Evidence of correspondence

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

1. **First submission with title "Slur Words, Boost Indonesian Bigram-Syllabification" (11 August 2019)**
2. **LoA with Major Revision (12 December 2019)**
3. **Response to Reviewers, Final submission with revised title "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" (04 January 2020)**
4. **LoA with Fully Accepted (07 January 2020)**
5. **Final Proof Reading (22 January 2020)**

Telkom University

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

IJST - Submission Confirmation

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Sun, Aug 11, 2019 at 6:18 PM
Reply-To: "International Journal of Speech Technology (IJST)" <ramya.thulasingam@springer.com>
To: Suyanto Suyanto <suyanto@telkomuniversity.ac.id>

Dear Dr. Suyanto,

Thank you for submitting your manuscript, Slur Words, Boost Indonesian Bigram-Syllabification, to International Journal of Speech Technology.

During the review process, you can keep track of the status of your manuscript by accessing the Editorial Manager website.

Your username is: suyanto
If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at https://www.editorialmanager.com/ijst/.

Should you require any further assistance please feel free to e-mail the Editorial Office by clicking on "Contact Us" in the menu bar at the top of the screen.

With kind regards,
Springer Journals Editorial Office
International Journal of Speech Technology

Now that your article will undergo the editorial and peer review process, it is the right time to think about publishing your article as open access. With open access your article will become freely available to anyone worldwide and you will easily comply with open access mandates. Springer's open access offering for this journal is called Open Choice (find more information on www.springer.com/openchoice). Once your article is accepted, you will be offered the option to publish through open access. So you might want to talk to your institution and funder now to see how payment could be organized; for an overview of available open access funding please go to www.springer.com/oafunding. Although for now you don't have to do anything, we would like to let you know about your upcoming options.

Recipients of this email are registered users within the Editorial Manager database for this journal. We will keep your information on file to use in the process of submitting, evaluating and publishing a manuscript. For more information on how we use your personal details please see our privacy policy at https://www.springernature.com/production-privacy-policy. If you no longer wish to receive messages from this journal or you have questions regarding database management, please contact the Publication Office at the link below.

In compliance with data protection regulations, you may request that we remove your personal registration details at any time. (Use the following URL: https://www.editorialmanager.com/ijst/login.asp?a=r). Please contact the publication office if you have any questions.

International Journal of Speech Technology

Slur Words, Boost Indonesian Bigram-Syllabification

--Manuscript Draft--

Manuscript Number:	
Full Title:	Slur Words, Boost Indonesian Bigram-Syllabification
Article Type:	Manuscript
Keywords:	Bahasa Indonesia; bigram; orthographic syllabification; slurring words
Corresponding Author:	Suyanto Suyanto, Dr. Telkom University Bandung, Jawa Barat INDONESIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Telkom University
Corresponding Author's Secondary Institution:	
First Author:	Suyanto Suyanto
First Author Secondary Information:	
Order of Authors:	Suyanto Suyanto
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	Evidence from child language acquisition is one of the important basis of syllable theories. Under five-year-old children commonly slur words by changing one or more consonants into other similar consonants based on both place and manner of articulations. The slurred words interestingly produce other words having different meanings but do not change the syllabification points. For examples, in Bahasa Indonesia, a word “ba.ra” (embers) is slurred to generate three new words: ”ba.la” (disaster), ”pa.ra” (rubber), and ”pa.la” (nutmeg) without changing the syllabification points since the graphemes <b> and <p> are in the same category of plosive-bilabial while <r> and <l> are thrill/lateral-dental. A preliminary study on 50k Indonesian words shows that slurring words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this research, a slurring procedure is proposed to boost a bigram-based orthographic syllabification that generally has a low performance for a dataset with many out-of-vocabulary bigrams. Some examinations using 5-fold cross-validations on the data-set of 50k Indonesian words prove that the proposed procedure is capable of increasing the performance of the standard bigram-syllabification, where the mean syllable error rate (SER) can be relatively decreased by up to 30.26%. Compare to the nearest neighbour-based syllabification, the proposed model is slightly worse but it gives lower complexity and high flexibility to be applied to named-entities.

Title Page containing all the Author Details

IJST manuscript No.
(will be inserted by the editor)

Slur Words, Boost Indonesian Bigram-Syllabification

Suyanto Suyanto

Received: date / Accepted: date

Suyanto Suyanto
School of Computing, Telkom University, Bandung, West Java 40257, Indonesia
Tel.: +62 22 7564108, Mobile: +62 812 845 12345
Orcid: https://orcid.org/0000-0002-8897-8091
E-mail: suyanto@telkomuniversity.ac.id

IJST manuscript No.
(will be inserted by the editor)

Slur Words, Boost Indonesian Bigram-Syllabification

Received: date / Accepted: date

Abstract Evidence from child language acquisition is one of the important basis of syllable theories. Under five-year-old children commonly slur words by changing one or more consonants into other similar consonants based on both place and manner of articulations. The slurred words interestingly produce other words having different meanings but do not change the syllabification points. For examples, in Bahasa Indonesia, a word "***ba.ra***" (embers) is slurred to generate three new words: "***ba.la***" (disaster), "***pa.ra***" (rubber), and "***pa.la***" (nutmeg) without changing the syllabification points since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. A preliminary study on 50k Indonesian words shows that slurring words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this research, a slurring procedure is proposed to boost a bigram-based orthographic syllabification that generally has a low performance for a dataset with many out-of-vocabulary bigrams. Some examinations using 5-fold cross-validations on the data-set of 50k Indonesian words prove that the proposed procedure is capable of increasing the performance of the standard bigram-syllabification, where the mean syllable error rate (SER) can be relatively decreased by up to 30.26%. Compare to the nearest neighbour-based syllabification, the proposed model is slightly worse but it gives lower complexity and high flexibility to be applied to named-entities.

Keywords Bahasa Indonesia · bigram · orthographic syllabification · slurring words

1 Introduction

One of the important pronunciation units in a language is syllable. It is strongly relevant to the rules of phonology. In [8], the researcher states that a syllable

is a representational unit used to learn the phonotactic constraints of speech-sounds. The syllable is generally believed to be central to the infant as well as the adult perception of speech [36]. The syllable theories are based on many evidence. One of them is evidence from child language acquisition [14].

In linguistic theory, a syllable consists of an obligatory nucleus with or without non-obligatory surrounding consonants called onset and coda [17]. The nucleus in Bahasa Indonesia can be a single vowel or a diphthong [3], [10]. Meanwhile, the onset and coda are consonants [3]. For instance, a word "*pantai*" (beach) contains two syllables: ⟨pan⟩ and ⟨tai⟩. The former consists of an onset ⟨p⟩, a nucleus of single vowel ⟨a⟩, and a coda ⟨n⟩. The later is composed of an onset ⟨t⟩, a nucleus of diphthong ⟨ai⟩, but no coda.

An automatic syllabification is defined as a process of dividing a word into syllables. This model is urgent for some researches as well as application developments in the field of linguistics, e.g. speech synthesis [35], [13], speech recognition [20], [46], [19], [27], [31], speech emotion recognition [33], [7], dialect identification [26], machine translation [25], spelling-checker [4], [32], information retrieval [21], grapheme-to-phoneme conversion [42], [38], etc.

The automatic syllabification is commonly implemented using two different approaches: either orthographic or phonemic-based. The previous works show that the phonemic-syllabification [44] performs better than the orthographic one [34], but it requires a linguist to provides perfect phoneme sequences. A model of phonemicization or grapheme-to-phoneme (G2P) can be created to replace the linguist role, but a small phoneme error rate (PER) decreases its performance in term of SER [44]. Besides, it potentially performs much worse for named-entities having many exceptions and ambiguities. Therefore, many researchers are interested in the orthographic (also known as graphemic) syllabification as it is much simpler and more flexible regarding the data-set for the learning process when the statistical models are used.

A syllabification is generally implemented using statistical models, instead of rule-based ones, since they are easier to implement and give lower SER [2]. The statistical models are usually implemented using either supervised or unsupervised learning technique, such as Näive Bayes model [5], decision tree-based model [5], [12], treebank model [29], random forest model [5], neural-based model [18] [11], [24], [45], support vector machine model [5], [6], finite-state transducers model [16], [22], context-free grammars model [30], hidden Markov model [23], syllabification by analogy [1], dropped-and-matched model [35], n-gram model [39], conditional random fields model [40], [37], nearest neighbour-based model [44], [34], and unsupervised-syllabification model based on a classification of graphemic-symbols into two categories: consonants and vowels [28].

The nearest neighbour is an interesting model since it produces a low SER [44], [34]. But, it has a high complexity of computation. It also requires complex language-specific knowledge. A graphemic encoding proposed in [34] produces a relatively high SER since it does not accurately represent a language-specific knowledge.

Another interesting model is the n-gram syllabification since it gives a competitive SER and a low complexity. Besides, it is simple to implement as well as language-independent that does not require any language-specific phonotactic knowledge. Unfortunately, it has a disadvantage for a small dataset with a high rate of out-of-vocabulary (OOV) bigrams. Many researchers have proposed various procedures to make some improvements. One of them is the segmental conditional random fields (SCRF) [37]. The SCRF is a bigram-based syllabification smoothed by a simple *Stupid Backoff* described in [9]. It performs excellent, with a high generalization, even for quite small training-sets. Unfortunately, it looks very complex since eight features generated by sonority, legality, and maximum onset are taken into account in calculating the bigram-probability.

Therefore, in this research, a new simple procedure of slurring words is proposed to boost the standard bigram-syllabification. This procedure is inspired by under five-year-old children slurring some words that contain one or more particular graphemes. For instance, an Indonesian word "***b****a.****r****a*" (embers) can be slurred to produce three new words: "***b****a.****l****a*" (disaster), "***p****a.****r****a*" (rubber), and "***p****a.****l****a*" (nutmeg) without changing the points of syllabifications since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. Slurring words obviously increase the number of bigrams, which means that the OOV rate can be reduced.

Bahasa Indonesia has eighteen suffixes [3]. An interesting phenomenon is slurring on a suffix generally produces another legal suffix, but also a few illegal ones (noises). For instance, slurring a suffix ⟨ber⟩ in "*be.ra.tu.ran*" (regular) produces another legal suffix ⟨per⟩ in "*pe.ra.tu.ran*" (rules). Slurring a suffix ⟨pe⟩ in "*pe.nam.pi.lan*" (performance) produces an OOV word "*be.nam.pi.lan*", but all syllables in the OOV word produce legal bigrams come from other words: "*be.nam*" (immerse); "*nam.pi*" that come from "*me.nam.pi*" (winnow), "*pe.nam.pi*" (shelter), "*pe.nam.pi.lan*" (performance), and some other words; and "*pi.lan*" that come from "*a.pi.lan*" (breastwork), "*kam.pi.lan*" (appearance), "*pi.pi.lan*" (flat), "*pe.nam.pi.lan*" (performance), and many other words. Slurring a suffix ⟨ter⟩ in "*ter.ba.wa*" (not deliberately taken away) produces illegal suffix ⟨der⟩ in an OOV word "*der.ba.wa*" with a bigram "*der.ba*" that is never found in 50 k words but, based on the Indonesian phonotactic rules, it is a legal bigram.

For English and other European languages, the procedure of slurring words may produce huge illegal syllable-unigrams and syllable-bigrams. However, for Bahasa Indonesia, the procedure creates more new legal syllable-unigrams and syllable-bigrams than the illegal ones. A preliminary study shows that 50 k Indonesian words produce a total of 161,981 legal syllable-unigrams. Slurring those 50 k words produces a total of 2,676,764 slurred syllable-unigrams, where 87.36% of them are legal unigrams those are the same as produced by the original words and the rest 12.64% are unseen syllable-unigrams. It means that the slurring procedure significantly increases the number of unigrams by 16.52 times (14.44 times legal unigrams and 2.08 times unseen unigrams). Furthermore, the 50 k words produce a total of 212,550 syllable-bigrams. Slurring

them produces a total of 3,317,292 slurred syllable-bigrams, where 77.45% are legal bigrams those are the same as produced by the original words and the rest 22.55% are considered unseen syllable-bigrams. It means that the slurring procedure impressively increases the number of bigrams up to 14.12 times (12.09 times legal bigrams and 2.03 times unseen unigrams). Those unseen syllable-unigrams and syllable-bigrams can be either legal or illegal based on the Indonesian phonotactic rules. However, it is not easy to classify them into both classes.

In this research, the impact of slurring words is investigated on an Indonesian orthographic syllabification. First, the standard bigram-syllabification (BS) smoothed by the *Stupid Backoff* is implemented. Next, the combined standard bigram-syllabification and slurring words (CBSS) is developed and then examined whether it is capable of improving the performance of BS in term of SER. Since it is not easy to detect the unseen syllable-unigrams and syllable-bigrams as legal or illegal, the CBSS is implemented using all of them (not just the legal ones). Hence, this research focuses on examining whether the proposed slurring procedure is capable of increasing the performance of the BS or not.

2 Research Method

The block diagram in Fig. 1 illustrates the training process of the model proposed in this research. It is a combination of standard and slurred bigram-syllabifications. A data-set of pairs of words and their corresponding syllabifications is used to develop a list of standard or normal syllables, a table of syllable-unigrams, and a table syllable-bigrams on the left side. It is also used to develop a list of slurred syllables, a table of slurred syllable-unigrams (slurred-unigrams), and a table of slurred syllable-bigrams (slurred-bigrams) on the right side. The generated tables of both normal and slurred syllable-unigrams and syllable-bigrams are then exploited in the testing process, as illustrated by Fig. 2, to maximize the final score to produce the best sequence of syllables that has the highest score.

The testing process in Fig. 2 illustrates an input sequence of graphemes ⟨*pandai*⟩ (smart) is quite hard to be syllabified since ⟨ai⟩ is a diphthong, not two independent vowels ⟨a⟩ and ⟨i⟩. First, three vowels {⟨a⟩, ⟨a⟩, and ⟨i⟩} contained in the grapheme sequence are detected in the positions {2, 5, 6}. A well known high accurate method called Sukhotin's algorithm proposed in [15] can be exploited to automatically detect vowels and diphthongs, but it is not used here. Instead, this research just uses the simple Indonesian typological knowledge explained in [3], where five graphemes {⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩, ⟨u⟩} can be single vowels; four grapheme sequences {⟨ai⟩, ⟨au⟩, ⟨ei⟩, ⟨oi⟩} may produce diphthongs; and other graphemes are considered as consonants.

Next, six possible syllabifications are generated, i.e. ⟨*pa.nda.i*⟩, ⟨*pan.da.i*⟩, ⟨*pa.ndai*⟩, ⟨*pan.dai*⟩, ⟨*pand.ai*⟩, and ⟨*pand.a.i*⟩, where two graphemes ⟨*ai*⟩ may produce a diphthong or two single vowels forming one or two nucleuses. The

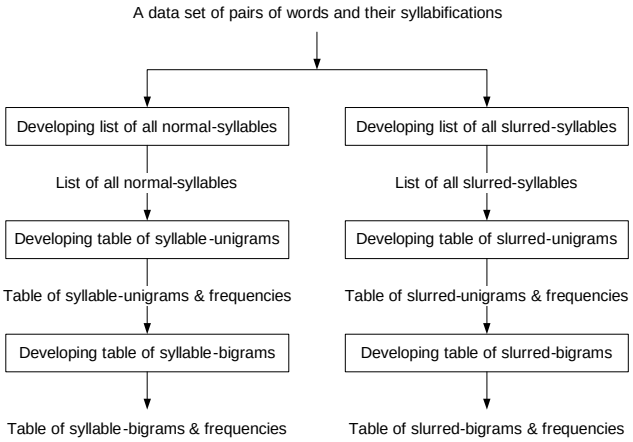


Fig. 1 Training process of the proposed combination of standard and slurred bigram-syllabifications (CBSS)

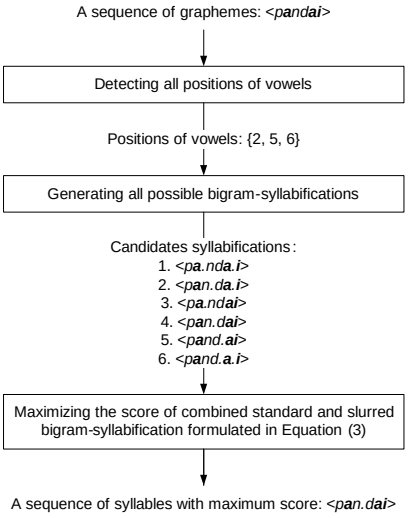


Fig. 2 Testing process of the proposed combination of standard and slurred bigram-syllabifications (CBSS)

score of each candidate is then calculated using the formula in Equation (3). In this case, the fourth candidate $\langle pan.dai \rangle$ gives the highest score since slurring this candidate produces two legal bigrams $\langle pan.tai \rangle$ (beach) and $\langle ban.tai \rangle$ (slaughter) as well as an OOV bigrams: $\langle ban.dai \rangle$ (it is not an Indonesian word) while the other five candidates produce all OOV bigrams. It means that CBSS is capable of syllabifying the input sequence of graphemes $\langle pandai \rangle$ into $\langle pan.dai \rangle$, where $\langle ai \rangle$ is correctly detected as a diphthong.

2.1 Standard bigram-syllabification model

A standard bigram-syllabification model (BS) works by maximizing the likelihoods of syllable sequences for a given word. The likelihood can be estimated using a probability chain, which is commonly smoothed by a simple *Stupid Backoff* to produce a more accurate probability, which is here called *score* since its value can be more than 1, for a training-set with many OOV words [9]. In this method, the score of bigram-syllabification S_{bs} is calculated as

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ are the frequencies of both syllable bigram and syllable unigram appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N is the training-set size, and α is the backoff factor that generally set to 0.4 [9]. The model of BS commonly gives a low performance for a small training-set that has a high rate of OOV syllable [37]. Besides using a smoothing procedure, the performance of BS can be improved by decreasing the OOV rate.

2.2 Combination of standard and slurred bigram-syllabification model

A procedure of slurring words in the training-set is proposed here to decrease the OOV rate in the BS model. Therefore, this procedure forms a new model called slurred-bigram-syllabification (SBS), which has a score S_{sbs} formulated as

$$S_{sbs}(w_i|w_{i-1}) = \begin{cases} B\frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} \text{ if } f_s(w_{i-1}w_i) > 0 \\ \\ U\alpha \frac{f_s(w_i)}{N_s} \text{ otherwise} \end{cases} \tag{2}$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the frequencies of both slurred-bigram and slurred-unigram appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N_s is the size of the slurred training-set, B is a weight of slurred-bigram, U is a weight of slurred-unigram, and α is the backoff factor as used in Equation 1. Both weights B and U are introduced here to smooth the score since the slurred words may produce some illegal bigrams and/or illegal unigrams. Hence, the value of B should be less than or equal 1.0 while the value of U is estimated to be much lower than B.

Finally, the combination of standard and slurred bigram-syllabification (CBSS) uses the score S_{cbss} that is simply stated as

$$S_{cbss} = S_{bs} + S_{sbs} \tag{3}$$

where S_{bs} is the score of bigram-syllabification in Equation (1) and S_{sbs} is the score of slurred-bigram-syllabification in Equation (2).

