adriana@student.telkomuniversity.ac.id

19-Jul-2020

Dear Dr. Suyanto:

Your manuscript entitled "Data Augmentation Methods for Low-Resource Orthographic Syllabification" has been successfully submitted online and is presently being given full consideration for publication in IEEE Access.

As noted during the submission of your manuscript, IEEE Access is a fully open access journal. Open Access provides unrestricted access to peer-reviewed articles via IEEE Xplore. In lieu of paid subscriptions, authors are required to pay an article processing charge of $1,750 after the article has been accepted for publication.

Your manuscript ID is Access-2020-35994. Please mention the manuscript ID in all future correspondence to the IEEE Access Editorial Office. You can also view the status of your manuscript at any time by checking your Author Center after logging in to https://mc.manuscriptcentral.com/ieee-access.

At this time, we kindly request your assistance in helping us improve IEEE Access by taking this QUICK 4-QUESTION SURVEY in the following link: https://research.ieee.org/jfe/form/SV_7R63xmcN4etQVEx

Thank you again for submitting your manuscript to IEEE Access.

Sincerely,

IEEE Access Editorial Office

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Data Augmentation Methods for Low-Resource Orthographic Syllabification

SUYANTO SUYANTO, (Member, IEEE), KEMAS M. LHAKSMANA, (Member, IEEE), MOCH ARIF BIJAKSANA, AND ADRIANA KURNIAWAN

School of Computing, Telkom University, Bandung, Indonesia

Corresponding author: Suyanto Suyanto (e-mail:suyanto@telkomuniversity.ac.id, Orcid: https://orcid.org/0000-0002-8897-8091).

ABSTRACT An n-gram syllabification model generally produces high error rate for a low-resource language, such as Indonesian, because of the high rate of out-of-vocabulary (OOV) n-grams. In this paper, a combination of three methods of data augmentations are proposed to solve the problem, namely swapping consonant-graphemes, flipping onsets, and transposing nuclei. An investigation on 50k Indonesian words shows that combination of the three data augmentation methods drastically increases the amount of both unigrams and bigrams. A previous procedure of flipping onsets has been proven to enhance the standard bigram-syllabification by relatively decreasing the syllable error rate (SER) by up to 18.02%. Meanwhile, the previous swapping consonant-graphemes has been proven to give a relative decrement of SER up to 31.39%. In this research, a new transposing nuclei-based augmentation method is proposed and combined with both flipping and swapping procedures to tackle the drawback of bigram syllabification in handling the OOV bigrams. An evaluation based on k-fold cross-validation (k-FCV), using $k = 5$, for 50 thousand Indonesian formal words concludes that the proposed combination of the three procedures relatively decreases the mean SER produced by the standard bigram model by up to 37.63%. The proposed model is comparable to the fuzzy k-nearest neighbour in every class (FkNNC)-based model. It is worse than the state-of-the-art model, which is developed using combination of bidirectional long short-term memory (BiLSTM), convolutional neural networks (CNN), and conditional random fields (CRF), but it offers a low complexity.

INDEX TERMS Indonesian, bigram, flipping onsets, orthographic syllabification, swapping consonant-graphemes, transposing nuclei

I. INTRODUCTION

A syllabification can be defined as a splitting a word into syllables automatically. It is important not just in some researches but also in many linguistics-based applications. It is generally used in speech recognition [1] [2], speech synthesis [3] [4] [5], emotion classification [6] [7], speaker's dialect identification [8], speaking rate estimation [9], speaking proficiency scoring [10], word count estimation [11], phonemicization [12] [13], collecting a minimum sentence set in developing speech corpus, as described in [14] [15] [16], etc.

The syllabification is preferably applied to graphemes than phonemes because of both simplicity and flexibility. Although a graphemic (or orthographic) syllabification generally gives lower accuracy [17] than the phonemic one [18], it can be easily applied to both unseen words and named-entities that have so many exceptions and ambiguities.

Most researchers prefer a statistical-based syllabification much more than the rule-based one as it is simpler and accurate [19]. For example, a simple Näive Bayes produces quite low SER of around 12.90% for the Romanian language [20]. Some other statistical models use decision tree [20] [21], treebank [22], random forest [20], neural network [23] [24] [25] [26], support vector machine [20] [27], finite-state transducers [28] [29], context-free grammars [30], hidden Markov model [31], syllabification by analogy [19], dropped-and-matched model [32], n-gram [33], conditional random fields [34] [35], nearest neighbour [17], and unsupervised model [36].

The neural language models proposed in [37] [38] give excellent results. A recent model based on BiLSTM-CNN-

CRF gives the state-of-the-art result [39]. But, some *n*-gram models produce comparable accuracies as well as offer simplicity and fast processing [40] [41] [42].

In [33], the researchers prove that *n*-gram syllabification, which is one of the simplest models, produces a low word error rate (WER) of 0.15% for the phonemic sequences of the Germany language. It can be generalized into any language since it does not need any knowledge of a particular language. However, the *n*-gram model is generally poor for a tiny dataset producing many OOV *n*-grams. In [43], the researcher proposes a simple combined standard bigram and flipping onsets model (BFO) to tackle the OOV problem. Compare to the standard bigram models, the simple procedure can relatively reduce the SER by 18.02%. However, its performance is not stable for a tiny dataset.

In [44], the researcher proposes a simple backoff smoothing procedure called swapping phonological similarities (CBSPS) model, which can boost a bigram-based orthographic syllabification. It also performs better than the BFO model. In this research, a new model called a combination of flipping-onsets with standard-trigram and augmented-bigram syllabification (CFTABS) is proposed to solve the problem. Standard trigram is expected to perform better than bigram. Three augmentation methods of swapping consonant-graphemes, flipping onsets, and transposing nuclei are proposed to reduce the OOV rate. Next, CFTABS is evaluated and compared to BiLSTM-CNN-CRF [39] using 50 k Indonesian words based on 5-fold cross-validation scheme.

II. PRELIMINARY STUDY ON INDONESIAN

For many languages, those three methods of data augmentations commonly create many illegal syllables for both unigrams and bigrams. But, for other simpler languages with a nearly one-to-one grapheme-to-phoneme mapping, such as Indonesian, they generate many legal ones. Swapping several consonant-graphemes in a word for all combinations may generate some new words. It is performed by replacing them with other similar ones in the same manner and/or place of articulations. For example, swapping consonant-graphemes in an original formal word "***b**e.**r**as*" (rice) generates three other words: "***b**e.**l**as*" (mercy), "***p**e.**r**as*" (squeeze), and "***p**e.**l**as*" (pity) with no shifting the syllabification boundaries as both ⟨b⟩ and ⟨p⟩ are pronounced as the plosive-bilabial phonemes while ⟨r⟩ and ⟨l⟩ are pronounced as the trill and lateral-dental phonemes. Flipping two onsets in the original word yields another word "***r**e.**b**as*" (boil) without shifting the syllabification point. Flipping two onsets in the three consonants-swapped words produces three other words: "***l**e.**b**as*" (too ripe), "***r**e.**p**as*" (fragile), and "***l**e.**p**as*" (free). Transposing two nuclei in the original word produces another new word, "*b**a**.r**e**s*" (OOV). Transposing two nuclei in all consonants-swapped and onsets-flipped words produces seven new words: "***b**a.**l**es*" (reply), "***p**a.**r**es*" (OOV), "***p**a.**l**es*" (discordant), "***r**e.**b**as*" (OOV), "*l**a**.b**e**s*" (OOV), "*r**a**.p**e**s*" (OOV), and "*l**a**.p**e**s*" (OOV). Thus, in this case, the three methods augment a short original formal word "***b**e.**r**as*" (rice) into nine new formal words and six OOV words.

Furthermore, the Indonesian language has 18 prefixes [45]. Swapping some graphemes in the prefixes produce much more other legal suffixes than the illegal ones called noises, as described in [44]. A preliminary study shows that the dataset of 50 k Indonesian words produces up to 161,981 unigrams. Applying three augmentation procedures to the dataset generates up to 9,620,054 augmented unigrams (87.20% are legal). The 50 k words generate a total of 111,412 bigrams. The augmentation procedures yield 7,308,702 augmented bigrams (77.26% are legal). Finally, the 50 k words generate a total of 136,812 syllable trigrams. In this research, the augmented syllable trigrams are not generated since they produce a high sparsity of trigrams and consequently they are not effective in the syllabification process. Based on a particular criterion, such as phonotactic rule, those grams can be classified into two classes: legal and illegal. However, it is quite hard to recognize them as legal or illegal. Therefore, CFTABS are designed to use all generated syllable unigrams and bigrams without filtering to focus this research on examining how much the proposed CFTABS decreases the SER.

III. RESEARCH METHOD

The training process of the proposed model can be easily explained as the combination of normal (or standard) and swapped syllabifications. A dataset containing pairs of words and syllabification points is scanned to create two lists of normal syllable unigrams and bigrams as well as two lists of augmented syllable unigrams and bigrams, which is illustrated in Fig. 1.

Next, both normal and swapped unigrams as well as bigrams are used to test the model, which is illustrated in Fig. 2, to maximize the final score to produce the best syllable sequence as the output. For example, let a given grapheme sequence is ⟨*beras*⟩ (rice). First, two vowel-graphemes ⟨***e***⟩ and ⟨***a***⟩ are searched and their positions are listed as {2, 4}. Next, two candidates syllabifications are then generated, i.e. ⟨*be.ras*⟩ and ⟨*ber.as*⟩. Flipping onsets in each candidate is then performed. The score of each candidate for both standard and flipped syllabifications are then calculated and finally a candidate having the biggest score is chosen as the output. For example, the candidate ⟨*be.ras*⟩ obtains the biggest score so that is selected as the best syllabification. It can be easily explained as follows. Although both original bigram ⟨*be.ras*⟩ and its flipped version ⟨*re.bas*⟩ do not appear in the training set, they may come from other words that are augmented using the thee proposed methods and listed in the table of augmented bigrams, such as "***b**e.**l**as*" (mercy), "***p**e.**r**as*" (squeeze), "***p**e.**l**as*" (pity), and other words described in Section II. Hence, this candidate has a high score (probability). Meanwhile, the other candidate ⟨*ber.as*⟩ and its flipped version ⟨*er.bas*⟩ cannot come from other augmented words that are listed in the table of augmented bigrams so that it has a lower score than the first candidate. Therefore, the proposed CFTABS model is capable of syllabifying the given grapheme sequence ⟨*beras*⟩ into ⟨*be.ras*⟩.

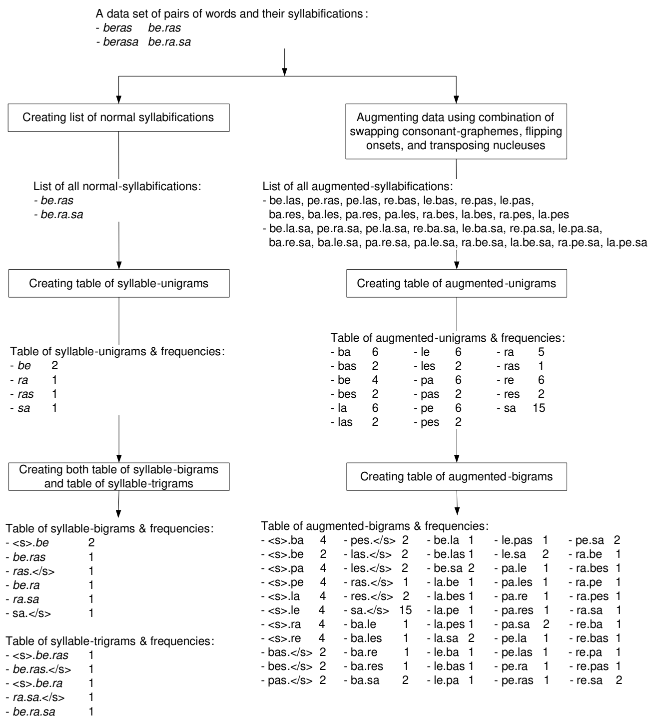

FIGURE 1. Training process of the proposed combination of flipping-onsets with standard-trigram and augmented-bigram syllabification (CFTABS).

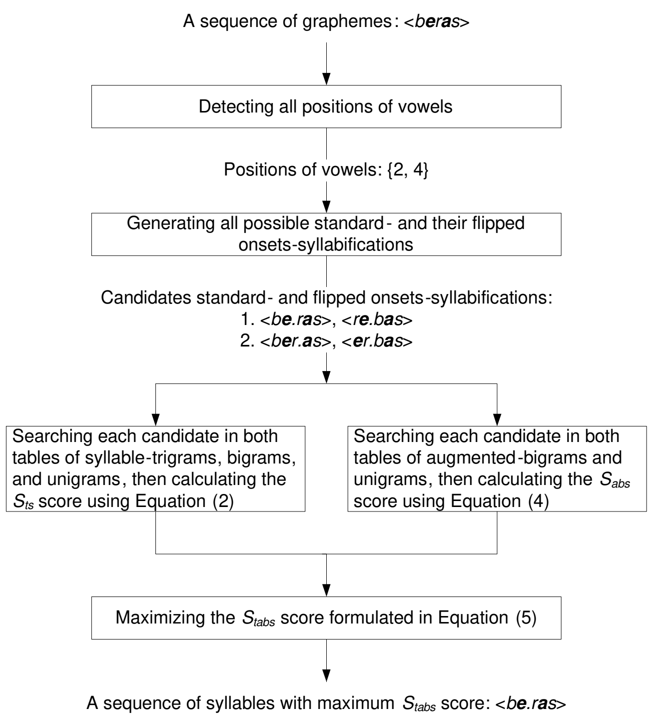

FIGURE 2. Block diagram of the proposed CFTABS model.