2.3 Slurred graphemes

Table 1 illustrates the graphemes and their slurs based on the categorization of phonemes (grapheme) described in [3] as well as their examples in some Indonesian formal words. Here, the graphemes and their slurs are simply mapped to those phoneme categorizations since they are strongly related to the corresponding phonemes [3] and [41]. A formal word containing one of those 14 graphemes, which are grouped into 7 categories, can be slurred to produce another formal word as shown in the last column (examples). In [3], both phonemes /g/ and /k/ are in the same category (plosive-velar), but they are not used here since slurring grapheme ⟨g⟩ into ⟨k⟩ commonly produces many illegal syllable-unigrams and syllable-bigrams, such as slurring a word "*me.mang.sa*" (prey on) generates "*me.mank.sa*" (OOV) with an illegal syllable-unigram "*mank*" and two illegal bigrams "*me.mank*" and "*mank.sa*". Instead, the grapheme ⟨q⟩ is used since, in Bahasa Indonesia, it is always pronounced as phoneme /k/ [3] and [43].

Table 1 Graphemes and their slurring as well as the example of the slurred words without changing their points or boundaries of syllabifications

Grapheme category	Graph.	Slur	Example
Plosive-Bilabial: {b, p}	b	p	***b****a.ru* (new) → ***p****a.ru* (lung)
	p	b	***p****a.du* (intact) → ***b****a.du* (checkered)
Plosive-Dental: {d, t}	d	t	***d****a.ri* (from) → ***t****a.ri* (dance)
	t	d	***t****a.hi* (from) → ***d****a.hi* (forehead)
Plosive-Velar: {k, q}	k	q	***k****a.ri* (curry) → ***q****a.ri* (reciter)
	q	k	*a.****q****i.dah* (creed) → *a.****k****i.dah* (creed)
Affricative-Palatal: {c, j}	c	j	***c****a.ri* (find) → ***j****a.ri* (finger)
	j	c	***j****an.da* (widow) → ***c****an.da* (joke)
Fricative-Labiodental: {f, v}	f	v	***f****i.si* (fission) → ***v****i.si* (vision)
	v	f	***v****o.li* (volley) → ***v****o.li* (thin metal)
Fricative-Dental: {s, z}	s	z	***s****a.man* (indict) → ***z****a.man* (era)
	z	s	*a.****z****am* (aim) → *a.****s****am* (acid)
Thrill/Lateral-Dental: {l, r}	l	r	***l****i.ma* (five) → ***r****i.ma* (rhyme)
	r	l	***r****a.bu* (Wednesday) → ***l****a.bu* (pumpkin)

Meanwhile, Table 2 illustrates the examples of some slurred words generated from the words containing two or more possible slurring-graphemes without changing the point or boundary of syllabification. A word "***b****a.****r****a*" (embers), which has two possible slurring graphemes ⟨b⟩ and ⟨r⟩, can be slurred to produce three other words, i.e. "***b****a.****l****a*" (disaster), "***p****a.****r****a*" (rubber), and "***p****a.****l****a*" (nutmeg) without changing the syllabification points. A word "***b****i.****r****u*" (blue), which has two possible slurring graphemes, can be slurred into three new words: "***b****i.****l****u*", "***p****i.****r****u*", and "***p****i.****l****u*" without changing the syllabification points. There is no formal word "***b****i.****l****u*" in Bahasa Indonesia (it means that

"***bi.lu***" is an OOV word), but it can be a sub-word for some other words, such as "*sem.**bi.lu***" (sharp reed skin like a knife). The word "***pi.ru***" is also an OOV word but it is a sub-word for the word "***pi.ru**.et*" (one of ballet dance styles). In contrast, the word "***pi.lu***" is a formal word that means "really sad" in English. No doubt, such slurred words increase the number of bigrams. Hence, slurring word can be seen as a method of data augmentation. This is expected to produce a more accurate score in Equation (3) so that a better syllabification can be achieved.

Table 2 Examples of some new words slurred from the words containing two or more possible slurring-graphemes without changing the point or boundary of syllabification

Word	Slurred words
ba.ra (embers)	*ba.la* (disaster), *pa.ra* (rubber), *pa.la* (nutmeg)
ba.ru (new)	*ba.lu* (widower), *pa.ru* (lung), *ba.lu* (hammer)
bi.ru (blue)	*pi.ru* (OOV), *bi.lu* (OOV), *pi.lu* (really sad)
ba.rat (west)	*ba.rad* (OOV), *ba.lat* (OOV), *ba.lad* (city), *pa.rat* (OOV), *pa.rad* (OOV), *pa.lat* (penis), *pa.lad* (OOV)
ca.ri (find)	*ca.li* (OOV), *ja.ri* (finger), *ja.li* (real)
ce.ri.ta (story)	*ce.ri.da* (OOV), *ce.li.ta* (OOV), *ce.li.da* (OOV), *je.ri.ta* (OOV), *je.ri.da* (OOV), *je.li.ta* (very beautiful), *je.li.da* (OOV)

3 Result and Discussion

The data-set used here is the same as described in [34]. It consists of 50k words equipped with boundaries or points of syllabifications. It is equally divided into five subsets (folds), where each subset consists of 10k words, to do the five-fold cross-validation. Three experiments are conducted in this research to sequentially tune the parameters. Firstly, the optimum unigram weight U is searched using $\alpha = 0.4$ as suggested in [9] and $B = 1.0$ based on an assumption that the slurred-bigrams have the same importance as the normal-bigrams. The bigram weight B is then optimized using the found optimum U and $\alpha = 0.4$. Next, the backoff factor α is verified using both optimum values of U and B. Finally, CBSS is compared to another syllabification model. Here, a percentage of errors in the syllable level, which is commonly known as SER, is used to measure all performances in those experiments.

3.1 Optimizing unigram weight U

The CBSS is firstly evaluated using $\alpha = 0.4$ and $B = 1.0$ to find the optimum unigram weight U. The results illustrated in Fig. 3 informs that U is very sensitive. A very small $U = 0.001$ produces high SERs for all folds. A big $U = 0.1$ or bigger also gives higher SERs. The unigram weight U reaches the optimum value of 0.05 that produces the lowest SERs for all folds with the average SER of 2.65%. As hypothesized, the optimum value of this parameter

is very low of only 0.05 (much lower than B), which means that the impact of the slurred unigrams is just 5% to do the syllabification.

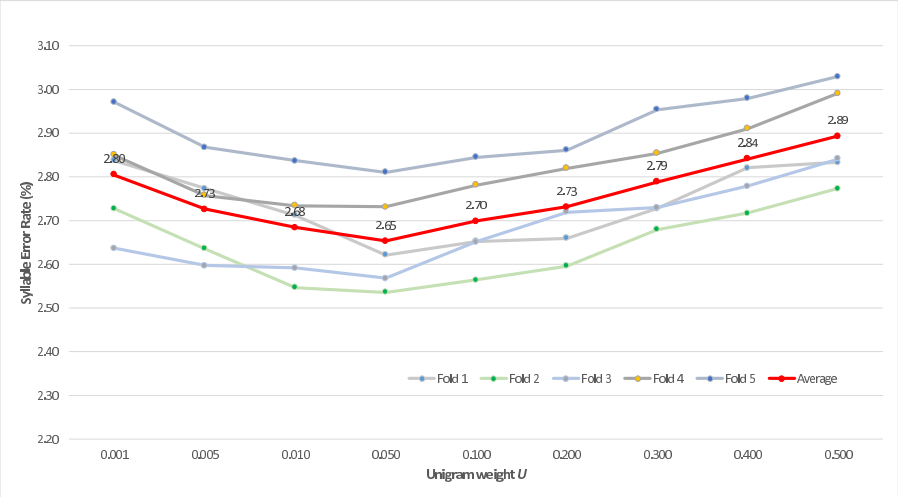


Fig. 3 SERs produced by CBSS using $\alpha = 0.4$, $B = 1.0$, and varying unigram weight U

3.2 Optimizing bigram weight B

The CBSS is then evaluated using $\alpha = 0.4$ and $U = 0.05$ to optimize the bigram weight B. The results in Fig. 3 shows that B is not sensitive. It is quite stable to produce low SERs for all folds when the value is in the interval of 0.8 to 1.1. It reaches the optimum value of 1.0 that produces the lowest average SER of 2.65%.

3.3 Verifying backoff factor α

Next, the use of $\alpha = 0.4$ suggested in [9] is verified using both optimum values $U = 0.05$ and B=1.0. Here, nine experiments are performed using $\alpha = 0.1$ to 0.9. The results in Fig. 5 informs that α is an easily tuned parameter. It gives the lowest average SER of 2.65% when the value is in the interval of 0.2 to 0.4. It means that the $\alpha = 0.4$ is verified in this research.

3.4 Comparison to other models

Finally, the best performance of CBSS is compared to two other syllabification models: BS and fuzzy k-nearest neighbour in every class (FkNNC), which uses

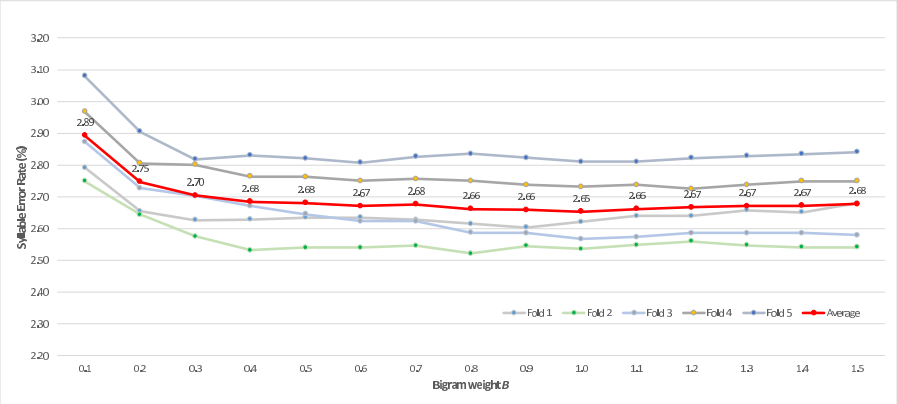


Fig. 4 SERs produced by CBSS using $\alpha = 0.4$, $U = 0.05$, and varying bigram weight B

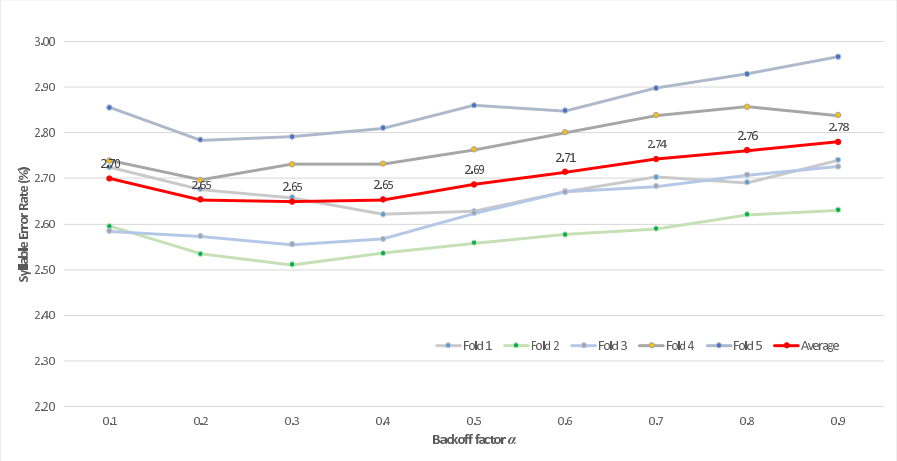


Fig. 5 SERs produced by CBSS using $B = 1.0$, $U = 0.05$, and varying α

the same data-set of 50 k words described in [34]. The same testing-sets of 5 folds are also used in this evaluation. To get fairness, all methods are compared in their best performances using the same dataset as illustrated by Fig. 6. It can be seen that CBSS produce lower average SER of 2.65% than BS with average SER of 3.80%. It means that CBSS relatively decrease the SER by up to 30.26%. It shows that the proposed slurring procedure is capable of boosting the performance of standard bigram-syllabification model. But, CBSS gives a slightly higher average SER than FkNNC that gives average SER of 2.27%. Nevertheless, CBSS has lower complexity than FkNNC since it just calculates the probabilities of bigrams to get the syllabification points while FkNNC should find the k nearest neighbours and then decide the syllabification points.

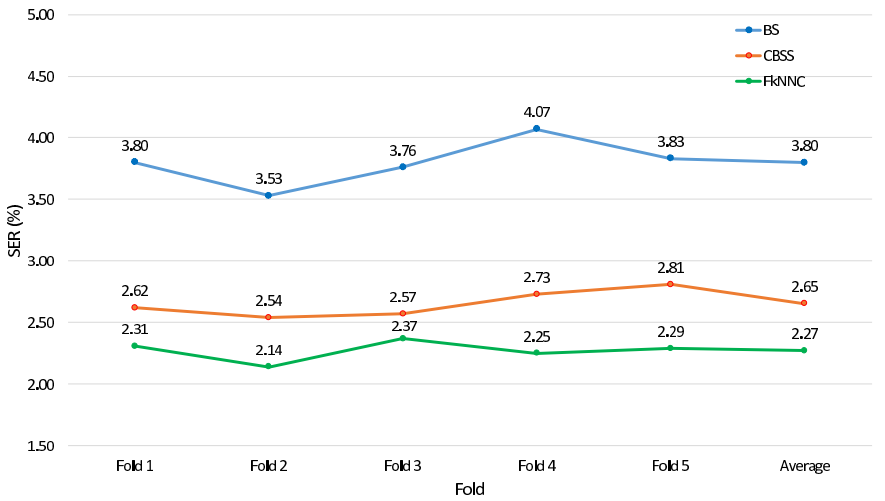


Fig. 6 SERs produced by three syllabification models: BS, CBSS, and FkNNC

3.5 Disadvantage of the proposed model

Since the input is a raw sequence of grapheme (not phoneme), CBSS has difficulty to distinguish a diphthong from a regular sequence of grapheme and suffix, such as a diphthong ⟨ai⟩ is hard to be differentiated from a regular sequence of grapheme ⟨a⟩ and suffix ⟨i⟩. For instance, a root "*intai*" (lurking) that is syllabified as ⟨in.tai⟩ while a derivative "*memintai*" (ask for) is segmented into ⟨me.min.ta.i⟩ since it is derived from a root ⟨min.ta⟩ that is prefixed by ⟨me⟩ and suffixed by ⟨i⟩. The former ⟨ai⟩ is a diphthong but the later is a sequence of grapheme ⟨a⟩ and suffix ⟨i⟩. The syllable errors produced by the proposed model mostly come from such case as Bahasa Indonesia has up to eighteen suffixes [3]. Such weakness probably can be overcome by incorporating a scheme of diphthong detection.

3.6 Possibility to be applied to named-entity

The proposed CBSS just uses bigrams and unigrams as well as their slurs to maximize the scores of syllabifications. This can be concluded that it can be applied to a data-set of name-entities since the slurring procedure is very common in such data-set. For example, slurring a named-entity "**b**an.**d**ung" (the capital city in West Java) produces three other name-entities, i.e. "**b**an.**t**ung" (a resort in Sukhothai, Thailand), "**p**an.**d**ung" (a village in Special Region of Yogyakarta), and "**p**an.**t**ung" (a folk song from Bolaang Mongondow, North Sulawesi).

4 Conclusion

The proposed slurring procedure is capable of boosting the standard bigram-syllabification model. It relatively reduces the average SER up to 30.26%. The performance is slightly worse than the FkNNC-based syllabification but it offers much lower complexity since it just calculates the probabilities of bigrams and unigrams to get the syllabification points. Besides, it is prospective to be applied to name-entities. In the future, a scheme of filtering possible legal-bigrams can be introduced to improve its performance.

Acknowledgements

I would like to thank my wife, Ari Virgandini, as well as my sons, Muhammad Arkan Ariyanto and Muhammad Agha Ariyanto, for the great inspirations of slurring your words.

References

1. Adsett, C.R., Marchand, Y.: A comparison of data-driven automatic syllabification methods. In: Proceedings of the 16th International Symposium on String Processing and Information Retrieval (SPIRE), pp. 174–181. Springer Berlin Heidelberg (2009). DOI 10.1007/978-3-642-03784-9
2. Adsett, C.R., Marchand, Y., Kešelj, V.: Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian. Computer Speech and Language **23**, 444–463 (2009). DOI 10.1016/j.csl.2009.02.004
3. Alwi, H., Dardjowidjojo, S., Lapoliwa, H., Moeliono, A.M.: Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3 edn. Balai Pustaka, Jakarta (1998)
4. Alyan, R., Günel, K., Yakhno, T.: Detecting Misspelled Words in Turkish Text Using Syllable n-gram Frequencies. In: The 2nd international conference on Pattern recognition and machine intelligence, October (2007). DOI 10.1007/978-3-540-77046-6
5. Balc, D., Beleiu, A., Potolea, R., Lemnaru, C.: A learning-based approach for Romanian syllabification and stress assignment. In: 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 37–42 (2015). DOI 10.1109/ICCP.2015.7312603
6. Bartlett, S., Kondrak, G., Cherry, C.: 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, pp. 308–316. Boulder, Colorado (2009). DOI 10.3115/1620754.1620799
7. Ben Alex, S., Babu, B.P., Mary, L.: Utterance and syllable level prosodic features for automatic emotion recognition. In: 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018, pp. 31–35. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/RAICS.2018.8635059. URL https://ieeexplore.ieee.org/document/8635059
8. Bernard, A.: An onset is an onset: Evidence from abstraction of newly-learned phonotactic constraints. Journal of Memory and Language **78**, 18–32 (2015). DOI https://doi.org/10.1016/j.jml.2014.09.001. URL http://www.sciencedirect.com/science/article/pii/S0749596X1400103X
9. Brants, T., Popat, A.C., Och, F.J.: 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, pp. 858–867 (2007)
10. Chaer, A.: Fonologi Bahasa Indonesia (Indonesian Phonology). Rineka Cipta, Jakarta (2009)