A. TRIGRAM-SYLLABIFICATION MODEL

A trigram-syllabification (TS) is a longer version of a bigram-syllabification (BS) described in [44]. It works by maximizing the likelihood (or probability) of a given sequence of syllables. A trigram-syllabification probability of L tokens $P(w_1, w_2, ..., w_L)$ is commonly calculated using a probability chain. This probability is commonly estimated using many smoothing methods to tackle the OOV problem. One of the smoothing methods is the *Stupid Backoff* described in [46], where the estimated probability called a score S (since it can be greater than 1) is formulated as

$$\begin{aligned} P(w_1, w_2, ..., w_L) &= \prod_{i=1}^{L} P(w_i|w_{i-2}w_{i-1}) \\ &\approx \prod_{i=1}^{L} S(w_i|w_{i-2}w_{i-1}), \end{aligned} \tag{1}$$

$$S_{ts}(w_i|w_{i-2}w_{i-1}) = \begin{cases} \frac{f(w_{i-2}w_{i-1}w_i)}{f(w_{i-2}w_{i-1})} \\ \text{if } f(w_{i-2}w_{i-1}w_i) > 0 \\ \\ \alpha S_{bs}(w_i|w_{i-1}) \text{ otherwise} \end{cases} \tag{2}$$

$$S_{bs}(w_i|w_{i-1}) = \begin{cases} \frac{f(w_{i-1}w_i)}{f(w_{i-1})} \text{ if } f(w_{i-1}w_i) > 0 \\ \\ \alpha \frac{f(w_i)}{N} \text{ otherwise} \end{cases} \tag{3}$$

where S_{ts} and S_{bs} are the scores of both trigram and bigram, respectively, $f(w_{i-2}w_{i-1}w_i)$, $f(w_{i-1}w_i)$, and $f(w_i)$ are the trigram, bigram, and unigram frequencies happen in the trainset, respectively, w_i is the ith syllable (or unigram), $w_{i-1}w_i$ is a bigram that is built from both (i-1)th and ith syllables, $w_{i-2}w_{i-1}w_i$ is a trigram that comes from (i-2)th, (i-1)th, and ith syllables, α is the backoff factor that commonly recommend to be 0.4 as described in [46], and N is the total grams in the trainset.

In general, TS produces quite low performance when the trainset is tiny with many OOV syllables [35]. Another technique to improve the performance of TS is a procedure of decreasing the OOV rate.

B. TFO MODEL

In this research, the combined trigram and flipping onsets-based syllabification model (TFO) is an improved version of the bigram and flipping onsets-based syllabification model (BFO) described in [43]. TFO is a modification of the TS, which is explained in the subsection III-A, by adding a simple procedure of flipping two first onsets in a word. Similar to the BFO described in [43], here TFO is easily described in a pseudocode below:

1) Detect positions of both vowels and diphthongs in the input grapheme sequence
2) Generate C possible bigram-syllabifications (all candidates) and then calculate their scores S_i using TS model, where $i = 1, 2, ..., C$;
3) For each candidate consisting of two or more syllables, generate a new candidate by flipping their onsets

contained in the first two syllables and then define the average score $\bar{S}_i$ calculated from both scores of TS and its flipped onsets; and

4) Choose the ith candidate that has the biggest $\bar{S}_i$ as the output.

C. CFTABS MODEL

In this research, the three proposed augmentation methods function to reduce the number of OOV grams in BS. They create a variant model, which is called augmented bigram-syllabification (ABS), with a new score formulated as

$$S_{abs}(w_i|w_{i-1}) = \begin{cases} B\frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} & \text{if } f_s(w_{i-1}w_i) > 0 \\ U\alpha\frac{f_s(w_i)}{N_s} & \text{otherwise,} \end{cases} \tag{4}$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the augmented- bigram and unigram frequencies happen in the normal trainset, w_i is the ith syllable (or unigram), $w_{i-1}w_i$ is a bigram that is composed from $(i-1)$th and ith syllables, N_s is the number of grams in the augmented training-set, B and U are the weights of augmented- bigram and unigram, respectively, and α is the backoff factor in Equation 3. As a bigram plays more critical role to syllabifying a word than the unigram, the value of B should be much bigger than U.

Next, a combined trigram and augmented bigram syllabification (TABS) model is created using a score S_{tabs} formulated as

$$S_{tabs} = S_{ts} + \beta S_{abs} \tag{5}$$

where S_{ts} is the trigram-syllabification score in Equation (2) and S_{abs} is the augmented-bigram-syllabification score in Equation (4), and β is the weight that is used in the augmented-bigram score.

Finally, both TFO and TABS models are then combined to be CFTABS, which takes into account unigrams, bigrams, and trigrams (that are built from the original words) as well as both augmented-unigrams and augmented-bigrams (that are developed from the augmented words). The detailed explanation of CFTABS with a simple example of syllabifying a word is illustrated in Fig 2.

D. THREE AUGMENTATION METHODS

Since most Indonesian graphemes are pronounced as the same phonemes, some onset graphemes can be swapped based on the phoneme categorization in [45]. Table 1 illustrates some Indonesian graphemes and their swaps. There are 14 graphemes and their swaps, each of which is simply mapped into those phoneme categorizations. This mapping is possible due to their strong relation to the corresponding phonemes [45] [47]. A word that consists of one or more graphemes in Table Table 1 can be swapped to produce one or more other words.

For example, both ⟨b⟩ and ⟨p⟩ are pronounced as plosive-bilabial phonemes. In general, swapping ⟨b⟩ into ⟨p⟩ generates some new legal unigrams and bigrams, such as "***ba***.*ku*"

TABLE 1. Graphemes and their swaps based on phoneme category

Phoneme category	Gr.	Sw,	Example
Plosive-Bilabial: {b, p}	b	p	***ba***.*ku* (standard) → ***pa***.*ku* (nail)
	p	b	***pe***.*ri* (fairy) → ***be***.*ri* (give)
Plosive-Dental: {d, t}	d	t	***de***.*bu* (dust) → ***te***.*bu* (cane)
	t	d	***ta***.*yang* (show) → ***da***.*yang* (court lady)
Plosive-Velar: {g, k}	g	k	***ga***.*bung* (join) → ***ka***.*bung* (mourning)
	k	g	***ka***.*mis* (thursday) → ***ga***.*mis* (clothes)
Affricative-Palatal: {c, j}	c	j	***ce***.*ruk* (niche) → ***je***.*ruk* (orange)
	j	c	***ja***.*wat* (stretched out) → ***ca***.*wat* (loincloth)
Fricative-Labiodental: {f, v}	f	v	***fo***.*li* (thin metal) → ***vo***.*li* (volley)
	v	f	***vi***.*si* (vision) → ***fi***.*si* (fission)
Fricative-Dental: {s, z}	s	z	*a*.***sam*** (acid) → *a*.***zam*** (aim)
	z	s	***ze***.*ni* (soldier) → ***se***.*ni* (art)
Thrill/Lateral-Dental: {l, r}	l	r	***lang***.*ka* (rare) → ***rang***.*ka* (frame)
	r	l	***ram***.*bu* (sign) → ***lam***.*bu* (canoe)

(standard) is swapped to be "***pa***.*ku*" (nail), "***ba***.*wang*" (onion) to be "***pa***.*wang*" (handler), "***be***.*ta*" (I am) to be "***pe***.*ta*" (map), etc. Swapping the grapheme ⟨p⟩ into ⟨b⟩ also generally creates several new syllable unigrams and bigrams, such as "***pe***.*ri*" (fairy) is swapped to be "***be***.*ri*" (give), "***pa***.*da*" (on) to be "***ba***.*da*" (after), "***pi***.*ta*" (tape) to be "***bi***.*ta*" (bytes), etc.

The grapheme ⟨d⟩ is in the same category with grapheme ⟨t⟩, i.e. plosive-dental. Swapping the grapheme ⟨d⟩ into ⟨t⟩ generally produces some new legal syllable unigrams and bigrams, such as "***de***.*bu*" (dust) is swapped to be "***te***.*bu*" (cane), "***de***.*bar*" (flutter) to be "***te***.*bar*" (spread out), "***da***.*ra*" (virgin) to be "***ta***.*ra*" (the same level), etc. Swapping the grapheme ⟨t⟩ into ⟨d⟩ also generally creates several new syllable unigrams and bigrams, such as "***ta***.*yang"* (show) can be swapped to be "***da***.*yang*" (court lady), "***ta***.*pa*" (asceticism) is swapped to be "***da***.*pa*" (ransom slaves), "***ta***.*rah*" (flat) to be "***da***.*rah*" (blood), etc.

Both ⟨g⟩ and ⟨k⟩ are plosive-velar. Swapping grapheme ⟨g⟩ into ⟨k⟩ generates several legal unigrams and bigrams, such as "***ga***.*bung*" (join) is swapped to be "***ka***.*bung*" (mourning), "***ge***.*tar*" (shakes) to be "***ke***.*tar*" (daunted), "***gi***.*la*" (crazy) to be "***ki***.*la*" (stake), etc. Swapping the grapheme ⟨k⟩ into ⟨g⟩ also generally creates several new syllable unigrams and bigrams, such as "***ka***.*mis*" (thursday) to be "***ga***.*mis*" (clothes), "***ke***.*mit*" (night guard) is swapped to be "***ge***.*mit*" (poke), "***ka***.*ri*" (curry) to be "***ga***.*ri*" (handcuffs), etc.

The grapheme ⟨c⟩ is in the same category with grapheme ⟨j⟩, i.e. affricative-palatal. Swapping the grapheme ⟨c⟩ into ⟨j⟩ commonly produces some legal syllable unigrams and bigrams, such as "***ce***.*ruk*" (niche) is swapped to be "***je***.*ruk*" (orange), "*ca.ri*" (search) is swapped to be "*ja.ri*" (finger), "*ca.har*" (liquid) to be "*ja.har*" (loud), etc. Swapping the grapheme ⟨j⟩ into ⟨c⟩ also generally creates several new syllable unigrams and bigrams, such as "*ja.wat"* (stretched out) is swapped to be "*ca.wat*" (loincloth), "*ja.ra*" (small drill) to be "*ca.ra*" (way), "*ja.ran*" (horse) to be "*ca.ran*" (woman fighting), etc.

In [45], the authors state that Indonesian does not have phoneme /v/, where the grapheme ⟨v⟩ is always pronounced as phoneme /f/ so that it can be swapped into ⟨f⟩. For instance, the word "*fo.li*" (thin metal) is swapped as "*vo.li*" (volley) and "*vi.si*" (vision) is swapped as "*fi.si*" (fission).

Both graphemes ⟨s⟩ and ⟨z⟩ are in the same category: fricative-dental. In general, swapping the grapheme ⟨s⟩ into ⟨z⟩ generates some new legal syllable unigrams and bigrams, such as *"a.sam"* (acid) is swapped to be *"a.zam"* (aim). Swapping the grapheme ⟨z⟩ into ⟨s⟩ converts a word "*ze.ni*" (soldier) into "*se.ni*" (art).

Finally, both graphemes ⟨l⟩ and ⟨r⟩ are considerably in the same category: thrill/lateral-dental. In general, swapping the grapheme ⟨s⟩ into ⟨z⟩ generates some new legal syllable unigrams and bigrams, such as "*lang.ka*" (rare) is swapped to be "*rang.ka*" (frame). Swapping the grapheme ⟨z⟩ into ⟨s⟩ converts a word "*ram.bu*" (sign) into "*lam.bu*" (canoe).

However, swapping graphemes does not always produce other formal words. Sometimes, it creates an illegal word. For instance, swapping grapheme "p" in the word "*pa.ha*" (thigh) creates a new word "*ba.ha*", which is illegal (OOV). But, the interesting fact is that the new word is a sub-word that comes from another word "*ba.ha.gi.a*" (happy). Of course, the swapped word increase the number of bigrams. Thus, the swapping word can be considered as one of the data augmentation methods. It can be expected to provide a more accurate score in Equation (5) to give a better syllabification.

Furthermore, Table 2 shows some examples of augmented words that are built from two original words without changing the syllabification points. First, the original word "*be.ri"* is swapped for all combinations to produce three new words: "*be.li"* (buy), "*pe.ri*" (fairy), and "*pe.li*" (OOV). Both original and swapped words are then augmented using the flipping onsets to generate four OOV words: "*re.bi*", "*le.bi*", "*re.pi*", and "*le.pi*". Finally, all the original, swapped, and flipped words are augmented using transposing their nuclei to produce eight OOV words: "*bi.re"*, "*bi.le"*, "*pi.re*", "*pi.le*", "*ri.be*", "*li.be*", "*ri.pe*", and "*li.pe*". Thus, this original word is augmented to be 15 new words, where only two words are formally found in the Indonesian dictionary while the rests are OOV words. But, an interesting phenomenon is the OOV words can be some sub-words for many other formal words. For example, the OOV word "*pe.li*" is a sub-word that comes from other words: "*pe.li.as*" (spell), "*pe.li.cin*" (lubricant), "*pe.li.ta*" (light), etc. The second formal word "*ba.tu*"is also augmented into 15 new words, where seven words are formal and the rests are OOV words that can also be some sub-words for many other formal words. Therefore, no doubt that the augmented words are capable of increasing the number of both unigrams and bigrams, which is expected to make the $_{cbsps}$ score in Equation (5) more accurate.