11. Daelemans, W., Bosch, A.V.D.: A neural network for hyphenation. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN92), pp. 1647–1650 (vol. 2). Brighton, United Kingdom (1992). DOI 10.1016/B978-0-444-89488-5.50176-7
12. Daelemans, W., Bosch, A.V.D., Weijters, T.: IGTree: Using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review **11**(1-5), 407–423 (1997). DOI 10.1.1.29.4517
13. Faldessai, N., Pawar, J., Naik, G.: Syllabification: An effective approach for a TTS system for Konkani. In: 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, ICEECCOT 2016, pp. 161–167. Institute of Electrical and Electronics Engineers Inc. (2017). DOI 10.1109/ICEECCOT.2016.7955207. URL https://ieeexplore.ieee.org/document/7955207
14. Fallows, D.: Experimental evidence for English syllabification and syllable structure. Journal of Linguistics **17**(2), 309–317 (1981). DOI 10.1017/S0022226700007027
15. Foster, C.C.: A Comparison of Vowel Identification Methods. Cryptologia **16**(3), 282–286 (1992). DOI 10.1080/0161-119291866955. URL https://doi.org/10.1080/0161-119291866955
16. Hlaing, T.H., Mikami, Y.: Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer. International Journal on Advances in ICT for Emerging Regions (ICTer) **6**(2), 2–9 (2014). DOI 10.4038/icter.v6i2.7150
17. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development (2001)
18. Hunt, A.: Recurrent neural networks for syllabification. Speech Communication **13**(3), 323–332 (1993). DOI https://doi.org/10.1016/0167-6393(93)90031-F. URL http://www.sciencedirect.com/science/article/pii/016763939390031F
19. Janakiraman, R., Kumar, J.C., Murthy, H.A.: Robust syllable segmentation and its application to syllable-centric continuous speech recognition. In: National Conference on Communications (NCC), pp. 1–5. Joint Telematics Group of IITs {&} IISc, Chennai, India (2010). DOI 10.1109/NCC.2010.5430189
20. Kamper, H., Jansen, A., Goldwater, S.: A segmental framework for fully-unsupervised large-vocabulary speech recognition. Computer Speech & Language **46**, 154–174 (2017). DOI https://doi.org/10.1016/j.csl.2017.04.008. URL http://www.sciencedirect.com/science/article/pii/S0885230816301905
21. Kettunen, K., McNamee, P., Baskaya, F.: Using Syllables As Indexing Terms in Full-Text Information Retrieval. In: Baltic HLT (2010). DOI 10.3233/978-1-60750-641-6-225
22. Kiraz, G.A., Bernd, M., Labs, B., Technologies, L., Hill, M.: Multilingual syllabification using weighted finite-state transducers. In: Proceedings of the Third ESCA/COCOSDA Workshop on Speech Synthesis, pp. 59–64 (1998)
23. Krantz, J., Dulin, M., De Palma, P., VanDam, M.: Syllabification by Phone Categorization. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, pp. 47–48. ACM, New York, NY, USA (2018). DOI 10.1145/3205651.3208781. URL http://doi.acm.org/10.1145/3205651.3208781
24. Kristensen, T.: A neural network approach to hyphenating Norwegian. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, pp. 148–153 vol.2. IEEE (2000). DOI 10.1109/IJCNN.2000.857889
25. Kunchukuttan, A., Bhattacharyya, P.: Orthographic Syllable as basic unit for SMT between Related Languages. CoRR (2016). URL http://arxiv.org/abs/1610.00634
26. Leemann, A., Kolly, M.J., Nolan, F., Li, Y.: The role of segments and prosody in the identification of a speaker's dialect. Journal of Phonetics **68**, 69–84 (2018). DOI https://doi.org/10.1016/j.wocn.2018.02.001. URL http://www.sciencedirect.com/science/article/pii/S0095447016300365
27. Majewski, P.: Syllable Based Language Model for Large Vocabulary Continuous Speech Recognition of Polish. In: P. Sojka, A. Horák, I. Kopeček, K. Pala (eds.) Text, Speech and Dialogue, pp. 397–401. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
28. Mayer, T.: 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, pp. 63–71 (2010)

29. Müller, K.: 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, pp. 410–417. ACL (2001)
30. Müller, K.: 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, pp. 11–20 (2006). DOI 10.3115/1622165.1622167
31. Nayak, S., Bhati, S., Rama Murty, K.S.: 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, pp. 6590–6594. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/ICASSP.2019.8683462. URL https://ieeexplore.ieee.org/document/8683462
32. Oncevay-marcos, A.: Spell-Checking based on Syllabification and Character-level Graphs for a Peruvian Agglutinative Language. In: The First Workshop on Subword and Character Level Models in NLP, pp. 109–116 (2017)
33. Origlia, A., Cutugno, F., Galatà, V.: Continuous emotion recognition with phonetic syllables. Speech Communication **57**, 155–169 (2014). DOI https://doi.org/10.1016/j.specom.2013.09.012. URL http://www.sciencedirect.com/science/article/pii/S0167639313001337
34. Parande, E.A.: Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. International Journal of Speech Technology **22**(1), 13–20 (2019). DOI 10.1007/s10772-018-09569-3. URL https://link.springer.com/article/10.1007/s10772-018-09569-3
35. Ramli, I., Jamil, N., Seman, N., Ardi, N.: An Improved Syllabification for a Better Malay Language Text-to- Speech Synthesis (TTS). Procedia - Procedia Computer Science **76**(Iris), 417–424 (2015). DOI 10.1016/j.procs.2015.12.280. URL http://dx.doi.org/10.1016/j.procs.2015.12.280
36. Räsänen, O., Doyle, G., Frank, M.C.: Pre-linguistic segmentation of speech into syllable-like units. Cognition **171**, 130–150 (2018). DOI https://doi.org/10.1016/j.cognition.2017.11.003. URL http://www.sciencedirect.com/science/article/pii/S0010027717302901
37. Rogova, K., Demuynck, K., Compernolle, D.V.: Automatic syllabification using segmental conditional random fields. Computational Linguistics in the Netherlands Journal **3**, 34–48 (2013). URL https://clinjournal.org/clinj/article/view/24/20
38. Rugchatjaroen, A., Saychum, S., Kongyoung, S., Chootrakool, P., Kasuriya, S., Wutiwiwatchai, C.: Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion. Speech Communication **106**, 105–111 (2019). DOI https://doi.org/10.1016/j.specom.2018.12.003. URL http://www.sciencedirect.com/science/article/pii/S0167639317303965
39. Schmid, H., Möbius, B., Weidenkaff, J.: Tagging syllable boundaries with joint n-gram models. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH **1**(1), 49–52 (2007). URL https://www.cis.uni-muenchen.de/ schmid/papers/syltagger.pdf
40. Singh, L.G., Laitonjam, L., Singh, S.R.: Automatic Syllabification for Manipuri language. In: the 26th International Conference on Computational Linguistics, pp. 349–357 (2016). URL https://www.aclweb.org/anthology/papers/C/C16/C16-1034/
41. Suyanto, Harjoko, A.: Nearest neighbour-based Indonesian G2P conversion. Telkomnika (Telecommunication, Computing, Electronics, and Control) **12**(2), 389–396 (2014). DOI http://dx.doi.org/10.12928/telkomnika.v12i2.57. URL http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/57
42. Suyanto, S.: Incorporating syllabification points into a model of grapheme-to-phoneme conversion. International Journal of Speech Technology **22**(2), 459–470 (2019). DOI 10.1007/s10772-019-09619-4. URL https://link.springer.com/article/10.1007/s10772-019-09619-4
43. Suyanto, S., Hartati, S., Harjoko, A.: Modified Grapheme Encoding and Phonemic Rule to Improve PNNR-Based Indonesian G2P. International Journal of Advanced Computer Science and Applications (IJACSA) **7**(3), 430–435 (2016). DOI 10.14569/IJACSA.2016.070358. URL https://thesai.org/Downloads/Volume7No3/Paper_58-Modified_Grapheme_Encoding_and_Phonemic_Rule.pdf

44. Suyanto, S., Hartati, S., Harjoko, A., Compernolle, D.V.: Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge. Speech Communication **85**, 109–118 (2016). DOI 10.1016/j.specom.2016.10.009. URL http://dx.doi.org/10.1016/j.specom.2016.10.009
45. Tian, J.: Data-driven approaches for automatic detection of syllable boundaries. In: Proceedings of the International Conference on Spoken Language Processing (ICSLP), pp. 61–64 (2004)
46. Wu, S.L.W.S.L., Shire, M., Greenberg, S., Morgan, N.: Integrating syllable boundary information into speech recognition. In: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 987–990 (1997). DOI 10.1109/ICASSP.1997.596105

Evidence of correspondence

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

1. First submission with title "Slur Words, Boost Indonesian Bigram-Syllabification" (11 August 2019)
2. **LoA with Major Revision (12 December 2019)**
3. Response to Reviewers, Final submission with revised title "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" (04 January 2020)
4. LoA with Fully Accepted (07 January 2020)
5. Final Proof Reading (22 January 2020)

Decision on your manuscript #IJST-D-19-00125

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Thu, Dec 12, 2019 at 8:43 AM
Reply-To: "International Journal of Speech Technology (IJST)" <ramya.thulasingam@springer.com>
To: Suyanto Suyanto <suyanto@telkomuniversity.ac.id>

Dear Dr. Suyanto:

We have received the reports from our advisors on your manuscript, "Slur Words, Boost Indonesian Bigram-Syllabification", which you submitted to International Journal of Speech Technology.

Based on the advice received, the Editor feels that your manuscript could be reconsidered for publication should you be prepared to incorporate major revisions. When preparing your revised manuscript, you are asked to carefully consider the reviewer comments which are attached, and submit a list of responses to the comments. Your list of responses should be uploaded as a file in addition to your revised manuscript.

In order to submit your revised manuscript electronically, please access the Editorial Manager website.

Your username is: suyanto
If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at https://www.editorialmanager.com/ijst/.

Please click "Author Login" to submit your revision.

We look forward to receiving your revised manuscript.

Best regards,
Amy Neustein, Ph.D.
Editor-in-Chief
International Journal of Speech Technology

COMMENTS FOR THE AUTHOR:

Reviewer #1: the paper is well organized and its English is very good. The topic covered is also a good one and of great value for our community.
I prefer that the author/s to use many datasets instead of one. this would give better comparisons and results.
it is not clear the justification of using the indicated equations.
instead of indicating the Disadvantages of the proposed model, I prefer to find a way to solve these deficiencies.
in general, the work is good

Reviewer #2: While observations of language acquisition may have motivated this research it is not germane to the manuscript and may be misleading in the abstract. I am not sure "slur" is the best way to describe the transform or mapping operation performed. In Section 1 the term "suffix" appears to be applied to what is actually a prefix. Further detail should be provided about the slurring process itself. In Sections 3.1-3.3 it would be preferable to optimize the parameters jointly given the modest additional computational cost. In Section 3.4 the additional complexity of FkNNC should be quantified. In general I might characterize this approach as a back-off procedure based on phonological similarity with known bigrams.

Evidence of correspondence

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

1. First submission with title "Slur Words, Boost Indonesian Bigram-Syllabification" (11 August 2019)
2. LoA with Major Revision (12 December 2019)
3. **Response to Reviewers, Final submission with revised title "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" (04 January 2020)**
4. LoA with Fully Accepted (07 January 2020)
5. Final Proof Reading (22 January 2020)

Reviewer #1:

The paper is well organized and its English is very good. The topic covered is also a good one and of great value for our community. I prefer that the author/s to use many datasets instead of one. this would give better comparisons and results. It is not clear the justification of using the indicated equations. Instead of indicating the disadvantages of the proposed model, I prefer to find a way to solve these deficiencies. in general, the work is good.

>> Two datasets, i.e. a dataset of 15k named-entities and a mixed dataset of formal world and named-entities, are now added to evaluate the performance of the proposed model.

>> The justification is now explained more detail and clearer as follows:

Hence, the value of *B* should be less than 1.0 because swapping procedure on 50k formal words creates up to 22.55% illegal bigrams. Meanwhile, the value of *U* is probably much lower than *B* since a unigram is less important than a bigram in deciding the score of syllabification.

>> The subsection indicating the disadvantages of the proposed model is now removed since quite hard to find a simple way to solve the limitations of the proposed model.

Reviewer #2: While observations of language acquisition may have motivated this research it is not germane to the manuscript and may be misleading in the abstract. I am not sure "slur" is the best way to describe the transform or mapping operation performed. In Section 1 the term "suffix" appears to be applied to what is actually a prefix. Further detail should be provided about the slurring process itself. In Sections 3.1-3.3 it would be preferable to optimize the parameters jointly given the modest additional computational cost. In Section 3.4 the additional complexity of FkNNC should be quantified. In general I might characterize this approach as a back-off procedure based on phonological similarity with known bigrams.

>> The sentences related to the observations of language acquisition are now removed in the Abstract.

>> All terms "suffix" in Section 1 are now replaced by "prefix".

>> The three parameters are now jointly optimized

>> The additional complexity of FkNNC is now quantified.

>> To match the characteristic of the proposed procedure, the title is now revised to be “Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection”.

International Journal of Speech Technology

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

--Manuscript Draft--

Manuscript Number:	IJST-D-19-00125R1
Full Title:	Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection
Article Type:	Manuscript
Keywords:	backoff smoothing; bigram; Indonesian language; orthographic syllabification; phonological similarity
Corresponding Author:	Suyanto Suyanto, Dr. Telkom University Bandung, Jawa Barat INDONESIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Telkom University
Corresponding Author's Secondary Institution:	
First Author:	Suyanto Suyanto
First Author Secondary Information:	
Order of Authors:	Suyanto Suyanto
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	Swapping one or more consonant-graphemes in a word into other phonologically similar ones, which is based on both place and manner of articulation, interestingly produces other words having different meanings without shifting the syllabification points. For examples, in Indonesian language, re- placing consonant-graphemes in the word ”ba.ra” (embers) generates three new words: ”ba.la” (disaster), ”pa.ra” (rubber), and ”pa.la” (nutmeg) with- out changing the syllabification points since the graphemes b and p are in the same category of plosive-bilabial while r and l are thrill/lateral-dental. An observation on 50k Indonesian words shows that replacing consonant- graphemes in those words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this research, a procedure of swapping consonant-graphemes based on phonological similarity is proposed to boost the standard stupid backoff smoothed bigram-based orthographic syllabification, which commonly has a low performance for a dataset with many out-of-vocabulary (OOV) bi-grams. Some examinations using 5-fold cross-validation on the dataset of 50k formal words of Indonesian language prove that the proposed procedure is capable of increasing the performance of the standard bigram-syllabification, where the mean syllable error rate (SER) can be relatively decreased by up to 31.39%. It also shows an improvement for the dataset of named-entities by relatively reducing the average SER by 9.53%. Compared to the nearest neighbour model, its performance is a little worse but it provides much lower complexity. Another important finding is that the proposed model can achieve a small SER by using a tiny training-set.

Response to Reviewer Comments (Without any Author Details)

Reviewer #1:

The paper is well organized and its English is very good. The topic covered is also a good one and of great value for our community. I prefer that the author/s to use many datasets instead of one. this would give better comparisons and results. It is not clear the justification of using the indicated equations. Instead of indicating the disadvantages of the proposed model, I prefer to find a way to solve these deficiencies. in general, the work is good.

>> Two datasets, i.e. a dataset of 15k named-entities and a mixed dataset of formal world and named-entities, are now added to evaluate the performance of the proposed model.

>> The justification is now explained more detail and clearer as follows:

Hence, the value of *B* should be less than 1.0 because swapping procedure on 50k formal words creates up to 22.55% illegal bigrams. Meanwhile, the value of *U* is probably much lower than *B* since a unigram is less important than a bigram in deciding the score of syllabification.

>> The subsection indicating the disadvantages of the proposed model is now removed since quite hard to find a simple way to solve the limitations of the proposed model.

Reviewer #2: While observations of language acquisition may have motivated this research it is not germane to the manuscript and may be misleading in the abstract. I am not sure "slur" is the best way to describe the transform or mapping operation performed. In Section 1 the term "suffix" appears to be applied to what is actually a prefix. Further detail should be provided about the slurring process itself. In Sections 3.1-3.3 it would be preferable to optimize the parameters jointly given the modest additional computational cost. In Section 3.4 the additional complexity of FkNNC should be quantified. In general I might characterize this approach as a back-off procedure based on phonological similarity with known bigrams.

>> The sentences related to the observations of language acquisition are now removed in the Abstract.

>> All terms "suffix" in Section 1 are now replaced by "prefix".

>> The three parameters are now jointly optimized

>> The additional complexity of FkNNC is now quantified.

>> To match the characteristic of the proposed procedure, the title is now revised to be “Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection”.

Title Page containing all the Author Details

IJST manuscript No.
(will be inserted by the editor)

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

Suyanto Suyanto

Received: date / Accepted: date

Suyanto Suyanto
School of Computing, Telkom University, Bandung, West Java 40257, Indonesia
Tel.: +62 22 7564108, Mobile: +62 812 845 12345
Orcid: https://orcid.org/0000-0002-8897-8091
E-mail: suyanto@telkomuniversity.ac.id

IJST manuscript No.
(will be inserted by the editor)

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

Received: date / Accepted: date

Abstract Swapping one or more consonant-graphemes in a word into other phonologically similar ones, which is based on both place and manner of articulation, interestingly produces other words having different meanings without shifting the syllabification points. For examples, in Indonesian language, replacing consonant-graphemes in the word "***ba.ra***" (embers) generates three new words: "***ba.la***" (disaster), "***pa.ra***" (rubber), and "***pa.la***" (nutmeg) without changing the syllabification points since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. An observation on 50k Indonesian words shows that replacing consonant-graphemes in those words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this research, a procedure of swapping consonant-graphemes based on phonological similarity is proposed to boost the standard stupid backoff smoothed bigram-based orthographic syllabification, which commonly has a low performance for a dataset with many out-of-vocabulary (OOV) bigrams. Some examinations using 5-fold cross-validation on the dataset of 50k formal words of Indonesian language prove that the proposed procedure is capable of increasing the performance of the standard bigram-syllabification, where the mean syllable error rate (SER) can be relatively decreased by up to 31.39%. It also shows an improvement for the dataset of named-entities by relatively reducing the average SER by 9.53%. Compared to the nearest neighbour model, its performance is a little worse but it provides much lower complexity. Another important finding is that the proposed model can achieve a small SER by using a tiny training-set.

Keywords backoff smoothing · bigram · Indonesian language · orthographic syllabification · phonological similarity

1 Introduction

One of the important pronunciation units in a language is a syllable. It is strongly relevant to the phonology rules. In [7], the researcher states that a syllable is a representational unit used to learn the phonotactic constraints of speech-sounds. The syllable is generally believed to be central to the infant as well as the adult perception of speech [34]. The syllable theories are based on much evidence. One of them is evidence from child language acquisition [11].

In linguistic theory, a syllable consists of an obligatory nucleus with or without non-obligatory surrounding consonants called onset and coda [16]. In the Indonesian language, the nucleus can be either vowel or diphthong [2]. Meanwhile, both onset and coda are consonants [2]. For instance, a word "*pantai*" (beach) contains two syllables: ⟨pan⟩ and ⟨tai⟩. The former consists of an onset ⟨p⟩, a nucleus of single vowel ⟨a⟩, and a coda ⟨n⟩. The later is composed of an onset ⟨t⟩, a nucleus of diphthong ⟨ai⟩, but no coda.

A model of syllable boundary detection, also known as automatic syllabification, is defined as a process of dividing a word into syllables. This model is urgent for some researches as well as application developments in the field of linguistics, e.g. grapheme-to-phoneme conversion (G2P) [36], [43], spelling-checker [23], [30], machine transliteration [29], speech synthesis [14], [10], [3], [27], speaking rate estimation [28], speaking proficiency scoring [17], word count estimation [34], speech recognition [28], [12], [31], [18], dialect identification [22], speech emotion recognition [6], [40], forensic-voice [38], early childhood digital literacy [21], etc.

The automatic syllabification is commonly implemented using two different approaches: either orthographic or phonemic-based. The previous works show that the phonemic-syllabification [44] performs better than the orthographic one [32] but it requires a linguist to provides perfect phoneme sequences. A model of phonemicization or G2P can be created to replace the linguist role but a small phoneme error rate (PER) decreases its performance in terms of SER [44]. Besides, it potentially performs much worse for named-entities having many exceptions and ambiguities. Therefore, many researchers are interested in the orthographic (also known as graphemic) syllabification as it is much simpler and more flexible regarding the dataset for the learning process when the statistical models are used.

A syllabification is generally implemented using statistical models, instead of rule-based ones, since they are easier to implement and give lower SER [1]. The statistical models are usually implemented using either supervised or unsupervised learning technique, such as Näive Bayes model [4], decision tree-based model [9], treebank model [25], random forest model [4], neural-based model [45], support vector machine model, [5], finite-state transducers model [20], [15], context-free grammars model [26], Hidden Markov Model [19], syllabification by analogy [1], dropped-and-matched model [33], n-gram model [37], conditional random fields model [39], [35], nearest neighbour-based model [44], [32], and unsupervised-syllabification model based on a classification of graphemic-symbols into two categories: consonants and vowels [24].

The nearest neighbour is one of the interesting models since it produces a low SER [44], [32]. Unfortunately, it has a high complexity of computation. It also requires complex language-specific knowledge. Besides, a graphemic encoding proposed in [32] produces a relatively high SER since it does not accurately represent a language-specific knowledge.

Another interesting model is the n-gram syllabification since it gives both competitive SER and low complexity. Besides, it is simple to implement as well as language-independent that does not require any language-specific phonotactic knowledge. Unfortunately, it has a disadvantage for a small dataset with a high rate of OOV bigrams. Many researchers have proposed various procedures to make some improvements, such as the segmental conditional random fields (SCRF) [35] and the bigram with flipping onsets (BFO) [42]. The SCRF is a bigram-based syllabification smoothed by a simple *stupid backoff* scheme described in [8]. It performs excellent, with a high generalization, even for quite small training-sets. Unfortunately, it looks very complex since eight features generated by sonority, legality, and maximum onset are taken into account in calculating the bigram-probability. Meanwhile, the BFO offers a simple computation but it produces quite high SER for the Indonesian language [42].

In this research, a new procedure of phonological similarity-based backoff smoothing is proposed to boost the standard stupid backoff smoothing bigram-syllabification. This procedure is inspired by a fact that replacing one or more consonant-graphemes in a word into other phonologically similar ones, which is based on both place and manner of articulations, may create other words without shifting the syllabification points. For instance, swapping two consonant-graphemes in an Indonesian word "***b****a.****r****u*" (new) produces three new words: "***b****a.****l****u*" (widower), "***p****a.****r****u*" (lung), and "***p****a.****l****u*" (hammer) without shifting the points of syllabifications since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. This procedure increases the number of bigrams, which means that the OOV rate can be reduced.

Indonesian language has eighteen prefixes [2]. An interesting phenomenon is swapping consonant-graphemes in a prefix generally not just produces another legal prefix but also a few illegal ones (noises). For instance, swapping grapheme ⟨b⟩ in a prefix ⟨ber⟩ in the word "*be.ra.tu.ran*" (regular) into ⟨p⟩ produces another legal prefix ⟨per⟩ in "*pe.ra.tu.ran*" (rules). Swapping a prefix ⟨pe⟩ in "*pe.nam.pi.lan*" (performance) produces an OOV word "*be.nam.pi.lan*". But, all syllables in the OOV word produce legal bigrams come from other words: "*be.nam*" (immerse); "*nam.pi*" that come from "*me.nam.pi*" (winnow), "*pe.nam.pi*" (shelter), "*pe.nam.pi.lan*" (performance), and some other words; and "*pi.lan*" that come from "*a.pi.lan*" (breastwork), "*kam.pi.lan*" (appearance), "*pi.pi.lan*" (flat), "*pe.nam.pi.lan*" (performance), and many other words. Swapping a prefix ⟨ter⟩ in "*ter.ba.wa*" (not deliberately taken away) produces illegal prefix ⟨der⟩ in an OOV word "*der.ba.wa*" with a bigram "*der.ba*" that is never found in 50k words but, based on the Indonesian phonotactic rules, it is a legal bigram.

For English and other European languages, the procedure of swapping consonant-graphemes in many words may produce huge illegal syllable-unigrams and syllable-bigrams. However, for the Indonesian language, the procedure creates more new legal syllable-unigrams and syllable-bigrams than the illegal ones. A preliminary study shows that 50k Indonesian words produce a total of 161,981 legal syllable-unigrams. Swapping those 50k words produces a total of 2,676,764 swapped syllable-unigrams, where 87.36% of them are legal unigrams that are the same as produced by the original words and the rest 12.64% are unseen syllable-unigrams. It means that the swapping procedure significantly increases the number of unigrams by 16.52 times (14.44 times legal unigrams and 2.08 times unseen unigrams). Furthermore, those 50k words produce a total of 212,550 syllable-bigrams. Swapping them produces a total of 3,317,292 swapped syllable-bigrams, where 77.45% are legal bigrams that are the same as produced by the original words and the rest 22.55% are considered unseen syllable-bigrams. It means that the swapping procedure impressively increases the number of bigrams up to 14.12 times (12.09 times legal bigrams and 2.03 times unseen unigrams). Those unseen syllable-unigrams and syllable-bigrams can be either legal or illegal based on the Indonesian phonotactic rules. However, it is not easy to classify them into both classes.

In this research, the impact of swapping consonant-graphemes in a word is investigated on an Indonesian orthographic syllabification. First, the standard bigram-syllabification (BS) smoothed by the *Stupid Backoff* is implemented. Next, the combination of standard bigram-syllabification and phonological similarity-based backoff smoothing (CBSPS) is developed and then examined whether it is capable of improving the performance of BS in terms of SER. Since it is not easy to detect the unseen syllable-unigrams and syllable-bigrams as legal or illegal, CBSPS is implemented using all of them (not just the legal ones). Hence, this research focuses on examining whether the proposed procedure can enhance the performance of BS or not.

2 Research Method

Fig. 1 shows the block diagram of the training process of the CBSPS model proposed in this research. It is a combination of standard and swapped bigram-syllabifications. A dataset of pairs of words and their corresponding syllabifications is used to develop a list of standard or normal syllables, a table of syllable-unigrams, and a table syllable-bigrams on the left side. It is also used to develop a list of swapped syllables, a table of swapped syllable-unigrams (or swapped-unigrams), and a table of swapped syllable-bigrams (or swapped-bigrams) on the right side. The generated tables of both normal and swapped syllable-unigrams and syllable-bigrams are then exploited in the testing process, as illustrated by Fig. 2, to maximize the final score to produce the best sequence of syllables that has the highest score.

The testing process in Fig. 2 illustrates an input sequence of graphemes ⟨*pandai*⟩ (smart) is quite hard to be syllabified since ⟨ai⟩ is a diphthong, not

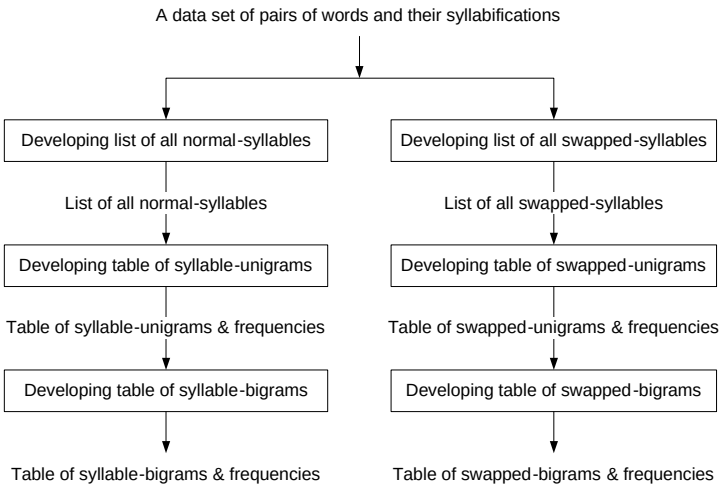


Fig. 1 Training process of the proposed CBSPS model

two independent vowels ⟨a⟩ and ⟨i⟩. First, three vowels {⟨a⟩, ⟨a⟩, and ⟨i⟩} contained in the grapheme sequence are detected in the positions {2, 5, 6}. A well known high accurate method called Sukhotin's algorithm proposed in [13] can be exploited to automatically detect vowels and diphthongs but it is not used here. Instead, this research just uses the simple Indonesian typological knowledge explained in [2], where five graphemes {⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩, ⟨u⟩} can be single vowels; four grapheme sequences {⟨ai⟩, ⟨au⟩, ⟨ei⟩, ⟨oi⟩} may produce diphthongs; and other graphemes are considered as consonants.

Next, six possible syllabifications are generated, i.e. ⟨*pa.nda.i*⟩, ⟨*pan.da.i*⟩, ⟨*pa.ndai*⟩, ⟨*pan.dai*⟩, ⟨*pand.ai*⟩, and ⟨*pand.a.i*⟩, where two graphemes ⟨*ai*⟩ may produce a diphthong or two single vowels forming one or two nucleuses. The score of each candidate is then calculated using the formula in Equation (3). In this case, the fourth candidate ⟨*pan.dai*⟩ gives the highest score since swapping consonant-graphemes in this candidate produces two legal bigrams ⟨*pan.tai*⟩ (beach) and ⟨*ban.tai*⟩ (slaughter) as well as an OOV bigrams: ⟨*ban.dai*⟩ (it is not an Indonesian word) while the other five candidates produce all OOV bigrams. It means that CBSPS is capable of syllabifying the input sequence of graphemes ⟨*pandai*⟩ into ⟨*pan.dai*⟩, where ⟨*ai*⟩ is correctly detected as a diphthong.

2.1 Standard bigram-syllabification model

The BS model works by maximizing the likelihood of syllable sequences for a given word. The likelihood can be estimated using a probability chain, which is commonly smoothed by a simple *Stupid Backoff* to produce a more accurate probability, which is here called *score* since its value can be more than 1, for

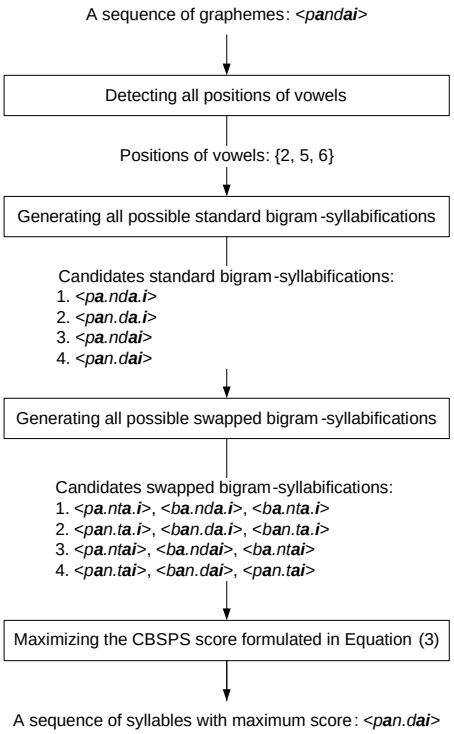


Fig. 2 Testing process of the proposed CBSPS model

a training-set with many OOV words [8]. In this method, the score of bigram-syllabification S_{bs} is calculated as

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ are the frequencies of both syllable bigram and syllable unigram appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N is the training-set size, and α is the backoff factor that generally set to 0.4 [8]. The model of BS commonly gives a low performance for a small training-set that has a high rate of OOV syllable [35]. Besides using a smoothing procedure, the performance of BS can be improved by decreasing the OOV rate.

2.2 Combination of standard and phonological similarity-based bigram model

A procedure of swapping consonant-graphemes in the training-set is proposed here to decrease the OOV rate in the BS model. This procedure forms a new model called phonological similarity-based bigram-syllabification, which has a score S_{ps} formulated as

$$S_{ps}(w_i|w_{i-1}) = \begin{cases} B\frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} \text{ if } f_s(w_{i-1}w_i) > 0 \\ \\ U\alpha\frac{f_s(w_i)}{N_s} \text{ otherwise} \end{cases} \quad (2)$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the frequencies of both swapped-bigram and swapped-unigram appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N_s is the size of the swapped training-set, B is a weight of swapped-bigram, U is a weight of swapped-unigram, and α is the backoff factor as used in Equation 1. Both weights B and U are introduced here to smooth the score since the swapped-consonant words may produce some illegal bigrams and/or illegal unigrams. Hence, the value of B should be less than 1.0 because swapping procedure on 50k formal words creates up to 22.55% illegal bigrams. Meanwhile, the value of U is probably much lower than B since a unigram is less important than a bigram in deciding the score of syllabification.

Finally, the CBSPS model uses the combined score S_{cbsps} that is simply calculated as

$$S_{cbsps} = S_{bs} + S_{ps} \quad (3)$$

where S_{bs} is the score of bigram-syllabification in Equation (1) and S_{ps} is the score of phonological similarity-based model in Equation (2).

2.3 Phonological similarity-based graphemes

Table 1 illustrates the graphemes and their phonological similarities based on the categorization of phonemes described in [2] as well as their examples in some Indonesian formal words. Here, the graphemes and their swaps are simply mapped to those phoneme categorizations since they are strongly related to the corresponding phonemes [2], [41]. A formal word containing one of those 14 graphemes, which are grouped into 7 categories, can be swapped to produce another formal word as shown in the last column (examples). In [2], both phonemes /g/ and /k/ are in the same category (plosive-velar). But, they are not used here since swapping grapheme ⟨g⟩ into ⟨k⟩ commonly produces many illegal syllable-unigrams and syllable-bigrams, such as swapping consonant-graphemes in the word "*me.mang.sa*" (prey on) generates "*me.mank.sa*" (OOV) with an illegal syllable-unigram "*mank*" and two illegal bigrams "*me.mank*" and "*mank.sa*". Instead, the grapheme ⟨q⟩ is used

here since, in Indonesian language, it is always pronounced as a phoneme /k/ [2], [44].

Table 1 Consonant-graphemes and their swapping as well as the example of the swapped-consonant words without changing their points or boundaries of syllabifications

Grapheme category	Graph.	Swap	Example
Plosive-Bilabial: {b, p}	b	p	***b****a.ru* (new) → ***p****a.ru* (lung)
	p	b	***p****a.du* (intact) → ***b****a.du* (checkered)
Plosive-Dental: {d, t}	d	t	***d****a.ri* (from) → ***t****a.ri* (dance)
	t	d	***t****a.hi* (feces) → ***d****a.hi* (forehead)
Plosive-Velar: {k, q}	k	q	***k****a.ri* (curry) → ***q****a.ri* (reciter)
	q	k	*a.**q**i.dah* (creed) → *a.**k**i.dah* (creed)
Affricative-Palatal: {c, j}	c	j	***c****a.ri* (find) → ***j****a.ri* (finger)
	j	c	***j****an.da* (widow) → ***c****an.da* (joke)
Fricative-Labiodental: {f, v}	f	v	***f****i.si* (fission) → ***v****i.si* (vision)
	v	f	***v****o.li* (volley) → ***v****o.li* (thin metal)
Fricative-Dental: {s, z}	s	z	***s****a.man* (indict) → ***z****a.man* (era)
	z	s	*a.**z**am* (aim) → *a.**s**am* (acid)
Thrill/Lateral-Dental: {l, r}	l	r	***l****i.ma* (five) → ***r****i.ma* (rhyme)
	r	l	***r****a.bu* (Wednesday) → ***l****a.bu* (pumpkin)

Meanwhile, Table 2 illustrates the examples of some swapped-consonant words generated from the words containing two or more possible swappng-graphemes without changing the point or boundary of syllabification. A word "***b****a.**r**a*" (embers), which has two possible swapping graphemes ⟨b⟩ and ⟨r⟩, can be swapped to produce three other words, i.e. "***b****a.**l**a*" (disaster), "***p****a.**r**a*" (rubber), and "***p****a.**l**a*" (nutmeg) without changing the syllabification points. A word "***b****i.**r**u*" (blue), which has two possible swapping graphemes, can be swapped into three new words: "***b****i.**l**u*", "***p****i.**r**u*", and "***p****i.**l**u*" without changing the syllabification points. There is no formal word "***b****i.**l**u*" in Indonesian language (it means that "***b****i.**l**u*" is an OOV word). But, it can be a sub-word for some other words, such as "*sem.**b**i.**l**u*" (sharp reed skin like a knife). The word "***p****i.**r**u*" is also an OOV word but it is a sub-word for the word "***p****i.**r**u.et*" (one of ballet dance styles). In contrast, the word "***p****i.**l**u*" is a formal word that means "really sad" in English. No doubt, such swapped-consonant words increase the number of bigrams. Hence, swapping word can be seen as a method of data augmentation. This is expected to produce a more accurate score in Equation (3) so that a better syllabification can be achieved.

3 Result and Discussion

The data-sets used here are formal words, named-entities, and the mixture of both data-sets. The first two data-sets are the same as described in [32]. The formal word dataset consists of 50k words equipped with boundaries or points of syllabifications. It is equally divided into five subsets (folds), where each subset consists of 10k words, to do the five-fold cross-validation. The dataset of named-entities contains 15k entries and their syllable boundaries. It is also

Table 2 Examples of some new words produced by swapping consonants-graphemes in the original words without shifting the point or boundary of syllabification

Original word	Swapped-consonant words
ba.ra (embers)	*ba.la* (disaster), *pa.ra* (rubber), *pa.la* (nutmeg)
ba.ru (new)	*ba.lu* (widower), *pa.ru* (lung), *pa.lu* (hammer)
bi.ru (blue)	*pi.ru* (OOV), *bi.lu* (OOV), *pi.lu* (really sad)
ba.rat (west)	*ba.rad* (OOV), *ba.lat* (OOV), *ba.lad* (city), *pa.rat* (OOV), *pa.rad* (OOV), *pa.lat* (penis), *pa.lad* (OOV)
ca.ri (find)	*ca.li* (OOV), *ja.ri* (finger), *ja.li* (real)
ce.ri.ta (story)	*ce.ri.da* (OOV), *ce.li.ta* (OOV), *ce.li.da* (OOV), *je.ri.ta* (OOV), *je.ri.da* (OOV), *je.li.ta* (very beautiful), *je.li.da* (OOV)

equally divided into 5 folds, each contains 3k words. The mixed dataset consists of 65k entries and their syllable boundaries. It is also equally divided into 5 folds, each contains 13k entries.

3.1 Evaluation on the dataset of formal words

In this evaluation, five experiments are conducted to sequentially tune the parameters. Firstly, the optimum unigram weight U is searched using $\alpha = 0.4$ as suggested in [8] and $B = 1.0$ based on an assumption that the swapped-bigrams have the same importance as the normal-bigrams. Secondly, the bigram weight B is then optimized using the found optimum U and $\alpha = 0.4$. Thirdly, the backoff factor α is verified using both optimum values of U and B. Fourthly, the three parameters are jointly optimized using the potential values resulted from the previous three experiments. Finally, CBSPS is fairly compared to other syllabification models. Here, a percentage of errors in the syllable level, which is commonly known as SER, is used to measure all performances in those experiments.

Optimizing unigram weight U. The CBSPS is firstly evaluated using $\alpha = 0.4$ and $B = 1.0$ to find the optimum unigram weight U. The results illustrated in Fig. 3 informs that U is very sensitive. A very small $U = 0.001$ produces high SERs for all folds. A big $U = 0.1$ or bigger also gives higher SERs. The unigram weight U reaches the optimum value of 0.05 that produces the lowest SERs for all folds with the average SER of 2.65%. As hypothesized, the optimum value of this parameter is very low of only 0.05 (much lower than B), which means that the impact of the swapped unigrams is just 5% to do the syllabification.

Optimizing bigram weight B. The CBSPS is then evaluated using $\alpha = 0.4$ and $U = 0.05$ to optimize the bigram weight B. The results in Fig. 4 shows that B is not sensitive. It is quite stable to produce low SERs for all folds when the value is in the interval of 0.8 to 1.1. It reaches the optimum value of 1.0 that produces the lowest average SER of 2.65%.