TABLE 2. Example of some augmented words that are generated using combination of swapping consonants-graphemes, flipping onsets, and transposing nuclei in the original words without shifting the syllabification points

Original word	Augmented words
be.ri (give)	*be.li* (buy), *pe.ri* (fairy), *pe.li* (OOV), *re.bi* (OOV), *le.bi* (OOV), *re.pi* (OOV), *le.pi* (OOV), *bi.re* (OOV), *bi.le* (OOV), *pi.re* (OOV), *pi.le* (OOV), *ri.be* (OOV), *li.be* (OOV), *ri.pe* (OOV), *li.pe* (OOV)
ba.tu (stone)	*ba.du* (checkered patterned), *pa.tu* (small pickaxe), *pa.du* (coherent), *ta.bu*(taboo), *da.bu* (OOV), *ta.pu* (OOV), *da.pu* (OOV), *bu.ta* (blind), *bu.da* (OOV), *pu.ta* (OOV), *pu.da* (OOV), *tu.ba* (tube), *du.ba* (OOV), *tu.pa* (OOV), *du.pa* (incense)

IV. RESULT AND DISCUSSION

The three parameters of the proposed CFTABS: U, B, and β are jointly optimized using a fixed $\alpha = 0.4$ as suggested in [46]. The result is illustrated in Fig. 3. The optimum parameters are $U = 0.1$, $B = 100$, and $\beta = 0.75$ that give the lowest SER of 2.37%. This result proves the hypothesis explained in subsection III-C, where the value of B should be much bigger than U since bigram is more critical than unigram in syllabifying a word. The optimum $\beta = 0.75$ also makes sense because the percentage of legal bigrams produced by the proposed augmentation methods is 77.26% as stated in Section II.

The CFTABS is then compared to five other models: BS [43], BFO [43], CBSPS [44], FkNNC [17], and BiLSTM-CNN-CRF [39]. Evaluation based on 5-FCV using a dataset consisting of 50 k Indonesian words explained in [17] [43] [44].

The results that are shown in Fig. 4 show that CFTABS better than three other bigram-based models but worse than FkNNC as well as BiLSTM-CNN-CRF. It produces SER of 2.37% that is lower than BS, BFO, and CBSPS with average SER of 3.80%, 3.11%, and 2.61% respectively. It means that CFTABS relatively reduces the average SER of BS by up to 37.63%. FkNNC gives a slightly lower SER of 2.27%. Meanwhile, BiLSTM-CNN-CRF reaches the lowest SER of 0.44%.

Based on the results, the proposed CFTABS is comparable to FkNNC. But, by offering a low complexity, it can be more preferable than FkNNC. It just computes the probabilities of tens or less candidates based on both original and augmented n-grams to decide a syllabification point. Meanwhile, FkNNC should: firstly, computes the dissimilarities between a candidate pattern of syllabification and the others in the trainset (up to 250 k patterns); secondly, chooses k neighbours in each class; finally, select the smallest dissimilarity to make a decision. Compared to BiLSTM-CNN-CRF in terms of complexity, the proposed CFTABS is also better. It needs much lower training time (only ten minutes) than BiLSTM-CNN-CRF (up to ten hours).

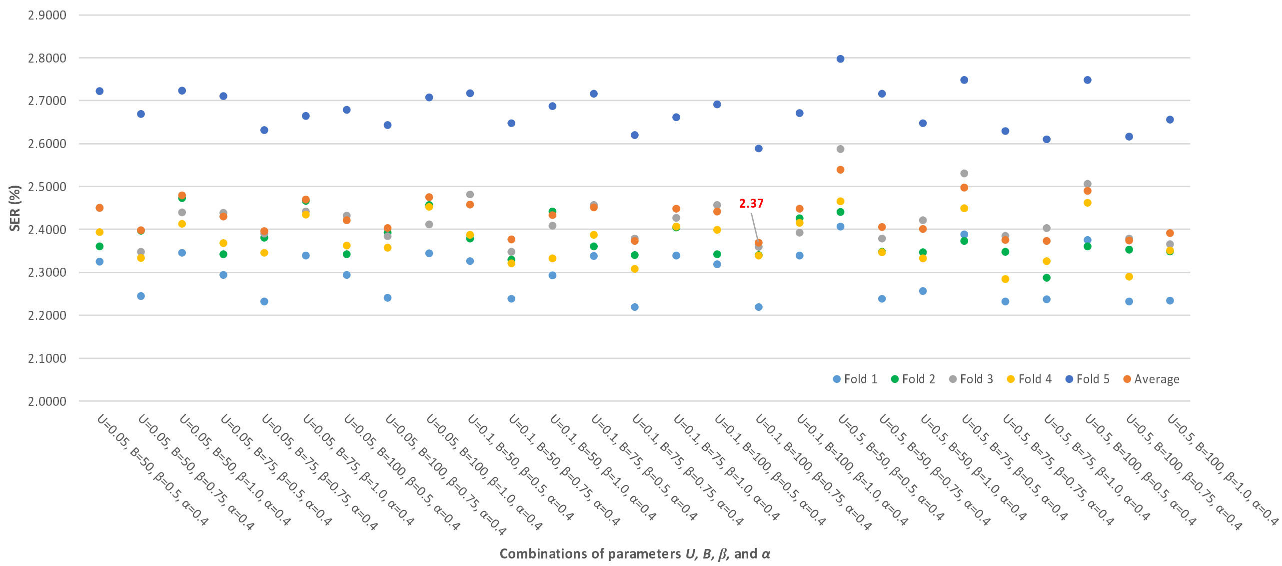

FIGURE 3. SERs produced by the proposed CFTABS using a fixed $\alpha = 0.4$ and jointly optimization of three parameters: U, B, and β.

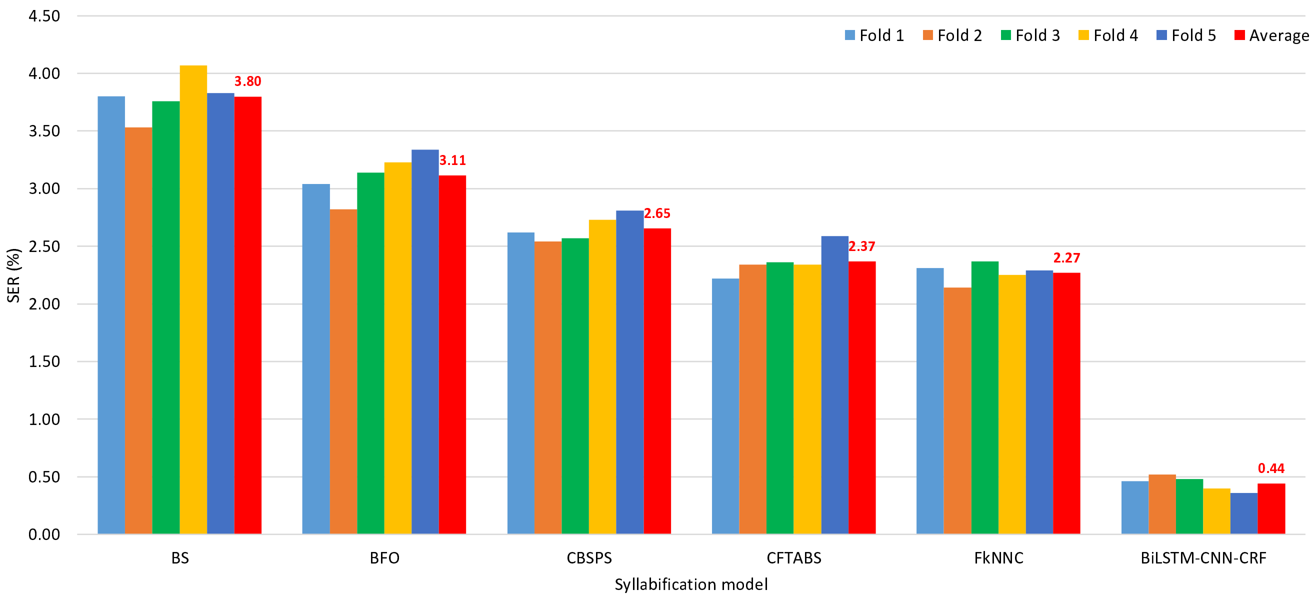

FIGURE 4. SERs produced by BS, BFO, CBSPS, CFTABS, and FkNNC for each fold and the average.

However, the proposed CFTABS is a bit unstable. It produces low SERs for Fold 1, Fold 2, Fold 3, and Fold 4 but it gives a higher SER for Fold 5. A filtering procedure can be introduced to select the possible legal-bigrams. For instance, a swapped word "*zdlug.dul*" (OOV) that comes from the formal word "*struk.tur*" (structure) should be detected as an illegal bigram.

Besides, CFTABS also has difficulty to differentiates a diphthong from a regular sequence of grapheme and suffix since the input is a grapheme sequence (not phoneme sequence). For example, a diphthong ⟨ei⟩ is hard to be distinguished from a grapheme sequence of ⟨e⟩ and the suffix ⟨i⟩. The detail investigation shows that most SER produced by this case since Indonesian has up to eighteen suffixes [45]. This problem can be solved by adding a procedure of diphthong recognition.

Another crucial problem is that CFTABS is applied on the syllable-level, which produces many OOV grams. Although three augmentation methods have been applied, the OOV rate is still high. Therefore, a grapheme-level is potentially applied to reduce the OOV rate. For instance, a word "*struktur*" (structure) just produces a syllable-level bigram of "*struk.tur*". But, it generates many grapheme-level bigrams, trigrams, until 8-gram: "*st*", "*tr*", "*ru*", ... "*struk.tur*". However, the use of grapheme-level approach will make the complexity of the model slightly higher.

V. CONCLUSION

The proposed CFTABS is capable of improving the performance of BS model, where the average SER is relatively decreased by up to 37.63%. It is comparable to the FkNNC-based syllabification and offer simplicity as well as flexibility since it just calculates the combined probabilities of both standard and augmented trigrams, bigrams, and unigrams to accurately define the syllabification points. Meanwhile, CFTABS gives a higher SER than BiLSTM-CNN-CFR but it provides a faster training time. In the future, a particular procedure to filter legal bigrams as well as unigrams can be introduced to increase its performance. Another improvement can also be performed by using grapheme-level grams, instead of the syllable ones.

ACKNOWLEDGEMENTS

This research is fully funded by the Research and Community Service, Telkom University.

REFERENCES

[1] S. Feng and T. Lee, "Exploiting Cross-Lingual Speaker and Phonetic Diversity for Unsupervised Subword Modeling," IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 27, no. 12, pp. 2000–2011, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8818297

[2] E. Pakoci, B. Popović, and D. Pekar, "Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition," Computational Intelligence and Neuroscience, vol. 2019, 2019.

[3] S. Geeta and B. L. Muralidhara, "Syllable as the basic unit for Kannada speech synthesis," in Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017, vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1205–1208. [Online]. Available: https://ieeexplore.ieee.org/document/8299954

[4] Aripin, H. Haryanto, and S. Sumpeno, "A realistic visual speech synthesis for Indonesian using a combination of morphing viseme and syllable concatenation approach to support pronunciation learning," International Journal of Emerging Technologies in Learning, vol. 13, no. 8, pp. 19–37, 2018.

[5] D. Magdum and M. Suman, "System for identifying and correcting invalid words in the devanagari script for text to speech engine," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6 Special Issue 4, pp. 1001–1006, 2019.

[6] S. Ben Alex, B. P. Babu, and L. Mary, "Utterance and syllable level prosodic features for automatic emotion recognition," in 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018. Institute of Electrical and Electronics Engineers Inc., 2019, pp. 31–35. [Online]. Available: https://ieeexplore.ieee.org/document/8635059

[7] L. Sun, S. Fu, and F. Wang, "Decision tree SVM model with Fisher feature selection for speech emotion recognition," Eurasip Journal on Audio, Speech, and Music Processing, vol. 2019, no. 1, 2019.

[8] A. Leemann, M.-J. Kolly, F. Nolan, and Y. Li, "The role of segments and prosody in the identification of a speaker's dialect," Journal of Phonetics, vol. 68, pp. 69–84, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0095447016300365

[9] S. Nayak, S. Bhati, and K. S. Rama Murty, "Zero Resource Speaking Rate Estimation from Change Point Detection of Syllable-like Units," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., 2019, pp. 6590–6594. [Online]. Available: https://ieeexplore.ieee.org/document/8683462

[10] D. O. Johnson and O. Kang, "Comparison of algorithms to divide noisy phone sequences into syllables for automatic unconstrained English speaking proficiency scoring," Artificial Intelligence Review, vol. 52, no. 3, pp. 1781–1804, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10462-017-9594-y

[11] O. Räsänen, S. Seshadri, J. Karadayi, E. Riebling, J. Bunce, A. Cristia, F. Metze, M. Casillas, C. Rosemberg, E. Bergelson, and M. Soderstrom, "Automatic word count estimation from daylong child-centered recordings in various language environments using language-independent syllabification of speech," Speech Communication, vol. 113, pp. 63–80, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639318304205

[12] S. Suyanto, "Incorporating syllabification points into a model of grapheme-to-phoneme conversion," International Journal of Speech Technology, vol. 22, no. 2, pp. 459–470, jun 2019. [Online]. Available: https://doi.org/10.1007/s10772-019-09619-4

[13] A. Rugchatjaroen, S. Saychum, S. Kongyoung, P. Chootrakool, S. Kasuriya, and C. Wutiwiwatchai, "Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion," Speech Communication, vol. 106, pp. 105–111, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639317303965

[14] B. N. Budi, Nurtomo, and S. Suyanto, "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. [Online]. Available: https://thesai.org/Downloads/Volume9No10/Paper_35-Greedy_Algorithms_to_Optimize_a_Sentence.pdf

[15] F. Alfiansyah and Suyanto, "Partial greedy algorithm to extract a minimum phonetically-and-prosodically rich sentence set," International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, pp. 530–534, 2018. [Online]. Available: https://thesai.org/Downloads/Volume9No12/Paper_74-Partial_Greedy_Algorithm_to_Extract_a_Minimum.pdf

[16] S. N. Hidayatullah and Suyanto, "Developing an adaptive language model for Bahasa Indonesia," International Journal of Advanced Computer Science and Applications, vol. 10, no. 1, pp. 488–492, 2019. [Online]. Available: https://thesai.org/Downloads/Volume10No1/Paper_63-Developing_an_Adaptive_Language_Model.pdf

[17] E. A. Parande and S. Suyanto, "Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure," International Journal of Speech Technology, vol. 22, no. 1, pp. 13–20, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10772-018-09569-3

[18] S. Suyanto, S. Hartati, A. Harjoko, and D. V. Compernolle, "Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge," Speech Communication, vol. 85, pp. 109–118, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2016.10.009

[19] C. R. Adsett, Y. Marchand, and V. Kešelj, "Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian," Computer Speech and Language, vol. 23, pp. 444–463, 2009.