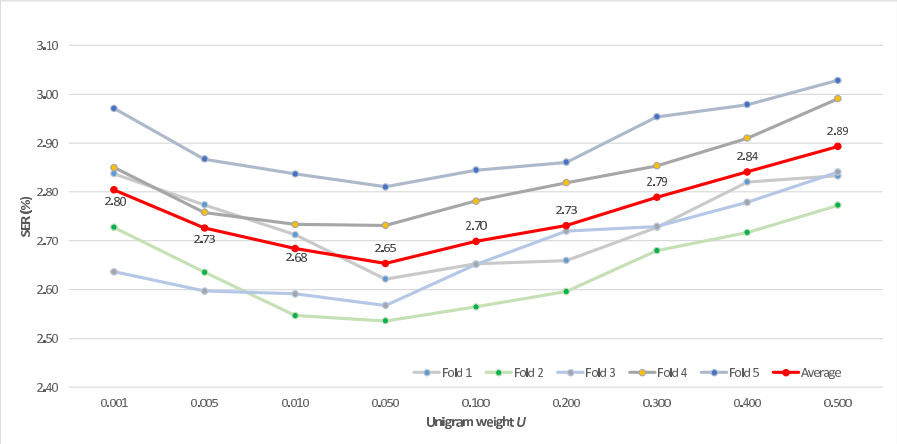


Fig. 3 SERs produced by CBSPS using $\alpha = 0.4$, $B = 1.0$, and varying unigram weight U

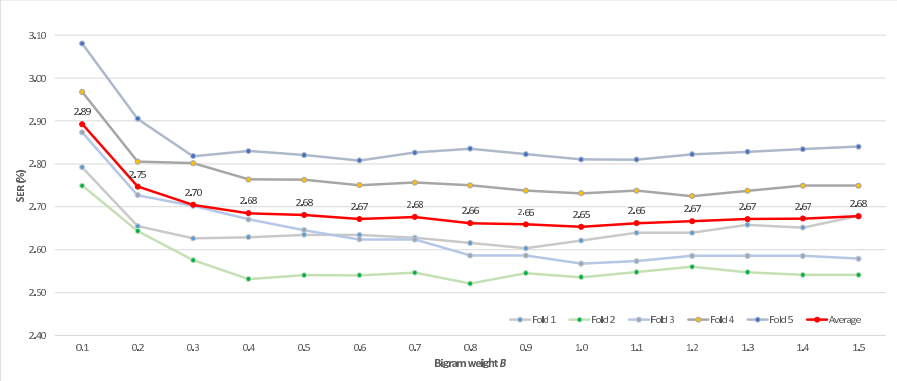


Fig. 4 SERs produced by CBSPS using $\alpha = 0.4$, $U = 0.05$, and varying bigram weight B

Verifying backoff factor α. Next, the use of $\alpha = 0.4$ suggested in [8] is verified using both optimum values $U = 0.05$ and $B = 1.0$. Here, nine experiments are performed using $\alpha = 0.1$ to 0.9. The results in Fig. 5 informs that α is an easily tuned parameter. It gives the lowest average SER of 2.65% when the value is in the interval of 0.2 to 0.4. It means that the $\alpha = 0.4$ is verified in this research.

Jointly parameters optimization. Next, the three parameters are then jointly optimized using the potential values produced by the sequential tuning, where $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 6 shows that the optimum combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.2$ that gives the lowest average SER of 2.61%.

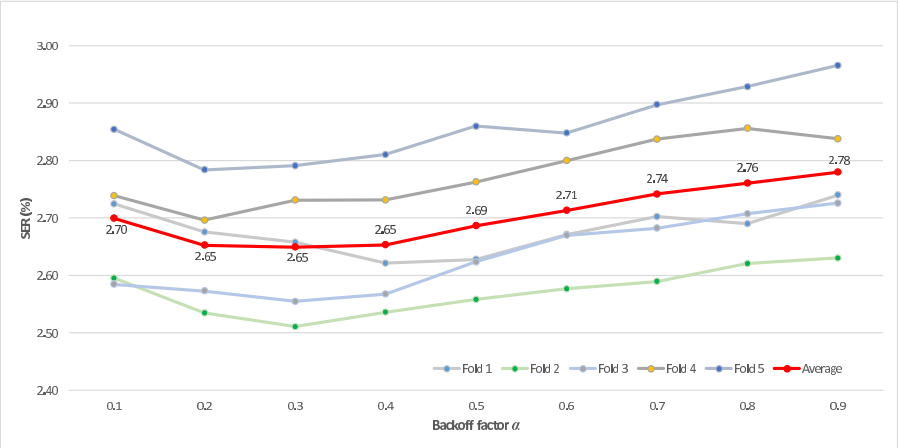


Fig. 5 SERs produced by CBSPS using $B = 1.0$, $U = 0.05$, and varying α

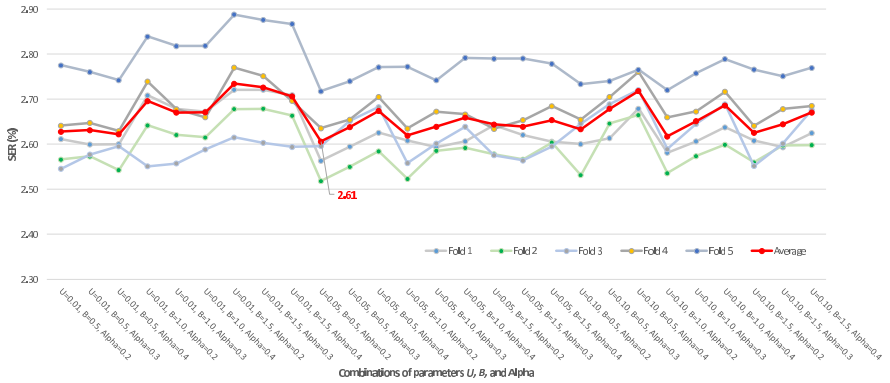


Fig. 6 SERs produced by CBSPS using jointly parameters optimization for the dataset of formal words

Comparison to other models. Finally, the best performance of CBSPS is compared to three other syllabification models: BS, BFO, and fuzzy k-nearest neighbour in every class (FkNNC), which uses the same dataset of 50k words described in [32]. The same testing-sets of 5 folds are also used in this evaluation. To get fairness, all methods are compared in their best performances for the same dataset. The results in Fig. 7 illustrate that CBSPS produces a lower average SER of 2.61% than BS with an average SER of 3.80%. Hence, CBSPS gives a relative reduction in SER up to 31.39%. It shows that the proposed swapping procedure is capable of boosting the performance of the standard bigram-syllabification model. The CBSPS also better than BFO that gives an

average SER of 3.11%. But, it is slightly worse than FkNNC, which reaches the lowest mean SER of 2.27%.

Nevertheless, CBSPS has much lower complexity than FkNNC since it just calculates the probabilities of bigrams to get the syllabification points while FkNNC should find the k nearest neighbours and then decide the syllabification points. CBSPS just searches tens or fewer bigrams as well as unigrams that are taken into account to determine the syllabification points, where the searching can be very fast using both indexed-sorted bigrams and unigrams. Meanwhile, FkNNC should compute up to 250 thousand distances between a candidate syllabification and all impossible indexed-sorted patterns in the training-set, choose the k closest patterns in both classes: point and not point of syllabification, and eventually select the class with the lowest total fuzzy-distance as the decision.

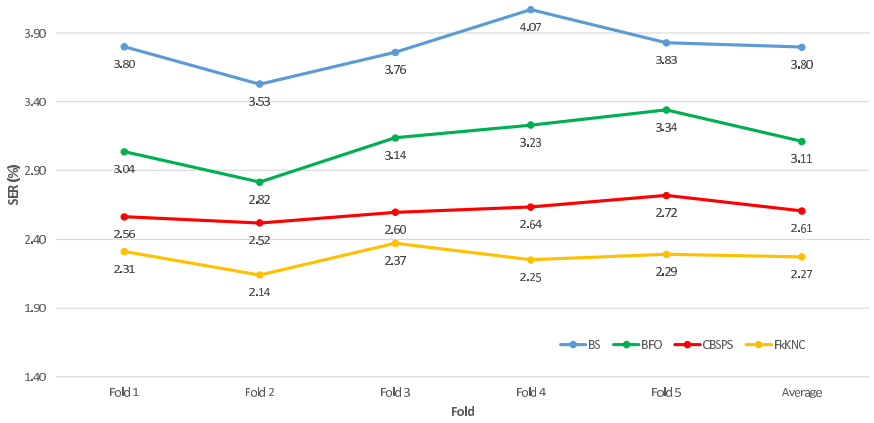


Fig. 7 SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of formal words

3.2 Evaluation on the dataset of named-entities

The proposed CBSPS just uses bigrams and unigrams as well as their phonological similarities to maximize the scores of syllabifications. Hence, it can be applied to a dataset of name-entities since the phonological similarities are very common in such a dataset. For example, mapping two graphemes ⟨b⟩ and ⟨d⟩ in a named-entity "**b**an.**d**ung" (the capital city in West Java) into their phonological similarities ⟨p⟩ and ⟨t⟩ produces three other named-entities: "**b**an.**t**ung" (a resort in Sukhothai, Thailand), "**p**an.**d**ung" (a village in Special Region of Yogyakarta), and "**p**an.**t**ung" (a folk song from Bolaang Mongondow, North Sulawesi).

Careful observation on the dataset of 15k named-entities informs that it produces a total of 45,799 legal syllable-unigrams. Swapping procedure on those 15k named-entities produces a total of 385,850 swapped syllable-unigrams, where 90.92% of them are legal unigrams that are the same as produced by the original words and the rest 9.08% are unseen syllable-unigrams. Hence, the swapping procedure significantly increases the number of unigrams by 8.43 times, i.e. 7.66 times legal unigrams and 0.77 times unseen unigrams. Furthermore, those 15k named-entities create a total of 30,516 syllable-bigrams. Swapping them produces a total of 275,472 swapped syllable-bigrams, where 84.61% are legal bigrams and the rest 15.39% are considered unseen syllable-bigrams. It means that the swapping procedure impressively increases the number of bigrams up to 9.03 times, i.e. 7.64 times legal bigrams and 1.59 times unseen unigrams. These facts indicate that the proposed CBSPS will be able to syllabify named entities better than BS. Therefore, CBSPS is evaluated here using the dataset of named-entities in a 5-fold cross-validation scheme. First, the three parameters U, B, and α are jointly optimized. The SER produced by the optimum values of those parameters is then compared to three other models: BS, BFO, and FkNNC.

Jointly parameters optimization. The three parameters are jointly optimized using the potential values resulted from the previous experiment on the formal word dataset, where $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 8 concludes that the optimum combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.3$ that gives the lowest average SER of 13.48%.

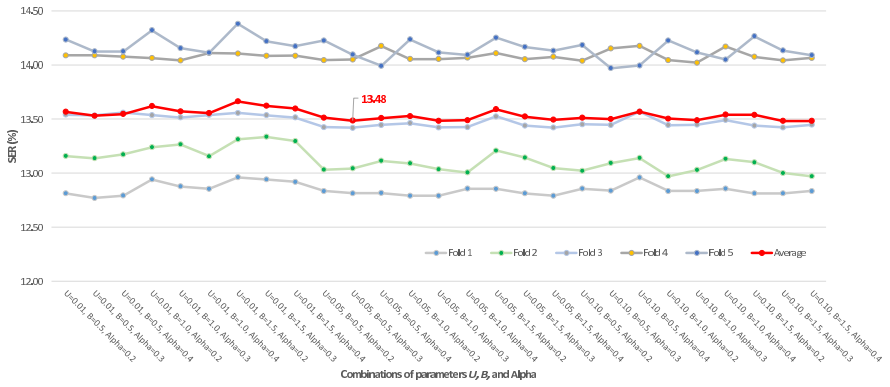


Fig. 8 SERs produced by CBSPS using jointly parameters optimization for the dataset of named-entities

Comparison to other models. The best performance of CBSPS is then compared to three other syllabification models: BS, BFO, and FkNNC, which uses the same dataset of 50k words described in [32]. The same testing-sets of 5 folds are also used in this evaluation. To get fairness, all methods are compared in their best performances using the same dataset as illustrated by Fig. 9. The CBSPS produces a lower average SER of 13.48% than BS (14.90%), which means that CBSPS relatively decreases the mean SER by 9.53%. It is also better than BFO that gives an average SER of 14.15%. But, it is much worse than FkNNC, which reaches the lowest mean SER of 6.78%. This is caused by a lot of vocal ambiguity in the named-entities. For instances, the person names "A.dy", "A.di", and "A.dhie" are pronounced as /ɑ.di/ and the person names "Bu.dy", "Bu.di", and "Bu.dhie" are pronounced as /bu.di/. Since CBSPS always considers the grapheme ⟨y⟩ as a consonant, not a semi-vowel nor a vowel, it fails to syllabify "Budy" into "bu.dy".

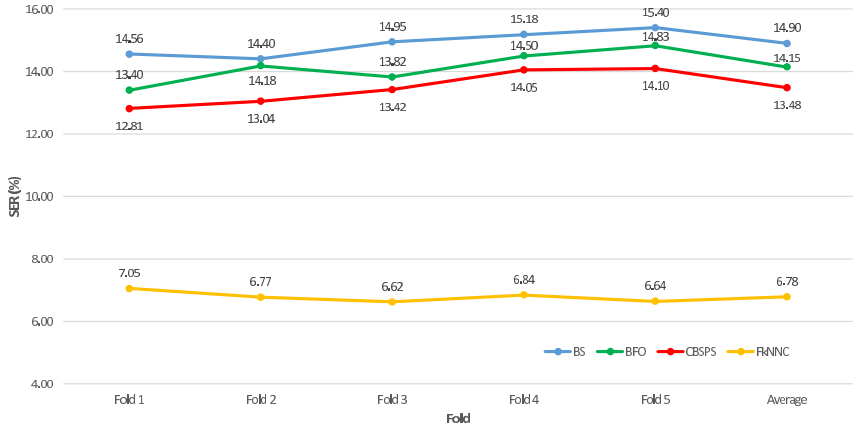


Fig. 9 SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of named-entities

3.3 Evaluation on the mixed dataset of formal words and named-entities

The proposed CBSPS is finally evaluated using the mixture dataset of 50k formal words and 15k named-entities to see its generalization. This dataset of 65k entries is equally divided into 5 folds, each contains 13k entries. First, the three parameters of CBSPS are jointly optimized. The result is then compared to two other models: BS and BFO.

Jointly parameters optimization. The three parameters are jointly optimized using the potential values resulted from the previous experiments on the

formal word dataset, where $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 10 shows that the optimum combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.2$ that gives the lowest average SER of 4.88%. A detail observation shows that SER come from the named-entities is slightly lower than the previous model, which is trained using the dataset of named-entities only, but SER from the formal words does not decrease at all. It means that CBSPS is just stable for the formal words but it is capable of generalizing bigrams from the formal words into the named-entities.

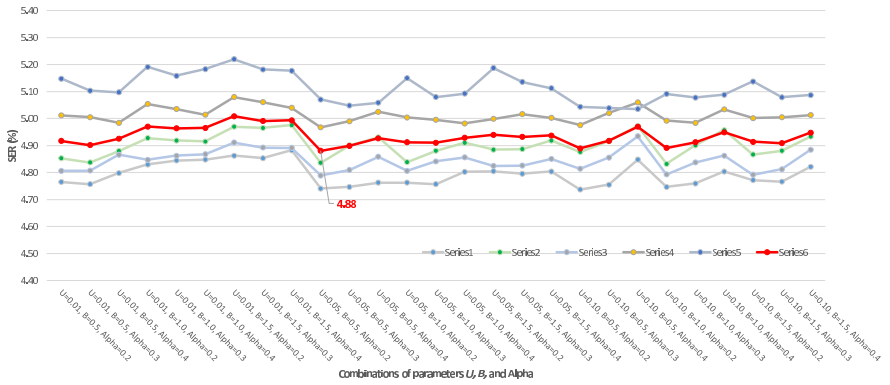


Fig. 10 SERs produced by CBSPS using $B = 1.0$, $U = 0.05$, and varying α for the mixed dataset of formal words and named-entities

Comparison to other models. The best performance of CBSPS is then compared to two other syllabification models, BS and BFO, using the mixed dataset of 65k words. Unfortunately, in this case, it cannot be compared to the FkNNC since there is no experimental result provided in [32]. The results in Fig. 11 show that CBSPS produces a much smaller average SER of 4.88% than BS (6.02%), which means that CBSPS relatively decreases the mean SER by 18.94%. It is also much better than BFO that produces an average SER of 5.74%. This result also informs that the performance of the proposed CBSPS is stable for all five folds.

3.4 Training-set sizes

To investigate the impact of training-set sizes, some varying training-set sizes are developed by randomly selecting words from the five folds in the dataset of formal words. First, each fold is defined as the fixed testing-set. Next, six training-set sizes of 1k, 5k, 10k, 20k, 30, and 40k are then randomly generated

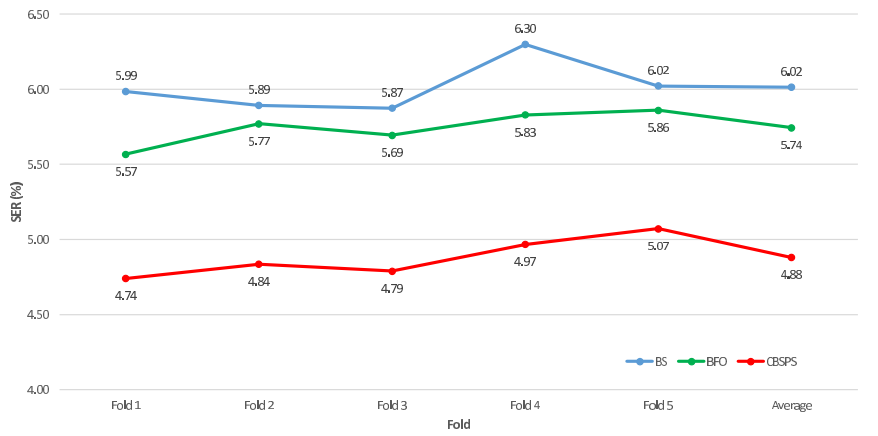


Fig. 11 SERs produced by BS, BFO, and CBSPS models for the mixed dataset of formal words and named-entities

five times from the remaining four folds (unseen data in the testing sets) so that each training-set size contains five subsets. The comparisons of BS, BFO, and CBSPS are then performed repeatedly five times for each subset. Unfortunately, in this experiment, they cannot be compared to the FkNNC since there is no experimental result provided in [32].

The experimental results in Fig. 12 shows that CBSPS, for all training-set sizes, yields lower average SERs than BS. For the training-set size of 1k, CBSPS and BFO give the same average SER of 16.13% while BS produces 19.30%. For the training-set of 5k, CBSPS yields the lowest mean SER of 6.37% than both BFO (7.01%) and BS (8.93%). The interesting results come from the training-set of 10k, where CBSPS produces impressively lower mean SER of 3.68% than both BFO (4.96%) and BS (6.29%).

For the training-set of 20k, CBSPS and BFO yield the same mean SER of 3.94% that is smaller than BS (4.68%). For the 30k one, CBSPS provides a much lower average SER (1.57%) than both BFO (3.32%) and BS (4.09%). It means that CBSPS gives a relative reduction of the average SER by up to 61.61%. Finally, for the training-set of 40k, CBSPS also reaches the smallest mean SER of 2.61% among BFO (3.31%) and BS (3.80%).

Those results are very interesting. Increasing the size of training-set does not always decrease the SER. Sometimes it actually raises the SER. It can be said that CBSPS is not stable. This can be easily explained here that swapped-consonant words that produce many illegal OOV bigrams and/or unigrams potentially increase the SER. In other words, the unstable SERs produced by CBSPS are caused by some illegal OOV swapped-consonant words generated in the training-sets. Therefore, a scheme of filtering possible legal-bigrams can be introduced to improve the performance of CBSPS.

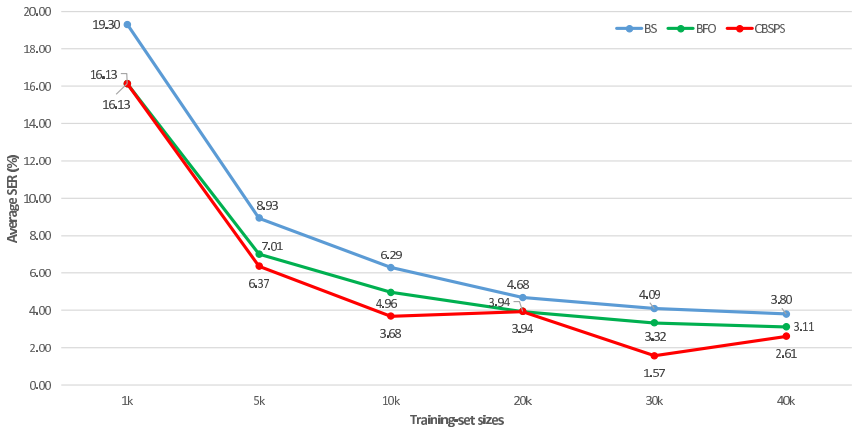


Fig. 12 SERs produced by BS, BFO, and CBSPS for six different training-set sizes taken from the 50k Indonesian formal words

4 Conclusion

The proposed CBSPS is capable of significantly boosting the standard bigram-syllabification model smoothed by the stupid backoff for the dataset of formal words. It relatively reduces the average SER up to 31.39%. Its performance is slightly worse than the FkNNC-based syllabification but it offers much lower complexity since it just calculates the probabilities of bigrams and unigrams to get the syllabification points. It slightly improves the standard bigram-syllabification model for the named-entities, where it relatively reduces the average SER by 9.53%. Another finding is the CBSPS gives a low SER of 3.68% for the testing-set of unseen 10k formal words by using a tiny training-set of 10k formal words only. In the future, a scheme of filtering possible legal-bigrams can be introduced to improve its performance.

Acknowledgements

I give a high appreciation to my three-year son, Muhammad Agha Ariyanto, for his great inspiration by speaking several words by swapping some consonants into the similar ones as well as several colleagues at Telkom University for the supports.

References

1. Adsett, C.R., Marchand, Y., Kešelj, V.: Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian. Computer Speech and Language **23**, 444–463 (2009). DOI 10.1016/j.csl.2009.02.004

2. Alwi, H., Lapoliwa, H., Darmowidjojo, S.: Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3 edn. Balai Pustaka (2003)
3. Aripin, Haryanto, H., Sumpeno, S.: 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 **13**(8), 19–37 (2018). DOI 10.3991/ijet.v13i08.8084
4. Balc, D., Beleiu, A., Potolea, R., Lemnaru, C.: A learning-based approach for Romanian syllabification and stress assignment. In: P. R. (ed.) Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015, pp. 37–42. Institute of Electrical and Electronics Engineers Inc. (2015). DOI 10.1109/ICCP.2015.7312603
5. Bartlett, S., Kondrak, G., Cherry, C.: 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, pp. 308–316. Boulder, Colorado (2009). DOI 10.3115/1620754.1620799
6. Ben Alex, S., Babu, B.P., Mary, L.: Utterance and syllable level prosodic features for automatic emotion recognition. In: 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018, pp. 31–35. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/RAICS.2018.8635059. URL https://ieeexplore.ieee.org/document/8635059
7. Bernard, A.: An onset is an onset: Evidence from abstraction of newly-learned phonotactic constraints. Journal of Memory and Language **78**, 18–32 (2015). DOI https://doi.org/10.1016/j.jml.2014.09.001. URL http://www.sciencedirect.com/science/article/pii/S0749596X1400103X
8. Brants, T., Popat, A.C., Och, F.J.: 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, pp. 858–867 (2007). URL https://www.aclweb.org/anthology/D07-1090
9. Daelemans, W., Bosch, A.V.D., Weijters, T.: IGTree: Using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review **11**(1-5), 407–423 (1997). DOI 10.1.1.29.4517
10. Faldessai, N., Pawar, J., Naik, G.: Syllabification: An effective approach for a TTS system for Konkani. In: 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, ICEECCOT 2016, pp. 161–167. Institute of Electrical and Electronics Engineers Inc. (2017). DOI 10.1109/ICEECCOT.2016.7955207. URL https://ieeexplore.ieee.org/document/7955207
11. Fallows, D.: Experimental evidence for English syllabification and syllable structure. Journal of Linguistics **17**(2), 309–317 (1981). DOI 10.1017/S0022226700007027
12. Feng, S., Lee, T.: Exploiting Cross-Lingual Speaker and Phonetic Diversity for Unsupervised Subword Modeling. IEEE/ACM Transactions on Audio Speech and Language Processing **27**(12), 2000–2011 (2019). DOI 10.1109/TASLP.2019.2937953. URL https://ieeexplore.ieee.org/document/8818297
13. Foster, C.C.: A Comparison of Vowel Identification Methods. Cryptologia **16**(3), 282–286 (1992). DOI 10.1080/0161-119291866955. URL https://doi.org/10.1080/0161-119291866955
14. Geeta, S., Muralidhara, B.L.: 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, pp. 1205–1208. Institute of Electrical and Electronics Engineers Inc. (2018). DOI 10.1109/WiSPNET.2017.8299954. URL https://ieeexplore.ieee.org/document/8299954
15. Hlaing, T.H., Mikami, Y.: Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer. International Journal on Advances in ICT for Emerging Regions (ICTer) **6**(2), 2–9 (2014). DOI 10.4038/icter.v6i2.7150
16. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall (2001)
17. Johnson, D.O., Kang, O.: Comparison of algorithms to divide noisy phone sequences into syllables for automatic unconstrained English speaking proficiency scoring. Artificial Intelligence Review pp. 1–24 (2017). DOI 10.1007/s10462-017-9594-y

18. Kamper, H., Jansen, A., Goldwater, S.: A segmental framework for fully-unsupervised large-vocabulary speech recognition. Computer Speech & Language **46**, 154–174 (2017). DOI https://doi.org/10.1016/j.csl.2017.04.008. URL http://www.sciencedirect.com/science/article/pii/S0885230816301905
19. Krantz, J., Dulin, M., De Palma, P., VanDam, M.: Syllabification by Phone Categorization. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, pp. 47–48. ACM, New York, NY, USA (2018). DOI 10.1145/3205651.3208781. URL http://doi.acm.org/10.1145/3205651.3208781
20. Krisnawati, L.D., Mahastama, A.W.: A Javanese Syllabifier Based on its Orthographic System. In: S.H.R.A.B.M.A.N.E.A.L.R.N.A.B.P. Dong M. Ruskanda F.Z. (ed.) International Conference on Asian Language Processing, pp. 244–249. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/IALP.2018.8629173. URL https://ieeexplore.ieee.org/document/8629173
21. Kulju, P., Mäkinen, M.: Phonological strategies and peer scaffolding in digital literacy game-playing sessions in a Finnish pre-primary class. Journal of Early Childhood Literacy (2019). DOI 10.1177/1468798419838576. URL https://journals.sagepub.com/doi/pdf/10.1177/1468798419838576
22. Leemann, A., Kolly, M.J., Nolan, F., Li, Y.: The role of segments and prosody in the identification of a speaker's dialect. Journal of Phonetics **68**, 69–84 (2018). DOI https://doi.org/10.1016/j.wocn.2018.02.001. URL http://www.sciencedirect.com/science/article/pii/S0095447016300365
23. Magdum, D., Suman, M.: System for identifying and correcting invalid words in the devanagari script for text to speech engine. International Journal of Innovative Technology and Exploring Engineering **8**(6 Special Issue 4), 1001–1006 (2019). DOI 10.35940/ijitee.F1206.0486S419
24. Mayer, T.: 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, pp. 63–71 (2010)
25. Müller, K.: 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, pp. 410–417. ACL (2001)
26. Müller, K.: 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, pp. 11–20 (2006). DOI 10.3115/1622165.1622167
27. Mulyanto, E., Yuniarno, E.M., Purnomo, M.H.: Adding an Emotions Filter to Javanese Text-to-Speech System. In: 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia, CENIM 2018 - Proceeding, pp. 142–146. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/CENIM.2018.8711229
28. Nayak, S., Bhati, S., Rama Murty, K.S.: 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, pp. 6590–6594. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/ICASSP.2019.8683462. URL https://ieeexplore.ieee.org/document/8683462
29. Ngo, G.H., Nguyen, M., Chen, N.F.: Phonology-Augmented Statistical Framework for Machine Transliteration Using Limited Linguistic Resources. IEEE/ACM Transactions on Audio Speech and Language Processing **27**(1), 199–211 (2019). DOI 10.1109/TASLP.2018.2875269. URL https://ieeexplore.ieee.org/document/8488567
30. Oncevay-marcos, A.: Spell-Checking based on Syllabification and Character-level Graphs for a Peruvian Agglutinative Language. In: The First Workshop on Subword and Character Level Models in NLP, pp. 109–116 (2017)
31. Pakoci, E., Popović, B., Pekar, D.: Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition. Computational Intelligence and Neuroscience **2019** (2019). DOI 10.1155/2019/5072918
32. Parande, E.A.: Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. International Journal of Speech Technology **22**(1), 13–20 (2019). DOI 10.1007/s10772-018-09569-3. URL https://link.springer.com/article/10.1007/s10772-018-09569-3

33. Ramli, I., Jamil, N., Seman, N., Ardi, N.: An Improved Syllabification for a Better Malay Language Text-to- Speech Synthesis (TTS). Procedia - Procedia Computer Science **76**(Iris), 417–424 (2015). DOI 10.1016/j.procs.2015.12.280. URL http://dx.doi.org/10.1016/j.procs.2015.12.280
34. Räsänen, O., Doyle, G., Frank, M.C.: Pre-linguistic segmentation of speech into syllable-like units. Cognition **171**, 130–150 (2018). DOI https://doi.org/10.1016/j.cognition.2017.11.003. URL http://www.sciencedirect.com/science/article/pii/S0010027717302901
35. Rogova, K., Demuynck, K., Compernolle, D.V.: Automatic syllabification using segmental conditional random fields. Computational Linguistics in the Netherlands Journal **3**, 34–48 (2013). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907935327&partnerID=40&md5=ec7496ee067bb9d2aca02f45d90d6bd0
36. Rugchatjaroen, A., Saychum, S., Kongyoung, S., Chootrakool, P., Kasuriya, S., Wutiwiwatchai, C.: Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion. Speech Communication **106**, 105–111 (2019). DOI https://doi.org/10.1016/j.specom.2018.12.003. URL http://www.sciencedirect.com/science/article/pii/S0167639317303965
37. Schmid, H., Möbius, B., Weidenkaff, J.: Tagging syllable boundaries with joint n-gram models. In: INTERSPEECH, vol. 1, pp. 49–52 (2007). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-56149127120&partnerID=40&md5=d6c048349e00f9fa7f7afec0dc34ea84
38. Segundo, E.S., Yang, J.: Formant dynamics of Spanish vocalic sequences in related speakers : A forensic-voice-comparison investigation. Journal of Phonetics **75**, 1–26 (2019). DOI 10.1016/j.wocn.2019.04.001. URL https://doi.org/10.1016/j.wocn.2019.04.001
39. Singh, L.G., Laitonjam, L., Singh, S.R.: Automatic Syllabification for Manipuri language. In: the 26th International Conference on Computational Linguistics, pp. 349–357 (2016). URL https://www.aclweb.org/anthology/papers/C/C16/C16-1034/
40. Sun, L., Fu, S., Wang, F.: Decision tree SVM model with Fisher feature selection for speech emotion recognition. Eurasip Journal on Audio, Speech, and Music Processing **2019**(1) (2019). DOI 10.1186/s13636-018-0145-5
41. Suyanto, Harjoko, A.: Nearest neighbour-based Indonesian G2P conversion. Telkomnika (Telecommunication, Computing, Electronics, and Control) **12**(2), 389–396 (2014). DOI http://dx.doi.org/10.12928/telkomnika.v12i2.57. URL http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/57/pdf_93
42. Suyanto, S.: Flipping onsets to enhance syllabification. International Journal of Speech Technology **22**(4), 1031–1038 (2019). DOI 10.1007/s10772-019-09649-y. URL https://link.springer.com/article/10.1007/s10772-019-09649-y
43. Suyanto, S.: Incorporating syllabification points into a model of grapheme-to-phoneme conversion. International Journal of Speech Technology **22**(2), 459–470 (2019). DOI 10.1007/s10772-019-09619-4. URL https://doi.org/10.1007/s10772-019-09619-4
44. Suyanto, S., Hartati, S., Harjoko, A., Van Compernolle, D.: Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge. Speech Communication **85** (2016). DOI 10.1016/j.specom.2016.10.009
45. Tian, J.: Data-driven approaches for automatic detection of syllable boundaries. In: Proceedings of the International Conference on Spoken Language Processing (ICSLP), pp. 61–64 (2004)

Evidence of correspondence

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

1. First submission with title "Slur Words, Boost Indonesian Bigram-Syllabification" (11 August 2019)
2. LoA with Major Revision (12 December 2019)
3. Response to Reviewers, Final submission with revised title "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" (04 January 2020)
4. **LoA with Fully Accepted (07 January 2020)**
5. Final Proof Reading (22 January 2020)

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

Decision on your manuscript #IJST-D-19-00125R1

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Tue, Jan 7, 2020 at 6:18 PM
Reply-To: "International Journal of Speech Technology (IJST)" <ramya.thulasingam@springer.com>
To: Suyanto Suyanto <suyanto@telkomuniversity.ac.id>

Dear Dr. Suyanto:

We are pleased to inform you that your manuscript, "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" has been accepted for publication in International Journal of Speech Technology.

You will receive an e-mail from Springer in due course with regard to the following items:

1. Offprints
2. Colour figures

3. Transfer of Copyright

Please remember to quote the manuscript number, IJST-D-19-00125R1, whenever inquiring about your manuscript.
With best regards,
Amy Neustein, Ph.D.
Editor-in-Chief
International Journal of Speech Technology

IJST manuscript No.
(will be inserted by the editor)

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

Suyanto Suyanto

Received: date / Accepted: date

Abstract Swapping one or more consonant-graphemes in a word into other phonologically similar ones, which based on both place and manner of articulation, interestingly produces some other words without shifting the syllable boundary (or point). For examples, in the Indonesian language, swapping consonant-graphemes in a word "***ba.ra***" (embers) creates three new words: "***ba.la***" (disaster), "***pa.ra***" (reference to a group), and "***pa.la***" (nutmeg) without changing the syllabification points since both graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while both ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. An observation on 50k Indonesian words shows that replacing consonant-graphemes in those words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this paper, a procedure of swapping consonant-graphemes based on phonological similarity is proposed to boost the standard bigram-based orthographic syllabification, which commonly has a low performance for a dataset with many out-of-vocabulary (OOV) bigrams. Some examinations on the 50k words using the k-fold cross-validation scheme, with $k = 5$, prove that the proposed procedure significantly boosts the standard bigram-syllabification, where it gives a relative reduction of mean syllable error rate (SER) up to 31.39%. It also shows an improvement for the dataset of 15k named-entities by relatively decreasing the average SER by 9.53%. It is better than a flipping onsets-based model for both datasets. Compared to a nearest neighbor-based model, its performance is a little worse, but it provides much lower complexity. Another important finding is that the proposed model can produce a relatively small SER, even for a tiny training-set.

Suyanto Suyanto
School of Computing, Telkom University, Bandung, West Java 40257, Indonesia
Tel.: +62 22 7564108, Mobile: +62 812 845 12345
Orcid: https://orcid.org/0000-0002-8897-8091
E-mail: suyanto@telkomuniversity.ac.id

Keywords backoff smoothing · bigram · Indonesian language · orthographic syllabification · phonological similarity

1 Introduction

One of the important pronunciation units in a language is a syllable. It is completely relevant to the phonology rules. In [7], the researcher states that a syllable is a representational unit used to learn the phonotactic constraints of speech-sounds. The syllable is generally believed to be central to the infant as well as the adult perception of speech [32]. The syllable theories are based on much evidence. One of them is evidence from child language acquisition [11].

In linguistic theory, a syllable consists of an obligatory nucleus with or without non-obligatory surrounding consonants called onset and coda [33]. In the Indonesian language, a nucleus should be either vowel or diphthong [2]. Meanwhile, both onset and coda are consonants [2]. For instance, a word "*pantai*" (beach) contains two syllables: ⟨pan⟩ and ⟨tai⟩. The former consists of an onset ⟨p⟩, a nucleus of single vowel ⟨a⟩, and a coda ⟨n⟩. The later is composed of an onset ⟨t⟩, a nucleus of diphthong ⟨ai⟩, but no coda.

A model of syllable boundary detection, also known as automatic syllabification, is defined as a process of dividing a word into syllables. This model is urgent for some researches as well as application developments in the linguistics area, e.g. grapheme-to-phoneme conversion (G2P) [34], [41], spelling-checker [22], [28], machine transliteration [27], speech synthesis [14], [10], [3], [25], speaking rate estimation [26], speaking proficiency scoring [16], word count estimation [32], speech recognition [26], [12], [29], [17], dialect identification [21], speech emotion recognition [6], [38], forensic-voice [36], early childhood digital literacy [20], etc.

An automatic syllabification is commonly implemented using two different approaches: either orthographic or phonemic-based. The previous works show that the phonemic-syllabification [42] performs better than the orthographic one [30], but it requires a linguist to provides perfect phoneme sequences. A G2P model can be created to replace the linguist role, but a low phoneme error rate produced by the G2P can significantly decrease its performance in terms of SER [42]. Besides, it potentially performs much worse for named-entities having many exceptions and ambiguities. Therefore, many researchers are interested in the orthographic (also known as graphemic) syllabification as it is much simpler and more flexible for any dataset, primarily when they use the statistical models.

In general, an automatic syllabification is implemented using statistical models, instead of rule-based ones, since they are easier to implement and give lower SER [1]. The statistical models usually use either supervised or unsupervised learning technique, such as Näive Bayes model [4], decision tree-based model [9], random forest-based model [4], recurrent networks-based model [43], support vector machine-based model [5], finite-state transducers model [19], [15], context-free grammars model [24], Hidden Markov Model [18], syllabifica-

tion by analogy [1], dropped-and-matched model [31], n-gram model [35], conditional random fields model [37], [33], nearest neighbor-based model [42], [30], and unsupervised-syllabification model based on a classification of graphemic-symbols into two categories: consonants and vowels [23].