[20] D. Balc, A. Beleiu, R. Potolea, and C. Lemnaru, "A learning-based approach for Romanian syllabification and stress assignment," in 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2015, pp. 37–42.

[21] W. Daelemans, A. Van Den Bosch, and T. Weijters, "IGTree: Using trees for compression and classification in lazy learning algorithms," Artificial Intelligence Review, vol. 11, no. 1-5, pp. 407–423, 1997.

[22] K. Müller, "Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpora training," in Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. ACL, 2001, pp. 410–417.

[23] A. Hunt, "Recurrent neural networks for syllabification," Speech Communication, vol. 13, no. 3, pp. 323–332, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016763939390031F

[24] W. Daelemans and A. V. D. Bosch, "A neural network for hyphenation," in Proceedings of the International Conference on Artificial Neural Networks (ICANN–92), Brighton, United Kingdom, 1992, pp. 1647–1650 (vol. 2).

[25] T. Kristensen, "A neural network approach to hyphenating Norwegian," in

Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IEEE, 2000, pp. 148–153 vol.2.
[26] J. Tian, "Data-driven approaches for automatic detection of syllable boundaries," in Proceedings of the International Conference on Spoken Language Processing (ICSLP), 2004, pp. 61–64.
[27] S. Bartlett, G. Kondrak, and C. Cherry, "On the syllabification of phonemes," in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 308–316.
[28] T. H. Hlaing and Y. Mikami, "Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer," International Journal on Advances in ICT for Emerging Regions (ICTer), vol. 6, no. 2, pp. 2–9, 2014.
[29] G. A. Kiraz, M. Bernd, B. Labs, L. Technologies, and M. Hill, "Multilingual syllabification using weighted finite-state transducers," in Proceedings of the Third ESCA/COCOSDA Workshop on Speech Synthesis, 1998, pp. 59–64.
[30] K. Müller, "Improving syllabification models with phonotactic knowledge," in Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology - SIGPHON '06, 2006, pp. 11–20.
[31] J. Krantz, M. Dulin, P. De Palma, and M. VanDam, "Syllabification by Phone Categorization," in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ser. GECCO '18. New York, NY, USA: ACM, 2018, pp. 47–48. [Online]. Available: http://doi.acm.org/10.1145/3205651.3208781
[32] I. Ramli, N. Jamil, N. Seman, and N. Ardi, "An Improved Syllabification for a Better Malay Language Text-to- Speech Synthesis (TTS)," Procedia - Procedia Computer Science, vol. 76, no. Iris, pp. 417–424, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.12.280
[33] H. Schmid, B. Möbius, and J. Weidenkaff, "Tagging syllable boundaries with joint n-gram models," Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 1, no. 1, pp. 49–52, 2007.
[34] L. G. Singh, L. Laitonjam, and S. R. Singh, "Automatic Syllabification for Manipuri language," in the 26th International Conference on Computational Linguistics, 2016, pp. 349–357. [Online]. Available: https://www.aclweb.org/anthology/papers/C/C16/C16-1034/
[35] K. Rogova, K. Demuynck, and D. V. Compernolle, "Automatic syllabification using segmental conditional random fields," Computational Linguistics in the Netherlands Journal, vol. 3, pp. 34–48, 2013. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907935327&partnerID=40&md5=ec7496ee067bb9d2aca02f45d90d6bd0
[36] T. Mayer, "Toward a totally unsupervised, language-independent method for the syllabification of written texts," in Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, 2010, pp. 63–71.
[37] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=ByJHuTgA-
[38] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=HkwZSG-CZ
[39] J. Krantz, M. Dulin, and P. De Palma, "Language-agnostic syllabification with neural sequence labeling," in Proceedings of 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019, pp. 804–810.
[40] R. Tang and J. Lin, "Progress and Tradeoffs in Neural Language Models," 2018.
[41] E. Shareghi, D. Gerz, I. Vulić, and A. Korhonen, "Show Some Love to Your n-grams: A Bit of Progress and Stronger n-gram Language Modeling Baselines," no. 2018, pp. 4113–4118, 2019.
[42] Y. Doval and C. Gómez-Rodríguez, "Comparing neural- and N-gram-based language models for word segmentation," Journal of the Association for Information Science and Technology, vol. 70, no. 2, pp. 187–197, 2019.
[43] S. Suyanto, "Flipping onsets to enhance syllabification," International Journal of Speech Technology, vol. 22, no. 4, pp. 1031–1038, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10772-019-09649-y
[44] ——, "Phonological similarity-based backoff smoothing to boost a bigram syllable boundary detection," International Journal of Speech Technology, 2020. [Online]. Available: https://doi.org/10.1007/s10772-020-09677-z
[45] H. Alwi, S. Darmowidjojo, H. Lapoliwa, and A. M. Moeliono, Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3rd ed. Jakarta: Balai Pustaka, 2014.
[46] T. Brants, A. C. Popat, and F. J. Och, "Large Language Models in Machine Translation," in The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, vol. 1, no. June, 2007, pp. 858–867.
[47] Suyanto and A. Harjoko, "Nearest neighbour-based Indonesian G2P conversion," Telkomnika (Telecommunication Computing Electronics and Control), vol. 12, no. 2, 2014.

SUYANTO received his B.Sc. on Informatics Engineering from STT Telkom (now: Telkom University), Bandung, Indonesia in 1998, the M.Sc. on Complex Adaptive Systems from Chalmers University of Technology, Goteborg, Sweden, in 2006, and the Ph.D. on Computer Science from Universitas Gadjah Mada in 2016. Since 2000, he joined STT Telkom as a lecturer in School of Computing. His research interests include artificial intelligence, machine learning, deep learning, swarm intelligence, speech processing, and computational linguistics.

KEMAS M. LHAKSMANA received his bachelor degree in Informatics from Institut Teknologi Bandung, Bandung, Indonesia, in 2005, and earned master's degree in Information Systems Development from HAN University of Applied Sciences, Arnhem, Netherlands, in 2009. He received Ph.D. degree in Social Informatics from Kyoto University, Kyoto, Japan. He has been a lecturer of Department of Software Engineering, Telkom University, Indonesia, since 2011. His research interest lies in the text mining, services computing, and multiagent systems.

MOCH ARIF BIJAKSANA received the B.Sc. on Electrical Engineering from Gadjah Mada University, Indonesia in 1991, the M.Tech on Coputer Science from RMIT Universty, Australia in 1999, and the Ph.D. on Computer Science from Queensland Universty of Technology (QUT), Australia in 2015. Since 2003, he joined Telkom University as a lecturer in School of Computing. His research interests include deep learning and natural language processing.

ADRIANA KURNIAWAN received her B.Sc. on Information System from Politeknik Caltex Riau, Indonesia in 2016, and master's degree in Informatics from Telkom University, Bandung, Indonesia in 2020. Her research interests include artificial intelligence, machine learning, deep learning, speech processing, and computational linguistics.

Evidence of correspondence

Data Augmentation Methods for Low-Resource Orthographic Syllabification

1. First submission (19 July 2020)
2. LoA with Minor Revision (06 Augustus 2020)
3. Responses to Reviewers, Final submission (08 August 2020)
4. Final paper of source latex submitted (08 August 2020)
5. Creative Commons Attribution License (08 August 2020)

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

IEEE Access - Decision on Manuscript ID Access-2020-35994

IEEE Access <onbehalfof@manuscriptcentral.com> Thu, Aug 6, 2020 at 9:49 PM
Reply-To: y.yang@ed.ac.uk
To: suyanto@telkomuniversity.ac.id
Cc: y.yang@ed.ac.uk, suyanto@telkomuniversity.ac.id, kemas.muslim@gmail.com, arifbijaksana@telkomuniversity.ac.id, adriana@student.telkomuniversity.ac.id

06-Aug-2020

Dear Dr. Suyanto:

Your manuscript entitled "Data Augmentation Methods for Low-Resource Orthographic Syllabification" has been accepted for publication in IEEE Access. The comments of the reviewers who reviewed your manuscript are included at the foot of this letter. We ask that you make changes to your manuscript based on those comments, before uploading final files.

However, NO CHANGES to the author list or the references will be permitted.

Finally, please improve the English grammar and check spelling, as it is only lightly edited before publication.

Once you have updated your article accordingly, please send all final versions of your files through the “Awaiting Final Files” queue in your Author Center on ScholarOne Manuscripts. Once you have completed the submission of your final files, you will not be able to make changes until you have received your page proofs from IEEE.

When submitting final files, you must submit all of the items in the list below. All files intended for publication need to be submitted during this step, even if some files are unchanged from initial submission. If you do not submit all files during this step, it will delay the publication of your article, or result in certain files not being published.

1) Manuscript in MS Word or LaTex.
2) A PDF of the final manuscript in double column, single-spaced format named "FINAL Article.pdf".
3) Biographies and author photos in MS Word.
4) Figures/photos saved as separate PDF, Word, .eps, .ps, or .tiff files (if not embedded in the source file)
5) Video(s) included in peer review (if any)
6) A Graphical Abstract (GA) which provides a concise, visual summary of the findings of your article. The GA should be a figure or image from the accepted article. If you submitted a video with your article, the video will automatically become the GA and you will need to supply a still image to act as an overlay. For more information on the GA, please visit https://ieeeaccess.ieee.org/submitting-an-article/
7) A Word file that indicates: a) the file name(s) of the GA and overlay (if applicable), b) a caption for the GA that should not exceed 60 words.

Copyright Information / APC instructions:

After you submit final files you will automatically be directed to the Electronic Copyright Form. Once the copyright information is completed, within a few business days you will receive an email from Copyright Clearance Center (CCC) to settle your balance by check, credit card, or wire transfer. If you need assistance with the site or payment process, please contact CCC Customer Service at IEEESupport@copyright.com.

Thank you for your fine contribution. On behalf of the Editors of IEEE Access, we look forward to your continued contributions to IEEE Access.

Sincerely,
Dr. Yunjie Yang
Associate Editor, IEEE Access
y.yang@ed.ac.uk, y.yang@ed.ac.uk

Reviewer(s)' Comments to Author:

Reviewer: 1

Recommendation: Accept (minor edits)

Comments:
The author must follow on style in REFERENCES writting

Additional Questions:
Does the paper contribute to the body of knowledge?: yes

Is the paper technically sound?: yes

Is the subject matter presented in a comprehensive manner?: yes

Are the references provided applicable and sufficient?: yes. However, the author must follow on style in REFERENCES writting

Reviewer: 2

Recommendation: Accept (minor edits)

Comments:
None.

Additional Questions:
Does the paper contribute to the body of knowledge?: Yes, It proposes a new method for low resource orthographic syllabification which gives better results.

Is the paper technically sound?: Yes, it is.

Is the subject matter presented in a comprehensive manner?: It seems fine now.

Are the references provided applicable and sufficient?: Yes

If you have any questions, please contact article administrator: Miss Manpreet Kaur m.kaur@ieee.org

IEEE Access

Decision Letter (Access-2020-35994)

From: y.yang@ed.ac.uk

To: suyanto@telkomuniversity.ac.id

CC: y.yang@ed.ac.uk, suyanto@telkomuniversity.ac.id, kemas.muslim@gmail.com, arifbijaksana@telkomuniversity.ac.id, adriana@student.telkomuniversity.ac.id

Subject: IEEE Access - Decision on Manuscript ID Access-2020-35994

Body: 06-Aug-2020

Dear Dr. Suyanto:

Your manuscript entitled "Data Augmentation Methods for Low-Resource Orthographic Syllabification" has been accepted for publication in IEEE Access. The comments of the reviewers who reviewed your manuscript are included at the foot of this letter. We ask that you make changes to your manuscript based on those comments, before uploading final files.

However, NO CHANGES to the author list or the references will be permitted.

Finally, please improve the English grammar and check spelling, as it is only lightly edited before publication.

Once you have updated your article accordingly, please send all final versions of your files through the "Awaiting Final Files" queue in your Author Center on ScholarOne Manuscripts. Once you have completed the submission of your final files, you will not be able to make changes until you have received your page proofs from IEEE.

When submitting final files, you must submit all of the items in the list below. All files intended for publication need to be submitted during this step, even if some files are unchanged from initial submission. If you do not submit all files during this step, it will delay the publication of your article, or result in certain files not being published.

1) Manuscript in MS Word or LaTex.
2) A PDF of the final manuscript in double column, single-spaced format named "FINAL Article.pdf".
3) Biographies and author photos in MS Word.
4) Figures/photos saved as separate PDF, Word, .eps, .ps, or .tiff files (if not embedded in the source file)
5) Video(s) included in peer review (if any)
6) A Graphical Abstract (GA) which provides a concise, visual summary of the findings of your article. The GA should be a figure or image from the accepted article. If you submitted a video with your article, the video will automatically become the GA and you will need to supply a still image to act as an overlay. For more information on the GA, please visit https://ieeeaccess.ieee.org/submitting-an-article/
7) A Word file that indicates: a) the file name(s) of the GA and overlay (if applicable), b) a caption for the GA that should not exceed 60 words.

Copyright Information / APC instructions:

After you submit final files you will automatically be directed to the Electronic Copyright Form. Once the copyright information is completed, within a few business days you will receive an email from Copyright Clearance Center (CCC) to settle your balance by check, credit card, or wire transfer. If you need assistance with the site or payment process, please contact CCC Customer Service at IEEESupport@copyright.com.

Thank you for your fine contribution. On behalf of the Editors of IEEE Access, we look forward to your continued contributions to IEEE Access.

Sincerely,
Dr. Yunjie Yang
Associate Editor, IEEE Access
y.yang@ed.ac.uk, y.yang@ed.ac.uk

Reviewer(s)' Comments to Author:

Reviewer: 1

Recommendation: Accept (minor edits)

Comments:
The author must follow on style in REFERENCES writting

Additional Questions:
Does the paper contribute to the body of knowledge?: yes

Is the paper technically sound?: yes

Is the subject matter presented in a comprehensive manner?: yes

Are the references provided applicable and sufficient?: yes. However, the author must follow on style in REFERENCES writting

Reviewer: 2

Recommendation: Accept (minor edits)

Comments:
None.

Additional Questions:
Does the paper contribute to the body of knowledge?: Yes, It proposes a new method for low resource orthographic syllabification which gives better results.

Is the paper technically sound?: Yes, it is.

Is the subject matter presented in a comprehensive manner?: It seems fine now.

Are the references provided applicable and sufficient?: Yes

If you have any questions, please contact article administrator: Miss Manpreet Kaur
m.kaur@ieee.org

Date Sent: 06-Aug-2020

Close Window

Evidence of correspondence

Data Augmentation Methods for Low-Resource Orthographic Syllabification

1. First submission (19 July 2020)
2. LoA with Minor Revision (06 Augustus 2020)
3. **Responses to Reviewers, Final submission (08 August 2020)**
4. Final paper of source latex submitted (08 August 2020)
5. Creative Commons Attribution License (08 August 2020)

Responses to Reviewers

Reviewer: 1

Recommendation: Accept (minor edits):

Comments:

The author must follow on style in REFERENCES writting

Additional Questions:

Does the paper contribute to the body of knowledge?: yes

Is the paper technically sound?: yes

Is the subject matter presented in a comprehensive manner?: yes

Are the references provided applicable and sufficient?: yes. However, the author must follow on style inREFERENCES writting.

> In this revised manuscript, we now follow on style in REFERENCES writting.

Reviewer: 2

Recommendation: Accept (minor edits)

Comments:

None.

Additional Questions:

Does the paper contribute to the body of knowledge?: Yes, It proposes a new method for low resource orthographic syllabification which gives better results.

Is the paper technically sound?: Yes, it is.

Is the subject matter presented in a comprehensive manner?: It seems fine now.

Are the references provided applicable and sufficient?: Yes

>> Thank you very much.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Data Augmentation Methods for Low-Resource Orthographic Syllabification

SUYANTO SUYANTO, (Member, IEEE), KEMAS M. LHAKSMANA, (Member, IEEE), MOCH ARIF BIJAKSANA, AND ADRIANA KURNIAWAN

School of Computing, Telkom University, Bandung, Indonesia

Corresponding author: Suyanto Suyanto (e-mail:suyanto@telkomuniversity.ac.id, Orcid: https://orcid.org/0000-0002-8897-8091).

This research is funded by the Directorate of Research and Community Service PPM, Telkom University, with grant number: KWR.0067/PNLT3/PPM-LIT/2020.

ABSTRACT An n-gram syllabification model generally produces a high error rate for a low-resource language, such as Indonesian, because of the high rate of out-of-vocabulary (OOV) n-grams. In this paper, a combination of three methods of data augmentations is proposed to solve the problem, namely swapping consonant-graphemes, flipping onsets, and transposing nuclei. An investigation on 50k Indonesian words shows that the combination of three data augmentation methods drastically increases the amount of both unigrams and bigrams. A previous procedure of flipping onsets has been proven to enhance the standard bigram-syllabification by relatively decreasing the syllable error rate (SER) by up to 18.02%. Meanwhile, the previous swapping consonant-graphemes has been proven to give a relative decrement of SER up to 31.39%. In this research, a new transposing nuclei-based augmentation method is proposed and combined with both flipping and swapping procedures to tackle the drawback of bigram syllabification in handling the OOV bigrams. An evaluation based on k-fold cross-validation (k-FCV), using $k = 5$, for 50 thousand Indonesian formal words concludes that the proposed combination of the three procedures relatively decreases the mean SER produced by the standard bigram model by up to 37.63%. The proposed model is comparable to the fuzzy k-nearest neighbor in every class (FkNNC)-based model. It is worse than the state-of-the-art model, which is developed using a combination of bidirectional long short-term memory (BiLSTM), convolutional neural networks (CNN), and conditional random fields (CRF), but it offers a low complexity.

INDEX TERMS Indonesian, flipping onsets, orthographic syllabification, swapping consonant-graphemes, transposing nuclei

I. INTRODUCTION

A syllabification can be defined as a splitting a word into syllables automatically. It is important not just in some researches but also in many linguistics-based applications. It is generally used in speech recognition [1] [2], speech synthesis [3] [4] [5], emotion classification [6] [7], speaker's dialect identification [8], speaking rate estimation [9], speaking proficiency scoring [10], word count estimation [11], phonemicization [12] [13], collecting a minimum sentence set in developing speech corpus, as described in [14] [15] [16], etc.

The syllabification is preferably applied to graphemes than phonemes because of both simplicity and flexibility. Although a graphemic (or orthographic) syllabification generally gives lower accuracy [17] than the phonemic one [18], it can be easily applied to both unseen words and named-entities that have so many exceptions and ambiguities.

Most researchers prefer a statistical-based syllabification much more than the rule-based one as it is simpler and accurate [19]. For example, a simple Näive Bayes produces quite low SER of around 12.90% for the Romanian language [20]. Some other statistical models use decision tree [20] [21], treebank [22], random forest [20], neural network [23] [24] [25] [26], support vector machine [20] [27], finite-state transducers [28] [29], context-free grammars [30], hidden Markov model [31], syllabification by analogy [19], dropped-and-matched model [32], n-gram [33], conditional random fields [34] [35], nearest neighbour [17], and unsupervised model [36].

The neural language models proposed in [37] [38] give excellent results. A recent model based on BiLSTM-CNN-CRF gives state-of-the-art result [39]. However, some n-gram models produce comparable accuracies as well as offer simplicity and fast processing [40] [41] [42].

In [33], the researchers prove that n-gram syllabification, which is one of the simplest methods, reaches a low word error rate (WER) of 0.15% for the phonemic sequences of the Germany language. It can be generalized into any language since it does not need any knowledge of a particular language. However, the n-gram model is generally poor for a tiny dataset producing many OOV n-grams. In [43], the researcher proposes a simple combined standard bigram and flipping onsets model (BFO) to tackle the OOV problem. Compare to the standard bigram models, it can relatively reduce the SER by 18.02%. However, its performance is not stable for a tiny dataset.

In [44], the researcher proposes a simple backoff smoothing procedure called swapping phonological similarities (CBSPS) model, which can boost a bigram-based orthographic syllabification. It also performs better than the BFO model. In this research, a new model called a combination of flipping-onsets with standard-trigram and augmented-bigram syllabification (CFTABS) is proposed to solve the problem. The standard trigram is expected to perform better than bigram. Three augmentation methods of swapping consonant-graphemes, flipping onsets, and transposing nuclei are proposed to reduce the OOV rate. Next, CFTABS is evaluated and compared to BiLSTM-CNN-CRF [39] using 50 k Indonesian words based on a 5-fold cross-validation scheme.

II. PRELIMINARY STUDY ON INDONESIAN

For many languages, those three methods of data augmentations commonly create many illegal syllables for both unigrams and bigrams. But, for other simpler languages with a nearly one-to-one grapheme-to-phoneme mapping, such as Indonesian, they generate many legal ones. Swapping several consonant-graphemes in a word for all combinations may generate some new words. It is performed by replacing them with other similar ones in the same manner and/or place of articulations. For example, swapping consonant-graphemes in an original formal word "***b**e.**r**as*" (rice) generates three other words: "***b**e.**l**as*" (mercy), "***p**e.**r**as*" (squeeze), and "***p**e.**l**as*" (pity) with no shifting the syllabification boundaries as both ⟨b⟩ and ⟨p⟩ are pronounced as the plosive-bilabial phonemes while ⟨r⟩ and ⟨l⟩ are pronounced as the trill and lateral-dental phonemes. Flipping two onsets in the original word yields another word "***r**e.**b**as*" (boil) without shifting the syllabification point. Flipping two onsets in the three consonants-swapped words produces three other words: "***l**e.**b**as*" (too ripe), "***r**e.**p**as*" (fragile), and "***l**e.**p**as*" (free). Transposing two nuclei in the original word produces another new word, "*b**a**.r**e**s*" (OOV). Transposing two nuclei in all consonants-swapped and onsets-flipped words produces seven new words: "***ba**.**les***" (reply), "***pa**.**res***" (OOV), "***pa**.**les***" (discordant), "***re**.**bas***" (OOV), "***la**.**bes***" (OOV), "***ra**.**pes***" (OOV), and "***la**.**pes***" (OOV). Thus, in this case, the three methods augment a short original formal word "***be**.**ras***" (rice) into nine new formal words and six OOV words.

Furthermore, the Indonesian language has 18 prefixes [45]. Swapping some graphemes in the prefixes produces many more other legal suffixes than the illegal ones called noises, as described in [44]. A preliminary study shows that the dataset of 50 k Indonesian words produces up to 161,981 unigrams. Applying three augmentation procedures to the dataset generates up to 9,620,054 augmented unigrams (87.20% are legal). The 50 k words generate a total of 111,412 bigrams. The augmentation procedures yield 7,308,702 augmented bigrams (77.26% are legal). Finally, the 50 k words generate a total of 136,812 syllable trigrams. In this research, the augmented syllable trigrams are not generated since they produce a high sparsity of trigrams, and consequently, they are not useful in the syllabification process. Based on a particular criterion, such as a phonotactic rule, those grams can be classified into two classes: legal and illegal. However, it is quite hard to recognize them as legal or illegal. Therefore, CFTABS are designed to use all generated syllable unigrams and bigrams without filtering to focus this research on examining how much the proposed CFTABS decreases the SER.

III. RESEARCH METHOD

The proposed model's training process can be easily explained as the combination of normal (or standard) and swapped syllabifications. A dataset containing pairs of words and their syllabification points is scanned to create two lists of normal syllable unigrams and bigrams, as well as two lists of augmented syllable unigrams and bigrams, as illustrated in Fig. 1.

Next, the normal and swapped unigrams, as well as the normal and swapped bigrams, are used to test the model to maximize the final score to produce the best syllable sequence as the output, as illustrated in Fig. 2. For example, let a given grapheme sequence is ⟨*beras*⟩ (rice). First, two vowel-graphemes ⟨***e***⟩ and ⟨***a***⟩ are searched and their positions are listed as {2, 4}. Next, two candidates syllabifications are then generated, i.e. ⟨*be.ras*⟩ and ⟨*ber.as*⟩. Flipping onsets in each candidate is then performed. The scores of each candidate for both standard and flipped syllabifications are then calculated and finally, a candidate having the biggest score is chosen as the output. For example, the candidate ⟨*be.ras*⟩ obtains the biggest score, and consequently, it is selected as the best syllabification. It can be easily explained as follows. Although both original bigram ⟨*be.ras*⟩ and its flipped version ⟨*re.bas*⟩ do not appear in the training set, they may come from other words that are augmented using the thee proposed methods and listed in the table of augmented bigrams, such as "***be**.**las***" (mercy), "***pe**.**ras***" (squeeze), "***pe**.**las***" (pity), and other words described in Section II. Hence, this candidate has a high score (probability). Meanwhile, the other candidate ⟨*ber.as*⟩ and its flipped version ⟨*er.bas*⟩ cannot come from

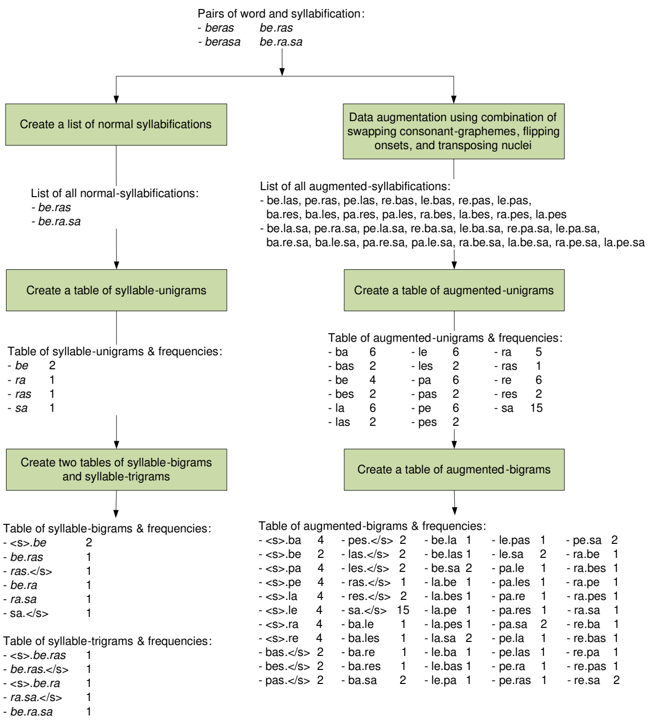

FIGURE 1. Training process of the proposed combination of flipping-onsets with standard-trigram and augmented-bigram syllabification (CFTABS).