The nearest neighbor is one of the exciting models since it produces a low SER [42], [30]. Unfortunately, it has a high complexity of computation. It also requires complex language-specific knowledge. Besides, a graphemic encoding proposed in [30] produces a relatively high SER since it does not accurately represent a language-specific knowledge.

Another attractive model is the n-gram syllabification since it gives both competitive SER and low complexity. Besides, it is simple to implement as well as language-independent that does not require any language-specific phonotactic knowledge. Unfortunately, it has a disadvantage for a small dataset with a high rate of OOV bigrams. Many researchers have proposed various procedures to make some improvements, such as the segmental conditional random fields (SCRF) [33] and the bigram with flipping onsets (BFO) [40]. The SCRF is a bigram-based syllabification smoothed by the Stupid Backoff scheme described in [8]. It performs excellent, with a high generalization, even for quite small training-sets. Unfortunately, it looks very complex since eight features generated by sonority, legality, and maximum onset should be taken into account in calculating the bigram-probability. Meanwhile, the BFO offers a simple computation, but it produces quite high SER for the Indonesian language [40].

Therefore, a new procedure of phonological similarity-based backoff smoothing is proposed in this paper to boost the Stupid Backoff smoothed bigram-syllabification. This procedure is inspired by a fact that replacing one or more consonant-graphemes in a word into other phonologically similar ones (based on both place and manner of articulations) may create other words without shifting the syllabification points. For instance, swapping two consonant-graphemes in an Indonesian word "***b****a.****r****u*" (new) produces three new words: "***b****a.****l****u*" (widower), "***p****a.****r****u*" (lung), and "***p****a.****l****u*" (hammer) without shifting the points of syllabifications since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill/lateral-dental. This procedure increases the number of bigrams, which means that the OOV rate can be reduced.

Indonesian language has eighteen prefixes [2]. An interesting phenomenon is that swapping one or more consonant-graphemes in a prefix generally not only produces another legal prefix but also a few illegal one (noise). For instance, swapping the grapheme ⟨b⟩ in a prefix ⟨ber⟩ in the word "*be.ra.tu.ran*" (regular) into ⟨p⟩ produces another legal prefix ⟨per⟩ in "*pe.ra.tu.ran*" (rules). Swapping a prefix ⟨pe⟩ in "*pe.nam.pi.lan*" (performance) produces an OOV word "*be.nam.pi.lan*". But, all syllables in the OOV word produce three legal bigrams come from other words, i.e. "*be.nam*" that come from "*mem.be.nam*" (to immerse); "*nam.pi*" that come from "*me.nam.pi*" (winnow), "*pe.nam.pi*" (shelter), "*pe.nam.pi.lan*" (performance), and other words; and "*pi.lan*" that come from "*a.pi.lan*" (breastwork), "*kam.pi.lan*" (appearance), "*pi.pi.lan*"

(flat), "*pe.nam.pi.lan*" (performance), and many other words. Swapping a prefix ⟨ter⟩ in "*ter.ba.wa*" (not deliberately taken away) produces illegal prefix ⟨der⟩ in an OOV word "*der.ba.wa*" with a bigram "*der.ba*" that is never found in 50k words but, based on the Indonesian phonotactic rules, it is a legal bigram.

For English and other European languages, the procedure of swapping consonant-graphemes in many words may create huge illegal syllable-unigrams and syllable-bigrams. However, for the Indonesian language, the procedure generates more new legal syllable-unigrams and syllable-bigrams than the illegal ones. A preliminary study shows that 50k Indonesian words produce a total of 161,981 legal syllable-unigrams. Swapping those 50k words produces a total of 2,676,764 swapped syllable-unigrams, where 87.36% of them are legal and the rest 12.64% are unseen. It means that the swapping procedure significantly increases the number of unigrams by up to 16.52 times. Furthermore, those 50k words produce a total of 212,550 syllable-bigrams. Swapping them produces a total of 3,317,292 swapped syllable-bigrams, where 77.45% are legal and the rest 22.55% are unseen. It means that the swapping procedure impressively increases the number of bigrams by 14.12 times. Those unseen syllable-unigrams and syllable-bigrams can be either legal or illegal based on the Indonesian phonotactic rules. However, it is not easy to classify them into both classes.

In this research, the impact of swapping consonant-graphemes in a word is investigated on an Indonesian orthographic syllabification. First, the standard bigram-syllabification (BS) smoothed by the Stupid Backoff scheme [8] is implemented. Next, the combination of standard bigram-syllabification and phonological similarity-based backoff smoothing (CBSPS) is developed, and its performance is then compared to BS. Since it is not easy to detect the unseen syllable-unigrams and bigrams as legal or illegal, CBSPS is implemented using all of them (not just the legal ones). Thus, this research focuses on examining whether CBSPS can enhance the performance of BS or not.

2 Research Method

The training process of CBSPS is simply illustrated in Fig. 1. A tiny training-set is used here to make it easy to understand. Let the training-set contains only two words: "*pandai*" (smart) and "*pantai*" (beach), which are syllabified as "*pan.dai*" and "*pan.tai*", respectively. Training this dataset produces both tables of syllable-unigrams and syllable-bigrams with their frequencies (see the left side). Meanwhile, on the right side, it creates both tables of swapped syllable-unigrams (or swapped-unigrams) and swapped syllable-bigrams (or swapped-bigrams) with higher frequencies than those on the left side. Both tables swapped-unigrams and swapped-bigrams have higher frequencies since combinatorially swapping the consonant-graphemes ⟨p⟩, ⟨d⟩, and ⟨t⟩ in both original words in the training-set into ⟨b⟩, ⟨t⟩, and ⟨d⟩, respectively, produces three new words each. Combinatorially swapping the consonant-graphemes

⟨p⟩ and ⟨d⟩ in the word "*pandai*" (smart) into ⟨b⟩ and ⟨t⟩ creates three new words: "*pantai*" (beach), "*bandai*" (OOV word), and "*bantai*" (slaughter). Meanwhile, combinatorially swapping the consonant-graphemes ⟨p⟩ and ⟨t⟩ in the word "*pantai*" into ⟨b⟩ and ⟨d⟩ also produces three new words: "*pandai*", "*bantai*", and "*bandai*". Thus, there are four new unique words produced by the swapping procedure: two words "*bandai*" and "*bantai*" occur twice each while two others "*pandai*" and "*pantai*" appear once each. Those generated tables of both normal and swapped syllable-unigrams and bigrams are then exploited in the testing process.

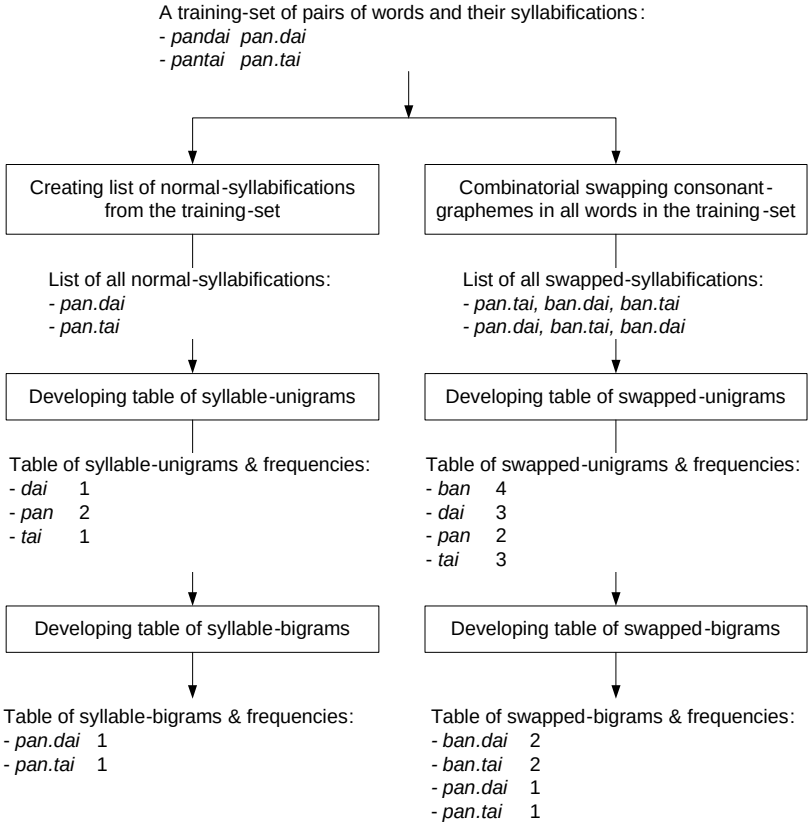


Fig. 1 Training process of CBSPS model

The testing process of CBSPS is described in Fig. 2. Let the input is an unseen word "*bantai*" (slaughter) that can be represented as a sequence of graphemes ⟨*bantai*⟩. This input is quite hard to be syllabified since ⟨ai⟩ is a diphthong, not two separate vowels ⟨a⟩ and ⟨i⟩. First, three vowels {⟨a⟩, ⟨a⟩, and ⟨i⟩} contained in the grapheme sequence are detected in the positions {2, 5, 6}. A well known high accurate method called Sukhotin's algorithm proposed

in [13] can be exploited to automatically detect vowels and diphthongs but it is not used here. Instead, this research just uses the simple Indonesian typological knowledge explained in [2], where five graphemes {⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩, ⟨u⟩} can be single vowels; four grapheme sequences {⟨ai⟩, ⟨au⟩, ⟨ei⟩, ⟨oi⟩} may produce diphthongs; and other graphemes are considered as consonants.

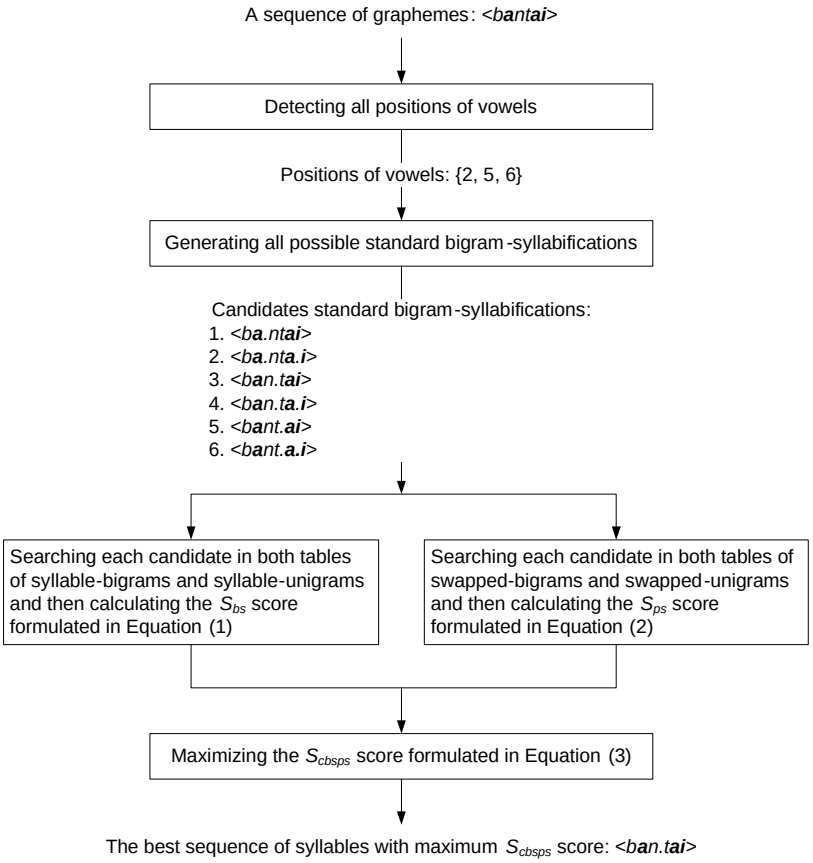


Fig. 2 Testing process of CBSPS model

All possible syllabifications (candidates) are then generated. In this case, there are six candidates: ⟨*ba.ntai*⟩, ⟨*ba.nta.i*⟩, ⟨*ban.tai*⟩, ⟨*ban.ta.i*⟩, ⟨*bant.ai*⟩, and ⟨*bant.a.i*⟩, where two graphemes ⟨*ai*⟩ may produce either a diphthong or two single vowels. After that, search each candidate in both tables of syllable-bigrams and syllable-unigrams to calculate the S_{bs} score using Equation (1) as well as in both tables of swapped-bigrams and swapped-unigrams to calculate the S_{ps} score using Equation (2). Finally, maximize the S_{cbsps} score formulated in Equation (3) to decide the best sequence of syllables, which has the highest score. In this case, the third candidate ⟨*ban.tai*⟩ has the highest score since it

may come from swapping consonant-graphemes in two other bigrams ⟨*pan.tai*⟩ (beach) and ⟨*pan.dai*⟩ (smart) while all the five rest candidates cannot come from any other bigram. Thus, CBSPS is capable of syllabifying the input sequence of graphemes ⟨*bantai*⟩ into ⟨*ban.tai*⟩, where two graphemes ⟨*ai*⟩ is correctly detected as a diphthong.

2.1 Standard bigram-syllabification

The standard bigram-syllabification (BS) model works by maximizing the likelihood of syllable sequences for an input word. The likelihood can be estimated using a probability chain, which is commonly smoothed by the Stupid Backoff scheme to produce a more accurate probability, which is here called *score* since its value can be more than 1, for a training-set with many OOV words [8]. In this method, the score of bigram-syllabification S_{bs} is calculated as

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ are the frequencies of syllable bigrams and unigrams appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N is the training-set size, and α is the factor of backoff smoothing that is generally tuned as 0.4 for many applications [8]. The model of BS commonly gives a low performance for a small training-set that has a high rate of OOV syllable [33]. Therefore, a procedure of decreasing the OOV rate can be introduced to improve the performance of BS.

2.2 Combination of standard and phonological similarity-based bigram

A procedure of swapping consonant-graphemes in the training-set is proposed here to decrease the OOV rate in the BS model. This procedure forms a new model called phonological similarity-based bigram-syllabification, which has a score S_{ps} formulated as

$$S_{ps}(w_i|w_{i-1}) = \begin{cases} B \frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} \text{ if } f_s(w_{i-1}w_i) > 0 \\ \\ U\alpha \frac{f_s(w_i)}{N_s} \text{ otherwise} \end{cases} \tag{2}$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the frequencies of both swapped-bigram and swapped-unigram appear in the training-set, w_i is the ith syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and ith syllables, N_s is the swapped training-set size, B is a weight of swapped-bigram, U is a weight of swapped-unigram, and α is the

backoff factor as used in Equation 1. Both weights B and U are introduced here to smooth the score since the swapped-consonant words may produce some illegal bigrams and/or illegal unigrams. Hence, the value of B should be less than 1.0 because swapping procedure on 50k formal words creates only 77.45% legal bigrams (the rest 22.55% are illegal bigrams). Meanwhile, the value of U is probably much lower than B since a unigram is less important than a bigram in deciding the score of syllabification.

Finally, the proposed CBSPS model uses the combined score S_{cbsps} that is simply calculated as

$$S_{cbsps} = S_{bs} + S_{ps} \tag{3}$$

where S_{bs} is the score of bigram-syllabification in Equation (1) and S_{ps} is the score of phonological similarity-based model in Equation (2).

2.3 Phonological similarity-based swapping consonant-graphemes

Table 1 illustrates 14 graphemes in the Indonesian language with their phonological similarities, which based on the categorization of phonemes described in [2], as well as their examples in some formal Indonesian words. Here, the graphemes and their swaps are simply mapped to those phoneme categorizations since they are strongly related to the corresponding phonemes [2], [39]. A formal word containing one of those 14 graphemes, which are grouped into seven categories, can be swapped to produce another formal word, as shown in the last column (Example). In [2], both phonemes /g/ and /k/ are in the same category (plosive-velar). But, they are not used here since swapping grapheme ⟨g⟩ into ⟨k⟩ commonly produces many illegal syllable-unigrams and bigrams, such as swapping consonant-graphemes in the word "*me.mang.sa*" (prey on) generates "*me.mank.sa*" (OOV) with an illegal syllable-unigram "*mank*" and two illegal bigrams "*me.mank*" and "*mank.sa*". Instead, the grapheme ⟨q⟩ is used here since it is always pronounced as a phoneme /k/ [2], [42].

Meanwhile, Table 2 illustrates the examples of some swapped-consonant words generated from the original words containing two or more possible swapping-graphemes without changing the points (or boundaries) of syllabifications. A word "***b****a.****r****a*" (embers), which has two possible swapping graphemes ⟨b⟩ and ⟨r⟩, can be combinatorially swapped to produce three new words: "***b****a.****l****a*" (disaster), "***p****a.****r****a*" (rubber), and "***p****a.****l****a*" (nutmeg) without shifting the syllabification points. A word "***b****i.****r****u*" (blue), which also has two possible swapping graphemes, can be combinatorially swapped into three new words: "***b****i.****l****u*", "***p****i.****r****u*", and "***p****i.****l****u*" without changing the syllabification points. There is no formal word "***b****i.****l****u*" in Indonesian language, which means that "***b****i.****l****u*" is an OOV word. But, it can be a sub-word for some other words, such as "*sem.****b****i.****l****u*" (sharp reed skin like a knife). The word "***p****i.****r****u*" is also an OOV word, but it is a sub-word for the word "***p****i.****r****u.et*" (one of ballet dance styles). In contrast, "***p****i.****l****u*" (really sad) is a formal word. Meanwhile, the words "***b****a.****r****a****t***" (west) and "***c****e.****r****i.****t****a*" (story), which have three possible swapping

Table 1 Consonant-graphemes and their swaps as well as the example of the swapped-consonant words without shifting their points or boundaries of syllabifications

Grapheme category	Graph.	Swap	Example
Plosive-Bilabial: {b, p}	b	p	***b****a.ru* (new) → ***p****a.ru* (lung)
	p	b	***p****a.du* (intact) → ***b****a.du* (checkered)
Plosive-Dental: {d, t}	d	t	***d****a.ri* (from) → ***t****a.ri* (dance)
	t	d	***t****a.hi* (feces) → ***d****a.hi* (forehead)
Plosive-Velar: {k, q}	k	q	***k****a.ri* (curry) → ***q****a.ri* (reciter)
	q	k	*a.****q****i.dah* (creed) → *a.****k****i.dah* (creed)
Affricative-Palatal: {c, j}	c	j	***c****a.ri* (find) → ***j****a.ri* (finger)
	j	c	***j****an.da* (widow) → ***c****an.da* (joke)
Fricative-Labiodental: {f, v}	f	v	***f****i.si* (fission) → ***v****i.si* (vision)
	v	f	***v****o.li* (volley) → ***f****o.li* (thin metal)
Fricative-Dental: {s, z}	s	z	***s****a.man* (indict) → ***z****a.man* (era)
	z	s	*a.****z****am* (aim) → *a.****s****am* (acid)
Thrill/Lateral-Dental: {l, r}	l	r	***l****i.ma* (five) → ***r****i.ma* (rhyme)
	r	l	***r****a.bu* (Wednesday) → ***l****a.bu* (pumpkin)

graphemes, can be combinatorially swapped into seven new words each. No doubt, such swapped-consonant words increase the number of unigrams as well as bigrams. Hence, swapping one or more consonant-graphemes in a word can be seen as a method of data augmentation. This is expected to produce a more accurate S_{cbsps} score in Equation (3) so that a better syllabification can be achieved.