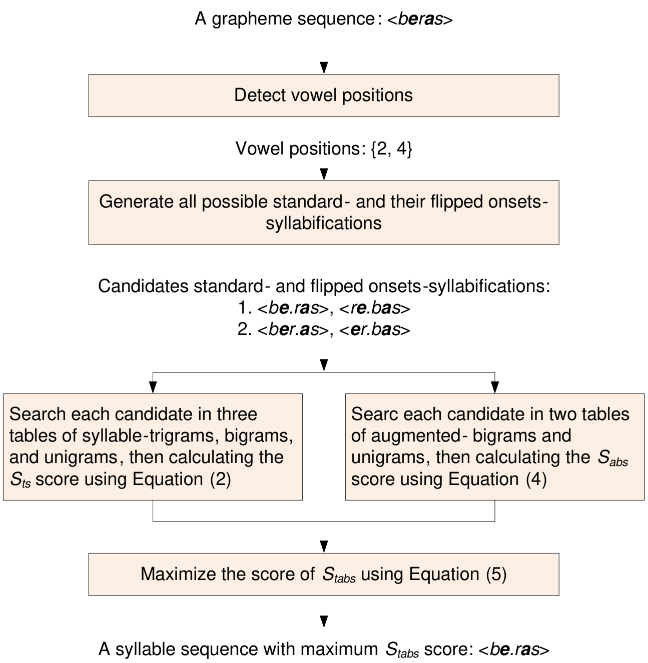

FIGURE 2. Testing process of the proposed CFTABS model.

other augmented words listed in the table of augmented bigrams so that it has a lower score than the first candidate. Therefore, the proposed CFTABS model is able to syllabify the given grapheme sequence $\langle$*beras*$\rangle$ into $\langle$*be.ras*$\rangle$.

A. TRIGRAM-SYLLABIFICATION MODEL

A trigram-syllabification (TS) is a longer version of a bigram-syllabification (BS) described in [44]. It works by maximizing the likelihood (or probability) of a given sequence of syllables. A trigram-syllabification probability of L tokens $P(w_1, w_2, ..., w_L)$ is commonly calculated using a probability chain. This probability is commonly estimated using many smoothing methods to tackle the OOV problem. One of the smoothing methods is the *Stupid Backoff* described in [46], where the estimated probability called a score S (since it can be greater than 1) is formulated as

$$P(w_1, w_2, ..., w_L) = \prod_{i=1}^{L} P(w_i|w_{i-2}w_{i-1}) \approx \prod_{i=1}^{L} S(w_i|w_{i-2}w_{i-1}), \tag{1}$$

$$S_{ts}(w_i|w_{i-2}w_{i-1}) = \begin{cases} \frac{f(w_{i-2}w_{i-1}w_i)}{f(w_{i-2}w_{i-1})} \\ \text{if } f(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha S_{bs}(w_i|w_{i-1}) \text{ otherwise} \end{cases} \tag{2}$$

$$S_{bs}(w_i|w_{i-1}) = \begin{cases} \frac{f(w_{i-1}w_i)}{f(w_{i-1})} \text{ if } f(w_{i-1}w_i) > 0 \\ \alpha \frac{f(w_i)}{N} \text{ otherwise} \end{cases} \tag{3}$$

where S_{ts} and S_{bs} are the scores of both trigram and bigram, respectively, $f(w_{i-2}w_{i-1}w_i)$, $f(w_{i-1}w_i)$, and $f(w_i)$ are the trigram, bigram, and unigram frequencies happen in the trainset, respectively, w_i is the ith syllable (or unigram), $w_{i-1}w_i$ is a bigram that is built from two syllables with indices of (i-1) and i, $w_{i-2}w_{i-1}w_i$ is a trigram that comes from three two syllables with indices of (i-2), (i-1), and i, α represents the backoff factor that commonly recommend to be 0.4 as described in [46], and N is the total grams in the trainset.

In general, TS produces quite low performance when the trainset is tiny with many OOV syllables [35]. Another technique to improve the performance of TS is a procedure of decreasing the OOV rate.

B. TFO MODEL

In this research, the combined trigram and flipping onsets-based syllabification model (TFO) is an improved version of the bigram and flipping onsets-based syllabification model (BFO) described in [43]. TFO is a modification of the TS, which is explained in subsection III-A, by adding a simple procedure of flipping two first onsets in a word. Similar to the BFO described in [43], here TFO is easily described in the pseudocode below:

1) Detect positions of both vowels and diphthongs in the input grapheme sequence
2) Generate C possible bigram-syllabifications (all candidates) and then calculate their scores S_i using TS model, where $i = 1, 2, ..., C$;

3) For each candidate consisting of two or more syllables, generate a new candidate by flipping their onsets contained in the first two syllables and then define the average score $\bar{S}_i$ calculated from both scores of TS and its flipped onsets; and
4) Choose the ith candidate that has the biggest $\bar{S}_i$ as the output.

C. CFTABS MODEL

In this research, the three proposed augmentation methods function to reduce the number of OOV grams in BS. They create a variant model, which is called augmented bigram-syllabification (ABS), with a new score formulated as

$$S_{abs}(w_i|w_{i-1}) = \begin{cases} B\frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} & \text{if } f_s(w_{i-1}w_i) > 0 \\ U\alpha\frac{f_s(w_i)}{N_s} & \text{otherwise,} \end{cases} \tag{4}$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the augmented- bigram and unigram frequencies happen in the normal trainset, w_i is the ith syllable (or unigram), $w_{i-1}w_i$ is a bigram that is composed from two syllables with indices of (i-1) and i, N_s represents the number of grams in the augmented training-set, B and U are the weights of augmented- bigram and unigram, respectively, and α is the backoff factor in Equation 3. As a bigram plays a more critical role in syllabifying a word than the unigram, the value of B should be much bigger than U.

Next, a combined trigram and augmented bigram syllabification (TABS) model is created using a score S_{tabs} formulated as

$$S_{tabs} = S_{ts} + \beta S_{abs} \tag{5}$$

where S_{ts} is the trigram-syllabification score in Equation (2) and S_{abs} is the augmented-bigram-syllabification score in Equation (4), and β is the weight that is used in the augmented-bigram score.

Finally, both TFO and TABS models are then combined to create CFTABS, which takes into account unigrams, bigrams, and trigrams (built from the original words), and both augmented-unigrams and augmented-bigrams (that are developed from the augmented words). The detailed explanation of CFTABS with a simple example of syllabifying a word is illustrated in Fig 2.

D. THREE AUGMENTATION METHODS

Since most Indonesian graphemes are pronounced as the same phonemes, some onset graphemes can be swapped based on the phoneme categorization in [45]. Table 1 illustrates some Indonesian graphemes and their swaps. There are 14 graphemes and their swaps, each of which is simply mapped into those phoneme categorizations. This mapping is possible due to their strong relation to the corresponding phonemes [45] [47]. A word that consists of one or more

TABLE 1. Graphemes and their swaps based on phoneme category

Phoneme category	Gr.	Sw,	Example
Plosive-Bilabial: {b, p}	b	p	***ba**.ku* (standard) → ***pa**.ku* (nail)
	p	b	***pe**.ri* (fairy) → ***be**.ri* (give)
Plosive-Dental: {d, t}	d	t	***de**.bu* (dust) → ***te**.bu* (cane)
	t	d	***ta**.yang* (show) → ***da**.yang* (court lady)
Plosive-Velar: {g, k}	g	k	***ga**.bung* (join) → ***ka**.bung* (mourning)
	k	g	***ka**.mis* (thursday) → ***ga**.mis* (clothes)
Affricative-Palatal: {c, j}	c	j	***ce**.ruk* (niche) → ***je**.ruk* (orange)
	j	c	***ja**.wat* (stretched out) → ***ca**.wat* (loincloth)
Fricative-Labiodental: {f, v}	f	v	***fo**.li* (thin metal) → ***vo**.li* (volley)
	v	f	***vi**.si* (vision) → ***fi**.si* (fission)
Fricative-Dental: {s, z}	s	z	*a.**sam*** (acid) → *a.**zam*** (aim)
	z	s	***ze**.ni* (soldier) → ***se**.ni* (art)
Thrill/Lateral-Dental: {l, r}	l	r	***lang**.ka* (rare) → ***rang**.ka* (frame)
	r	l	***ram**.bu* (sign) → ***lam**.bu* (canoe)

graphemes in Table Table 1 can be swapped to produce one or more other words.

For example, both ⟨b⟩ and ⟨p⟩ are pronounced as plosive-bilabial phonemes. In general, swapping ⟨b⟩ into ⟨p⟩ generates some new legal unigrams and bigrams, such as "***ba**.ku*" (standard) is swapped to be "***pa**.ku*" (nail), "***ba**.wang*" (onion) to be "***pa**.wang*" (handler), "***be**.ta*" (I am) to be "***pe**.ta*" (map), etc. Swapping the grapheme ⟨p⟩ into ⟨b⟩ also generally creates several new syllable unigrams and bigrams, such as "***pe**.ri*" (fairy) is swapped to be "***be**.ri*" (give), "***pa**.da*" (on) to be "***ba**.da*" (after), "***pi**.ta*" (tape) to be "***bi**.ta*" (bytes), etc.

The grapheme ⟨d⟩ is in the same category with grapheme ⟨t⟩, i.e. plosive-dental. Swapping the grapheme ⟨d⟩ into ⟨t⟩ generally produces some new legal syllable unigrams and bigrams, such as "***de**.bu*" (dust) is swapped to be "***te**.bu*" (cane), "***de**.bar*" (flutter) to be "***te**.bar*" (spread out), "***da**.ra*" (virgin) to be "***ta**.ra*" (the same level), etc. Swapping the grapheme ⟨t⟩ into ⟨d⟩ also generally creates several new syllable unigrams and bigrams, such as "***ta**.yang"* (show) can be swapped to be "***da**.yang*" (court lady), "***ta**.pa*" (asceticism) is swapped to be "***da**.pa*" (ransom slaves), "***ta**.rah*" (flat) to be "***da**.rah*" (blood), etc.

Both ⟨g⟩ and ⟨k⟩ are plosive-velar. Swapping grapheme ⟨g⟩ into ⟨k⟩ generates several legal unigrams and bigrams, such as "***ga**.bung*" (join) is swapped to be "***ka**.bung*" (mourning), "***ge**.tar*" (shakes) to be "***ke**.tar*" (daunted), "***gi**.la*" (crazy) to be "***ki**.la*" (stake), etc. Swapping the grapheme ⟨k⟩ into ⟨g⟩ also generally creates several new syllable unigrams and bigrams, such as "***ka**.mis*" (thursday) to be "***ga**.mis*" (clothes), "***ke**.mit*" (night guard) is swapped to be "***ge**.mit*" (poke), "***ka***.*ri*" (curry) to be "***ga***.*ri*" (handcuffs), etc.

The grapheme ⟨c⟩ is in the same category with grapheme ⟨j⟩, i.e. affricative-palatal. Swapping the grapheme ⟨c⟩ into ⟨j⟩ commonly produces some legal syllable unigrams and bigrams, such as "***ce***.*ruk*" (niche) is swapped to be "***je***.*ruk*" (orange), "***ca***.*ri*" (search) is swapped to be "***ja***.*ri*" (finger), "***ca***.*har*" (liquid) to be "***ja***.*har*" (loud), etc. Swapping the grapheme ⟨j⟩ into ⟨c⟩ also generally creates several new syllable unigrams and bigrams, such as "***ja***.*wat*" (stretched out) is swapped to be "***ca***.*wat*" (loincloth), "***ja***.*ra*" (small drill) to be "***ca***.*ra*" (way), "***ja***.*ran*" (horse) to be "***ca***.*ran*" (woman fighting), etc.

In [45], the authors state that Indonesian does not have phoneme /v/, where the grapheme ⟨v⟩ is always pronounced as phoneme /f/ so that it can be swapped into ⟨f⟩. For instance, the word "***fo***.*li*" (thin metal) is swapped as "***vo***.*li*" (volley) and "***vi***.*si*" (vision) is swapped as "***fi***.*si*" (fission).

Both graphemes ⟨s⟩ and ⟨z⟩ are in the same category: fricative-dental. In general, swapping the grapheme ⟨s⟩ into ⟨z⟩ generates some new legal syllable unigrams and bigrams, such as *"a.**sam**"* (acid) is swapped to be *"a.**zam**"* (aim). Swapping the grapheme ⟨z⟩ into ⟨s⟩ converts a word "***ze***.*ni*" (soldier) into "***se***.*ni*" (art).

Finally, both graphemes ⟨l⟩ and ⟨r⟩ are considerably in the same category: thrill/lateral-dental. In general, swapping the grapheme ⟨s⟩ into ⟨z⟩ generates some new legal syllable unigrams and bigrams, such as "***lang***.*ka*" (rare) is swapped to be "***rang***.*ka*" (frame). Swapping the grapheme ⟨z⟩ into ⟨s⟩ converts a word "***ram***.*bu*" (sign) into "***lam***.*bu*" (canoe).

However, swapping the graphemes does not always produce other formal words. Sometimes, it creates an illegal word. For instance, swapping grapheme "p" in the word "***pa***.*ha*" (thigh) creates a new word "***ba***.*ha*", which is illegal (OOV). But, the interesting fact is that the new word is a sub-word that comes from another word "***ba.ha***.*gi.a*" (happy). Of course, the swapped words enlarge the number of bigrams. Thus, the grapheme swapping can be a data augmentation method that is expected to enhance the score formulated in Equation (5) to give a better syllabification.

Furthermore, Table 2 shows some examples of augmented words that are built from two original words without changing the syllabification points. First, the original word "***be.ri***" is swapped for all combinations to produce three new words: "***be.li***" (buy), "***pe.ri***" (fairy), and "***pe.li***" (OOV). Both original and swapped words are then augmented using the flipping onsets to generate four OOV words: "***re.bi***", "***le.bi***", "***re.pi***", and "***le.pi***". Finally, all the original, swapped, and flipped words are augmented using transposing their nuclei to produce eight OOV words: "***bi.re***", "***bi.le***", "***pi.re***", "***pi.le***", "***ri.be***", "***li.be***", "***ri.pe***", and "***li.pe***". Thus, this original word is augmented to be 15 new words, where only two words are formally found in the Indonesian dictionary while the rests are OOV words. But, an interesting phenomenon is the OOV words can be some sub-words for many other formal words. For example, the OOV word "***pe.li***" is a sub-word that comes from other words: "***pe.li***.*as*" (spell), "***pe.li***.*cin*" (lubricant), "***pe.li***.*ta*" (light), etc. The second formal word "***ba.tu***"is also augmented into 15 new words, where seven words are formal and the rests are OOV words that can also be some sub-words for many other formal words. Therefore, no doubt that the augmented words are capable of increasing the number of both unigrams and bigrams, which is expected to make the $_{cbsps}$ score in Equation (5) more accurate.

TABLE 2. Example of some augmented words that are generated using combination of swapping consonants-graphemes, flipping onsets, and transposing nuclei in the original words without changing the points of syllabifications

Original word	Augmented words
be.ri (give)	***be.li*** (buy), ***pe.ri*** (fairy), ***pe.li*** (OOV), ***re.bi*** (OOV), ***le.bi*** (OOV), ***re.pi*** (OOV), ***le.pi*** (OOV), ***bi.re*** (OOV), ***bi.le*** (OOV), ***pi.re*** (OOV), ***pi.le*** (OOV), ***ri.be*** (OOV), ***li.be*** (OOV), ***ri.pe*** (OOV), ***li.pe*** (OOV)
ba.tu (stone)	***ba.du*** (checkered patterned), ***pa.tu*** (small pickaxe), ***pa.du*** (coherent), ***ta.bu***(taboo), ***da.bu*** (OOV), ***ta.pu*** (OOV), ***da.pu*** (OOV), ***bu.ta*** (blind), ***bu.da*** (OOV), ***pu.ta*** (OOV), ***pu.da*** (OOV), ***tu.ba*** (tube), ***du.ba*** (OOV), ***tu.pa*** (OOV), ***du.pa*** (incense)

IV. RESULT AND DISCUSSION

The three parameters of the proposed CFTABS: U, B, and β are jointly optimized using a fixed $\alpha = 0.4$ as suggested in [46]. The result is illustrated in Fig. 3. The optimum parameters are $U = 0.1$, $B = 100$, and $\beta = 0.75$ that give the lowest SER of 2.37%. This result proves the hypothesis explained in subsection III-C, where the value of B should be much bigger than U since bigram is more critical than unigram in syllabifying a word. The optimum $\beta = 0.75$ also makes sense because the percentage of legal bigrams produced by the proposed augmentation methods is 77.26%, as stated in Section II.

The CFTABS is then compared to five other models: BS [43], BFO [43], CBSPS [44], FkNNC [17], and BiLSTM-CNN-CRF [39]. Evaluation based on 5-FCV using a dataset consisting of 50 k Indonesian words explained in [17] [43] [44].

The result in Fig. 4 shows that CFTABS better than three other bigram-based models but worse than FkNNC and BiLSTM-CNN-CRF. It produces SER of 2.37% that is lower than BS, BFO, and CBSPS with average SER of 3.80%, 3.11%, and 2.61%, respectively. It means that CFTABS relatively reduces the average SER of BS by up to 37.63%. FkNNC gives a slightly lower SER of 2.27%. Meanwhile, BiLSTM-CNN-CRF reaches the lowest SER of 0.44%.

Based on the results, the proposed CFTABS is comparable to FkNNC. However, by offering a low complexity, it can be favored than FkNNC. It just computes the probabilities of tens or fewer candidates based on both original and augmented *n*-grams to decide a syllabification point. Meanwhile, FkNNC should: firstly, computes the dissimilarities between a candidate pattern of syllabification and the others in the trainset (up to 250 k patterns); secondly, chooses *k* neighbors in each class; finally, select the smallest dissimilarity to make

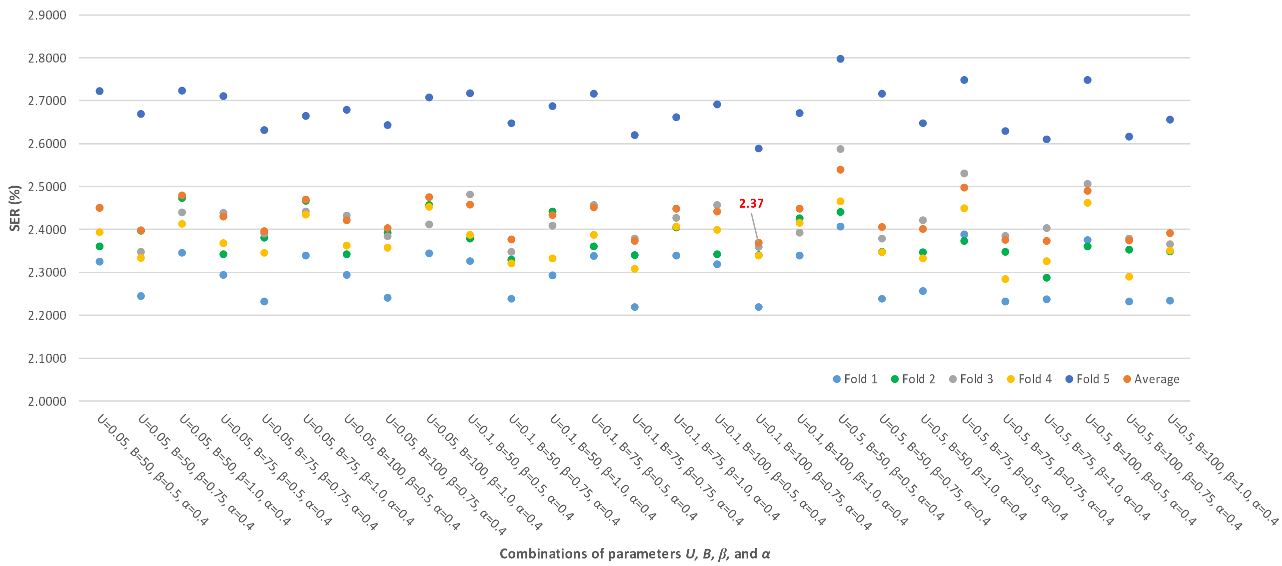

FIGURE 3. SERs produced by the proposed CFTABS using a fixed $\alpha = 0.4$ and jointly optimization of three parameters: U, B, and β.

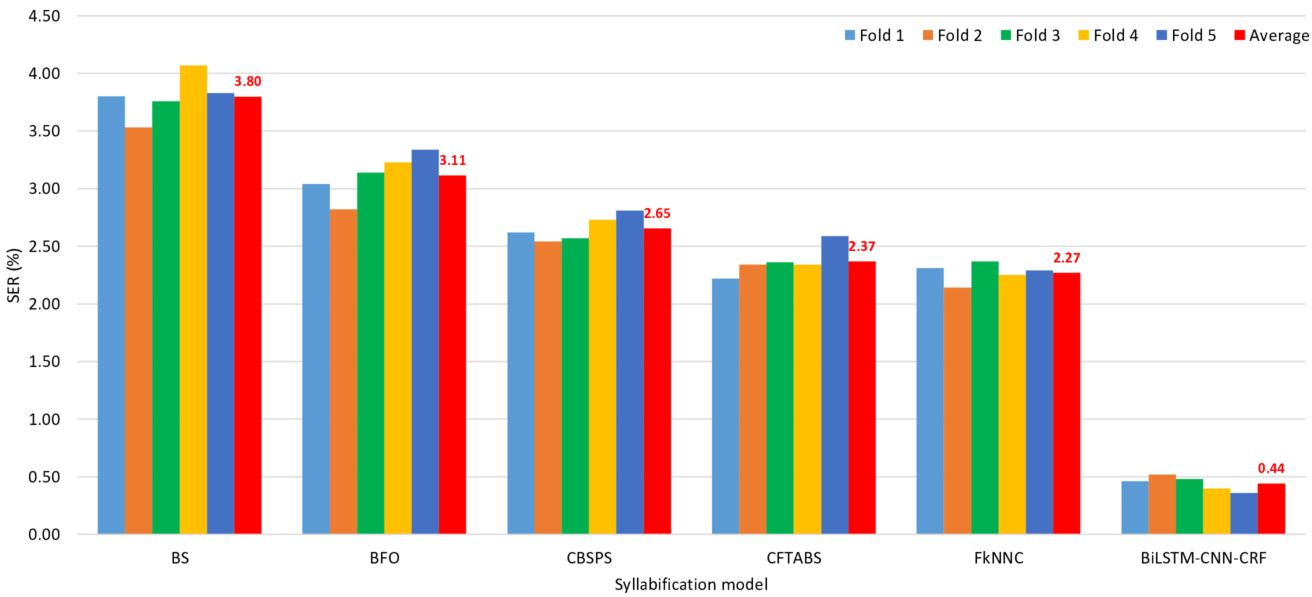

FIGURE 4. SERs produced by BS, BFO, CBSPS, CFTABS, and FkNNC for each fold and the average.

a decision. Compared to BiLSTM-CNN-CRF in terms of complexity, the proposed CFTABS is also better. It needs much lower training time (only ten minutes) than BiLSTM-CNN-CRF (up to ten hours).

However, the proposed CFTABS is a bit unstable. It produces low SERs for Fold 1 to Fold 4, but it gives a higher SER for Fold 5. A filtering procedure can be introduced to select the possible legal-bigrams. For instance, a swapped word "*zdlug.dul*" (OOV) that comes from the formal word "*struk.tur*" (structure) should be detected as an illegal bigram.

Besides, CFTABS also has difficulty to differentiates a diphthong from a regular sequence of grapheme and suffix since the input is a grapheme sequence (not phoneme sequence). For example, a diphthong ⟨ei⟩ is hard to be distinguished from a grapheme sequence of ⟨e⟩ and the suffix ⟨i⟩. The detailed investigation shows that most SER produced by this case since Indonesian has up to eighteen suffixes [45]. This problem can be solved by adding a procedure of diphthong recognition.

Another crucial problem is that CFTABS is applied on the syllable-level, which produces many OOV grams. Although three augmentation methods have been applied, the OOV rate is still high. Therefore, a grapheme-level is potentially applied to reduce the OOV rate. For instance, a word "*struktur*" (structure) just produces a syllable-level bigram of "*struk.tur*". But, it generates many grapheme-level bigrams, trigrams, until 8-gram: "*st*", "*tr*", "*ru*", ... "*struk.tur*". However, the use of a grapheme-level approach will make the complexity of the model slightly higher.

V. CONCLUSION

The proposed CFTABS is capable of improving the performance of the BS model, where the average SER is relatively decreased by up to 37.63%. It is comparable to the FkNNC-based syllabification and offers simplicity as well as flexibility since it just calculates the combined probabilities of both standards and augmented trigrams, bigrams, and unigrams to define the syllabification points accurately. Meanwhile, CFTABS gives a higher SER than BiLSTM-CNN-CFR, but it provides a faster training time. In the future, a particular procedure to filter legal bigrams and unigrams can be introduced to increase its performance. Another improvement can also be performed by using grapheme-level grams, instead of the syllable ones.

ACKNOWLEDGEMENTS

This research is funded by the Directorate of Research and Community Service PPM, Telkom University, with grant number: KWR.0067/PNLT3/PPM-LIT/2020.

REFERENCES

[1] S. Feng and T. Lee, "Exploiting Cross-Lingual Speaker and Phonetic Diversity for Unsupervised Subword Modeling," IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 27, no. 12, pp. 2000–2011, 2019.

[2] E. Pakoci, B. Popović, and D. Pekar, "Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition," Computational Intelligence and Neuroscience, vol. 2019, 2019.

[3] S. Geeta and B. L. Muralidhara, "Syllable as the basic unit for Kannada speech synthesis," in Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017. Institute of Electrical and Electronics Engineers Inc., pp. 1205–1208.

[4] Aripin, H. Haryanto, and S. Sumpeno, "A realistic visual speech synthesis for Indonesian using a combination of morphing viseme and syllable concatenation approach to support pronunciation learning," International Journal of Emerging Technologies in Learning, vol. 13, no. 8, pp. 19–37, 2018.

[5] D. Magdum and M. Suman, "System for identifying and correcting invalid words in the devanagari script for text to speech engine," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6 Special Issue 4, pp. 1001–1006, 2019.

[6] S. Ben Alex, B. P. Babu, and L. Mary, "Utterance and syllable level prosodic features for automatic emotion recognition," in 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018. Institute of Electrical and Electronics Engineers Inc., pp. 31–35.

[7] L. Sun, S. Fu, and F. Wang, "Decision tree SVM model with Fisher feature selection for speech emotion recognition," Eurasip Journal on Audio, Speech, and Music Processing, vol. 2019, no. 1, 2019.

[8] A. Leemann, M.-J. Kolly, F. Nolan, and Y. Li, "The role of segments and prosody in the identification of a speaker's dialect," Journal of Phonetics, pp. 69–84.

[9] S. Nayak, S. Bhati, and K. S. Rama Murty, "Zero Resource Speaking Rate Estimation from Change Point Detection of Syllable-like Units," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Institute of Electrical and Electronics Engineers Inc., pp. 6590–6594.

[10] D. O. Johnson and O. Kang, "Comparison of algorithms to divide noisy phone sequences into syllables for automatic unconstrained English speaking proficiency scoring," Artificial Intelligence Review, no. 3, pp. 1781–1804.

[11] O. Räsänen, S. Seshadri, J. Karadayi, E. Riebling, J. Bunce, A. Cristia, F. Metze, M. Casillas, C. Rosemberg, E. Bergelson, and M. Soderstrom, "Automatic word count estimation from daylong child-centered recordings in various language environments using language-independent syllabification of speech," Speech Communication, pp. 63–80.

[12] S. Suyanto, "Incorporating syllabification points into a model of grapheme-to-phoneme conversion," International Journal of Speech Technology, no. 2, pp. 459–470, jun.

[13] A. Rugchatjaroen, S. Saychum, S. Kongyoung, P. Chootrakool, S. Kasuriya, and C. Wutiwiwatchai, "Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion," Speech Communication, pp. 105–111.

[14] B. N. Budi, Nurtomo, and S. Suyanto, "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), no. 10, pp. 291–296.

[15] F. Alfiansyah and Suyanto, "Partial greedy algorithm to extract a minimum phonetically-and-prosodically rich sentence set," International Journal of Advanced Computer Science and Applications, no. 12, pp. 530–534.

[16] S. N. Hidayatullah and Suyanto, "Developing an adaptive language model for Bahasa Indonesia," International Journal of Advanced Computer Science and Applications, no. 1, pp. 488–492.

[17] E. A. Parande and S. Suyanto, "Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure," International Journal of Speech Technology, no. 1, pp. 13–20.

[18] S. Suyanto, S. Hartati, A. Harjoko, and D. V. Compernolle, "Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge," Speech Communication, pp. 109–118.

[19] C. R. Adsett, Y. Marchand, and V. Kešelj, "Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian," Computer Speech and Language, vol. 23, pp. 444–463, 2009.

[20] D. Balc, A. Beleiu, R. Potolea, and C. Lemnaru, "A learning-based approach for Romanian syllabification and stress assignment," in 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2015, pp. 37–42.

[21] W. Daelemans, A. Van Den Bosch, and T. Weijters, "IGTree: Using trees for compression and classification in lazy learning algorithms," Artificial Intelligence Review, vol. 11, no. 1-5, pp. 407–423, 1997.

[22] K. Müller, "Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpora training," in Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. ACL, 2001, pp. 410–417.

[23] A. Hunt, "Recurrent neural networks for syllabification," Speech Communication, no. 3, pp. 323–332.

[24] W. Daelemans and A. V. D. Bosch, "A neural network for hyphenation," in Proceedings of the International Conference on Artificial Neural Networks (ICANN–92), Brighton, United Kingdom, 1992, pp. 1647–1650 (vol. 2).

[25] T. Kristensen, "A neural network approach to hyphenating Norwegian," in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IEEE, 2000, pp. 148–153 vol.2.

[26] J. Tian, "Data-driven approaches for automatic detection of syllable boundaries," in Proceedings of the International Conference on Spoken Language Processing (ICSLP), 2004, pp. 61–64.

[27] S. Bartlett, G. Kondrak, and C. Cherry, "On the syllabification of phonemes," in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 308–316.

[28] T. H. Hlaing and Y. Mikami, "Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer," International Journal on Advances in ICT for Emerging Regions (ICTer), vol. 6, no. 2, pp. 2–9, 2014.

[29] G. A. Kiraz, M. Bernd, B. Labs, L. Technologies, and M. Hill, "Multilingual syllabification using weighted finite-state transducers," in Proceedings of the Third ESCA/COCOSDA Workshop on Speech Synthesis, 1998, pp. 59–64.

[30] K. Müller, "Improving syllabification models with phonotactic knowledge," in Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology - SIGPHON '06, 2006, pp. 11–20.
[31] J. Krantz, M. Dulin, P. De Palma, and M. VanDam, "Syllabification by Phone Categorization," in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ser. GECCO '18. New York, NY, USA: ACM, pp. 47–48.
[32] I. Ramli, N. Jamil, N. Seman, and N. Ardi, "An Improved Syllabification for a Better Malay Language Text-to- Speech Synthesis (TTS)," Procedia - Procedia Computer Science, no. Iris, pp. 417–424.
[33] H. Schmid, B. Möbius, and J. Weidenkaff, "Tagging syllable boundaries with joint n-gram models," Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 1, no. 1, pp. 49–52, 2007.
[34] L. G. Singh, L. Laitonjam, and S. R. Singh, "Automatic Syllabification for Manipuri language," in the 26th International Conference on Computational Linguistics, pp. 349–357.
[35] K. Rogova, K. Demuynck, and D. V. Compernolle, "Automatic syllabification using segmental conditional random fields," Computational Linguistics in the Netherlands Journal, pp. 34–48.
[36] T. Mayer, "Toward a totally unsupervised, language-independent method for the syllabification of written texts," in Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, 2010, pp. 63–71.
[37] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," in International Conference on Learning Representations, 2018.
[38] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in International Conference on Learning Representations, 2018.
[39] J. Krantz, M. Dulin, and P. De Palma, "Language-agnostic syllabification with neural sequence labeling," in Proceedings of 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019, pp. 804–810.
[40] R. Tang and J. Lin, "Progress and Tradeoffs in Neural Language Models," 2018.
[41] E. Shareghi, D. Gerz, I. Vulić, and A. Korhonen, "Show Some Love to Your n-grams: A Bit of Progress and Stronger n-gram Language Modeling Baselines," no. 2018, pp. 4113–4118, 2019.
[42] Y. Doval and C. Gómez-Rodríguez, "Comparing neural- and N-gram-based language models for word segmentation," Journal of the Association for Information Science and Technology, vol. 70, no. 2, pp. 187–197, 2019.
[43] S. Suyanto, "Flipping onsets to enhance syllabification," International Journal of Speech Technology, no. 4, pp. 1031–1038.
[44] ——, "Phonological similarity-based backoff smoothing to boost a bigram syllable boundary detection," International Journal of Speech Technology.
[45] H. Alwi, S. Darmowidjojo, H. Lapoliwa, and A. M. Moeliono, Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3rd ed. Jakarta: Balai Pustaka, 2014.
[46] T. Brants, A. C. Popat, and F. J. Och, "Large Language Models in Machine Translation," in The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, vol. 1, no. June, 2007, pp. 858–867.
[47] Suyanto and A. Harjoko, "Nearest neighbour-based Indonesian G2P conversion," Telkomnika (Telecommunication Computing Electronics and Control), vol. 12, no. 2, 2014.

SUYANTO received his B.Sc. on Informatics Engineering from STT Telkom (now: Telkom University), Bandung, Indonesia in 1998, the M.Sc. on Complex Adaptive Systems from Chalmers University of Technology, Goteborg, Sweden, in 2006, and the Ph.D. on Computer Science from Universitas Gadjah Mada in 2016. Since 2000, he joined STT Telkom as a lecturer in the School of Computing. His research interests include artificial intelligence, machine learning, deep learning, swarm intelligence, speech processing, and computational linguistics.

KEMAS M. LHAKSMANA received his bachelor degree in Informatics from Institut Teknologi Bandung, Bandung, Indonesia, in 2005, and earned master's degree in Information Systems Development from HAN University of Applied Sciences, Arnhem, Netherlands, in 2009. He received Ph.D. degree in Social Informatics from Kyoto University, Kyoto, Japan. He has been a lecturer of the Department of Software Engineering, Telkom University, Indonesia, since 2011. His research interest lies in text mining, services computing, and multiagent systems.

MOCH ARIF BIJAKSANA received the B.Sc. on Electrical Engineering from Gadjah Mada University, Indonesia in 1991, the M.Tech on Computer Science from RMIT University, Australia in 1999, and the Ph.D. on Computer Science from Queensland University of Technology (QUT), Australia in 2015. Since 2003, he joined Telkom University as a lecturer in the School of Computing. His research interests include deep learning and natural language processing.

ADRIANA KURNIAWAN received her B.Sc. on Information System from Politeknik Caltex Riau, Indonesia in 2016, and master's degree in Informatics from Telkom University, Bandung, Indonesia in 2020. Her research interests include artificial intelligence, machine learning, deep learning, speech processing, and computational linguistics.

• • •

Evidence of correspondence

Data Augmentation Methods for Low-Resource Orthographic Syllabification

1. First submission (19 July 2020)
2. LoA with Minor Revision (06 Augustus 2020)
3. Responses to Reviewers, Final submission (08 August 2020)
4. **Final paper of source latex submitted (08 August 2020)**
5. Creative Commons Attribution License (08 August 2020)

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

Final Files for Access-2020-35994 Successfully Submitted and Author Survey

IEEE Access <onbehalfof@manuscriptcentral.com> Sat, Aug 8, 2020 at 3:29 PM
Reply-To: m.kaur@ieee.org
To: suyanto@telkomuniversity.ac.id

08-Aug-2020

Dear Dr. Suyanto:

Thank you for submitting the final files of your manuscript entitled, "Data Augmentation Methods for Low-Resource Orthographic Syllabification". The documents are being transferred to the IEEE production staff for publication in IEEE Access. The early access version of your article will be posted on IEEE Xplore within 2-3 days.

You will be contacted by the production editor with galley proofs within the next two weeks. Additionally, you will be receiving an email in the next few days with instructions for accessing the IEEE Author Gateway, which is the online portal where you can track your manuscript through the production/publication process.

NEW: Once the copyright information is completed, within 3-5 business days, you will receive an email from Copyright Clearance Center (CCC) to log into their payment portal site and settle your balance by check, credit card, or wire transfer. If you need assistance with the site or payment process, please contact CCC Customer Service at: IEEESupport@copyright.com.

If you accidentally clicked out of the Electronic Copyright Form (ECF), you can access it by logging into your Author Center in ScholarOne Manuscripts at: http://mc.manuscriptcentral.com/ieee-access. Within the Author Center, please click on “Manuscripts with Decisions”, and underneath “Action” please click “Transfer Copyright.”

Thank you again for your fine contribution to IEEE Access.

Finally, to assist IEEE Access in maintaining high author satisfaction, we would appreciate if you could complete the following short survey regarding your publishing experience: https://research.ieee.org/jfe/form/SV_06AT9OOjXIDZoFv

Sincerely,

Miss Manpreet Kaur
IEEE Access

Evidence of correspondence

Data Augmentation Methods for Low-Resource Orthographic Syllabification

1. First submission (19 July 2020)
2. LoA with Minor Revision (06 Augustus 2020)
3. Responses to Reviewers, Final submission (08 August 2020)
4. Final paper of source latex submitted (08 August 2020)
5. **Creative Commons Attribution License (08 August 2020)**

A. By submitting your work to IEEE, you agree to comply with the IEEE Publication Services and Products Board Operations Manual (the "Operations Manual"), including, but not limited to, the specific provisions referenced herein(except to the extent any provision of the Operations Manual requires assignment of copyright in your work to IEEE).
B. Submission to this IEEE journal does not guarantee publication. By submitting your work to this journal you, as author, recognize that your work may be rejected for any reason. All submissions shall be reviewed by the Editor in accordance with section 8.2.2 of the Operations Manual.
C. Should your paper be rejected IEEE will not exercise any of the rights granted to it under the Creative Commons Attribution 4.0 Unported License.
D. IEEE takes intellectual property protection seriously and is opposed to plagiarism in any fashion. Accordingly, you consent to having your work submitted to a plagiarism detection tool and to be bound by IEEE policies concerning plagiarism and author misconduct.
E. IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. You must ensure that your work meets the requirements as stated in section 8.2.1 of the Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at https://www.ieee.org/publications/rights/author-rights-responsibilities.html.
F. You warrant that your work, including and any accompanying materials, is original and that you are the author of the work. To the extent your work incorporates text passages, figures, data or other material from the works of others, you represent and warrant that you have obtained all third party permissions and consents to grant the rights herein and have provided copies of such permissions and consents to IEEE. As stated in section 8.2.1B12 of the Operations Manual: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it."
G. You are advised of Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."
H. You agree that publication of a notice of violation as a corrective action for a confirmed case of plagiarism, as described in Section 8.2.4 of the IEEE PSPB Publications Operations Manual, does not violate any of your moral rights.
I. You agree to indemnify and hold IEEE and its parents, subsidiaries, affiliates, officers, employees, agents, partners and licensors harmless from any claim or demand, including reasonable attorneys' fees, due to or arising out of: (1) content you submit, post, transmit or otherwise make available through IEEE's publishing program; (2) your use of this IEEE journal; (3) your violation of these Terms of Use; or (4) your violation of any rights of another party.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON, YOU CERTIFY THAT SUCH ACTION CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE WITH UNITED STATES LAW, WHICH AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

Suyanto Suyanto
Signature

08-08-2020
Date

Questions about the submission of the form or manuscript must be sent to the publication's editor. Please direct all questions about IEEE copyright policy to:
IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966