Table 2 Examples of some new words produced by swapping consonants-graphemes in the original words without shifting the point or boundary of syllabification

Original word	Swapped-consonant words
ba.ra (embers)	*ba.la* (disaster), *pa.ra* (reference to a group), *pa.la* (nutmeg)
ba.ru (new)	*ba.lu* (widower), *pa.ru* (lung), *pa.lu* (hammer)
bi.ru (blue)	*pi.ru* (OOV), *bi.lu* (OOV), *pi.lu* (really sad)
ba.rat (west)	*ba.rad* (OOV), *ba.lat* (OOV), *ba.lad* (city), *pa.rat* (OOV), *pa.rad* (OOV), *pa.lat* (penis), *pa.lad* (OOV)
ce.ri.ta (story)	*ce.ri.da* (OOV), *ce.li.ta* (OOV), *ce.li.da* (OOV), *je.ri.ta* (OOV), *je.ri.da* (OOV), *je.li.ta* (very beautiful), *je.li.da* (OOV)

3 Result and Discussion

There are three datasets used here: formal Indonesian words, named-entities, and the mixture of both datasets, where the first two datasets are the same as used in [30]. The formal word dataset consists of 50k words equipped with boundaries (or points) of syllabifications. It is equally divided into five subsets (folds), each consists of 10k words, to do the five-fold cross-validation. The dataset of named-entities contains 15k entries with their syllable boundaries. It is also equally divided into five folds; each contains 3k words. The mixed dataset consists of 65k entries and their syllable boundaries. It is also equally divided into five folds; each contains 13k entries.

3.1 Evaluation on the dataset of formal words

In this evaluation, five experiments are conducted to tune the parameters sequentially. Firstly, the optimum unigram weight U is searched using $\alpha = 0.4$ as suggested in [8] and $B = 1.0$ based on the assumption that the swapped-bigrams have the same importance as the normal-bigrams. Secondly, the bigram weight B is then optimized using the found optimum U and $\alpha = 0.4$. Thirdly, the backoff factor α is verified using both optimum values of U and B. Fourthly, the three parameters are jointly optimized using the potential values resulted from the previous three experiments. Finally, CBSPS is fairly compared to other syllabification models. Here, a percentage of errors in the syllable level, which is commonly known as SER, is used to measure all performances in those experiments.

Optimizing unigram weight U. The proposed CBSPS is firstly evaluated using $\alpha = 0.4$ and $B = 1.0$ to find the optimum unigram weight U. The results illustrated in Fig. 3 informs that U is very sensitive. A very small $U = 0.001$ produces high SERs for all folds. A big $U = 0.1$ or bigger also gives higher SERs. The unigram weight U reaches the optimum value of 0.05 that produces the lowest SERs for all folds with the average SER of 2.65%. As hypothesized, the optimum value of this parameter is pretty low of 0.05 (much lower than B), which means that the impact of the swapped unigrams is just 5% in calculating the S_{cbsps} score in Equation (3).

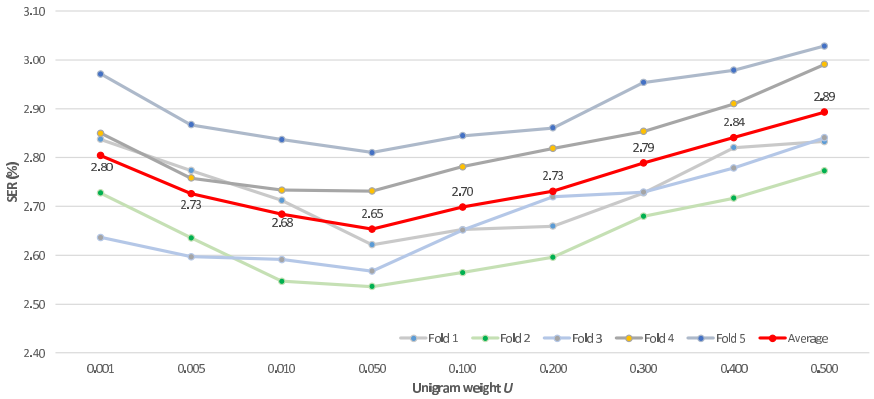


Fig. 3 SERs produced by CBSPS using $\alpha = 0.4$, $B = 1.0$, and varying unigram weight U

Optimizing bigram weight B. The proposed CBSPS is then evaluated using $\alpha = 0.4$ and $U = 0.05$ to optimize the bigram weight B. The results in Fig. 4 shows that B is not sensitive. It is quite stable to produce low SERs for all

folds when the value is in the interval of 0.8 to 1.1. It reaches the optimum value of 1.0 that produces the lowest average SER of 2.65%.

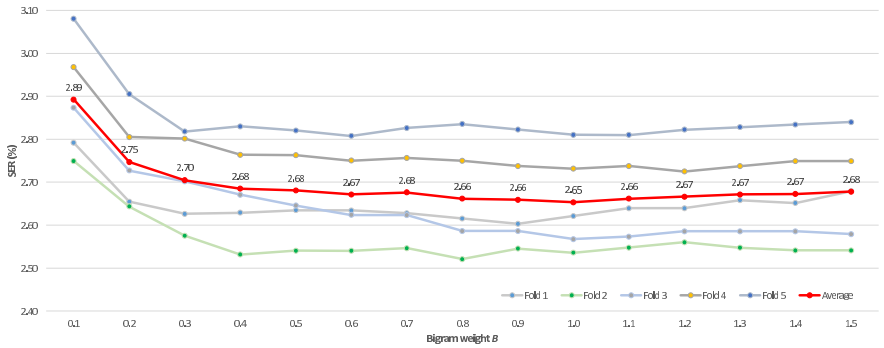


Fig. 4 SERs produced by CBSPS using $\alpha = 0.4$, $U = 0.05$, and varying bigram weight B

Verifying backoff factor α. Next, the use of $\alpha = 0.4$ suggested in [8] is verified using both optimum values $U = 0.05$ and $B = 1.0$. Here, nine experiments are performed using $\alpha = 0.1$ to 0.9. The results in Fig. 5 informs that α is an easily tuned parameter. It gives the lowest average SER of 2.65% when the value is in the interval of 0.2 to 0.4. It means that the $\alpha = 0.4$ suggested in [8] is also suitable for CBSPS.

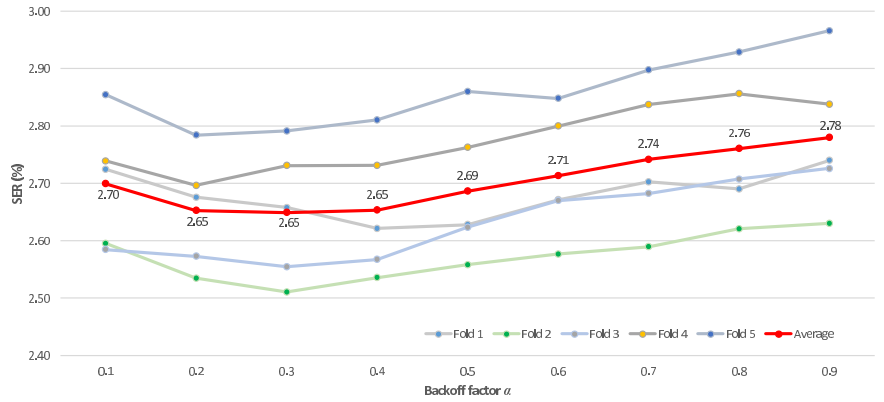


Fig. 5 SERs produced by CBSPS using $U = 0.05$, $B = 1.0$, and varying α

Jointly parameters optimization. Next, the three parameters are then jointly optimized using the potential values resulted from the previous sequential tunings, i.e. $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 6 shows that the optimum combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.2$ that gives the lowest average SER of 2.61%.

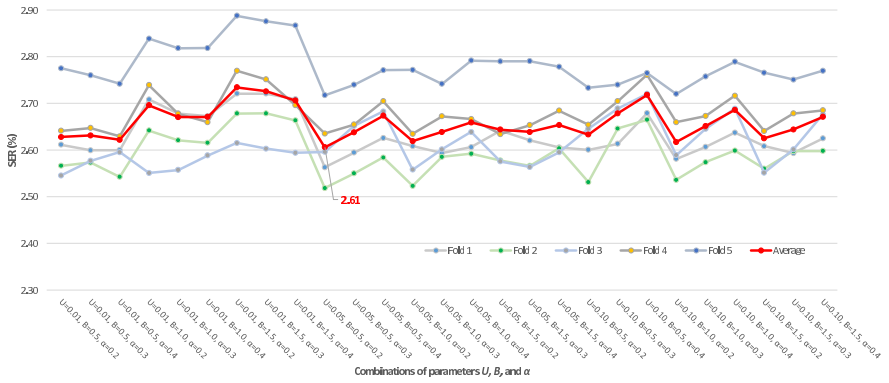


Fig. 6 SERs produced by CBSPS using jointly parameters optimization for the dataset of formal words

Comparison to other models. Finally, the best performance of CBSPS is compared to three other syllabification models: BS, BFO, and the fuzzy-based k-nearest neighbor model called FkNNC described in [30]. All models are compared in their best performances for the same dataset of 50k formal Indonesian words in [30] to get fairness. The results in Fig. 7 inform that CBSPS produces a lower average SER of 2.61% than BS with an average SER of 3.80%. Hence, CBSPS gives a relative reduction of mean SER up to 31.39%. It proves that CBSPS is capable of significantly boosting the BS model. CBSPS is also better than BFO (with average SER of 3.11%), which means it decreases the mean SER by 16.08%. However, it is slightly worse than FkNNC, which reaches the lowest mean SER of 2.27%.

Nevertheless, CBSPS has a much lower complexity than FkNNC since it just calculates the probabilities of bigrams while FkNNC should find the k nearest neighbors to define the syllabification points. CBSPS just searches tens or fewer bigrams as well as unigrams that are taken into account to calculate the S_{cbsps} score, where the searching can be very fast using both indexed-sorted bigrams and unigrams. Meanwhile, FkNNC [30] should compute up to 250 thousand distances between a candidate syllabification and all impossible indexed-sorted patterns in the training-set, choose the k closest patterns in both classes (a point and not a point of syllabification), and eventually select the class with the lowest total fuzzy-distance as the decision.

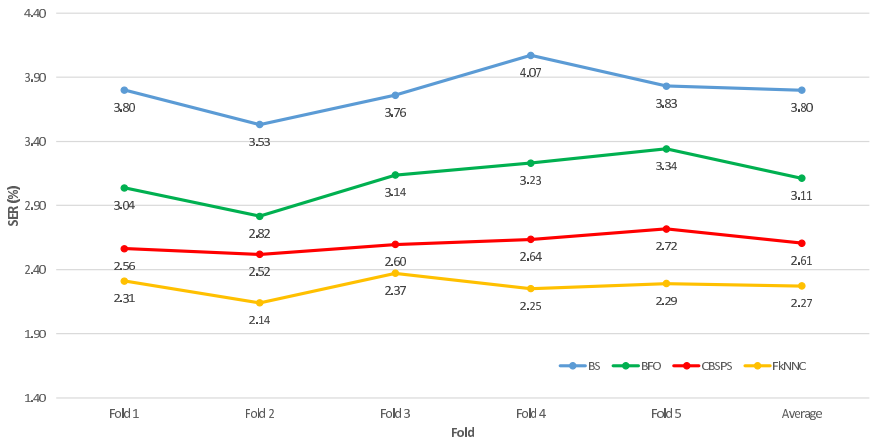


Fig. 7 SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of formal words

3.2 Evaluation on the dataset of named-entities

The proposed CBSPS just uses both syllable bigrams and unigrams as well as their phonological similarities to maximize the scores of syllabifications. Hence, it can be applied to a dataset of named-entities since the phonological similarities are very common in this dataset. For example, mapping two graphemes ⟨b⟩ and ⟨d⟩ in a named-entity "**b**an.**d**ung" (the capital city in West Java) into their phonological similarities ⟨p⟩ and ⟨t⟩ produces three other named-entities: "**b**an.**t**ung" (a resort in Sukhothai, Thailand), "**p**an.**d**ung" (a village in Special Region of Yogyakarta), and "**p**an.**t**ung" (a folk song from Bolaang Mongondow, North Sulawesi).

Careful observation on the dataset of 15k named-entities informs that it produces a total of 45,799 legal syllable-unigrams. Swapping procedure on those 15k named-entities produces a total of 385,850 swapped syllable-unigrams, where 90.92% of them are legal and the rest 9.08% are unseen. Hence, the swapping procedure significantly increases the number of unigrams by up to 8.43 times. Furthermore, those 15k named-entities create a total of 30,516 syllable-bigrams. Swapping them produces a total of 275,472 swapped syllable-bigrams, where 84.61% of them are legal and the rest 15.39% are unseen. It means that the swapping procedure impressively increases the number of bigrams by up to 9.03 times. These facts imply that the proposed CBSPS will be better than BS in syllabifying the named-entities. Therefore, CBSPS is evaluated here using this dataset of named-entities in a 5-fold cross-validation scheme. First, the three parameters U, B, and α are jointly optimized. The SER produced by the optimum values of those parameters is then compared to three other models: BS, BFO, and FkNNC.

Jointly parameters optimization. The three parameters are jointly optimized using the potential values resulted from the previous experiment on the dataset of formal words, where $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 8 concludes that the optimum combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.3$ that gives the lowest average SER of 13.48%.

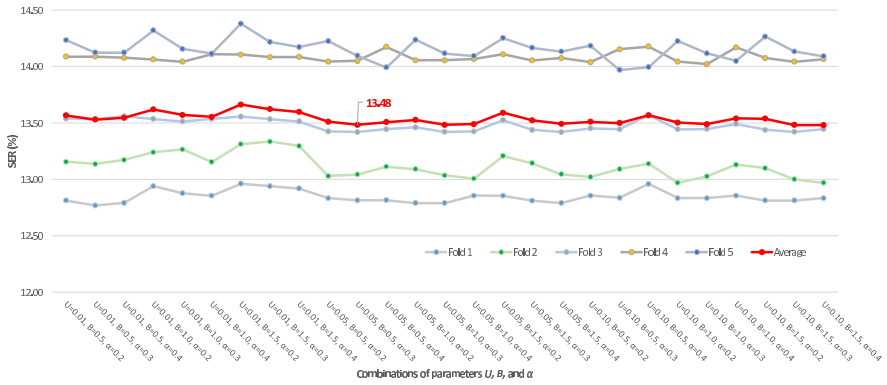


Fig. 8 SERs produced by CBSPS using jointly parameters optimization for the dataset of named-entities

Comparison to other models. The best performance of CBSPS is then compared to three other syllabification models BS, BFO, and FkNNC using the same dataset of 15k named-entities described in [30]. All models are compared in their best performances to get fairness. The results in Fig. 9 show that CBSPS produces a lower average SER (13.48%) than BS (14.90%), which means that it relatively decreases the mean SER by 9.53%. It is also better than BFO (14.15%) by relatively reducing the average SER by 4.83%. However, it is much worse than FkNNC, which reaches the lowest mean SER of 6.78%. This result is caused by much vowel ambiguity in the named-entities. For instances, the person names "A.dy", "A.di", and "A.dhie" are pronounced as /ɑ.di/ and the person names "Bu.dy", "Bu.di", and "Bu.dhie" are pronounced as /bu.di/. Since CBSPS always considers the grapheme ⟨y⟩ as a consonant, not a semi-vowel nor a vowel, it fails to syllabify "Budy" into "Bu.dy".

3.3 Evaluation on the mixed dataset of formal words and named-entities

The proposed CBSPS is finally evaluated using the mixed dataset of 50k formal words and 15k named-entities to see its generalization. This dataset of 65k entries is equally divided into five folds; each contains 13k entries. First, the

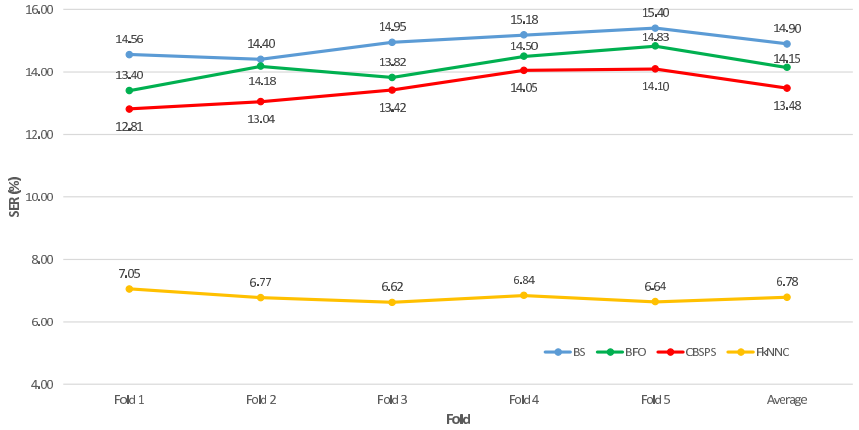


Fig. 9 SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of named-entities

three parameters of CBSPS are jointly optimized. Its best performance is then compared to both BS and BFO models. Unfortunately, it cannot be compared to the FkNNC since there is no experimental result for this mixed dataset provided in [30].

Jointly parameters optimization. The three parameters are jointly optimized using the potential values resulted from the previous experiments, i.e. $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 10 show that the best combination of the three parameters is $U = 0.05$, $B = 0.5$, and $\alpha = 0.2$, which gives the lowest average SER of 4.88%. Further investigation indicates that the SER produced by the named-entities is slightly lower than the previous model (trained using the named-entities only), but the SER from the formal words does not decrease at all. It means that CBSPS is capable of generalizing bigrams from the formal words into the named-entities, but not vice versa.

Comparison to other models. The best performance of CBSPS is then compared to both BS and BFO using the mixed dataset of 65k words. The results in Fig. 11 show that CBSPS produces smaller average SER (4.88%) than both BS (6.02%) and BFO (5.74%), which means that it gives relative reductions of 18.94% and 14.98%, respectively. The results also show that the performance of CBSPS is stable for all five folds.

3.4 Evaluation on the training-set sizes

First, some different sized training-sets are developed by randomly selecting words from the five folds in the dataset of formal words. Each fold is defined

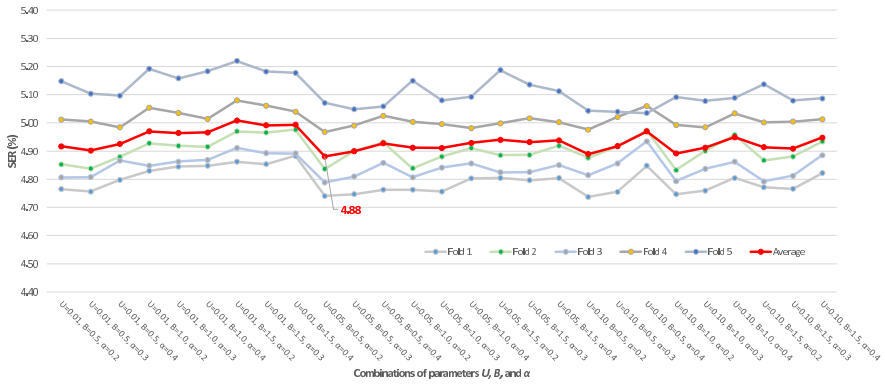


Fig. 10 SERs produced by CBSPS using jointly parameters optimization for the mixed dataset of formal words and named-entities

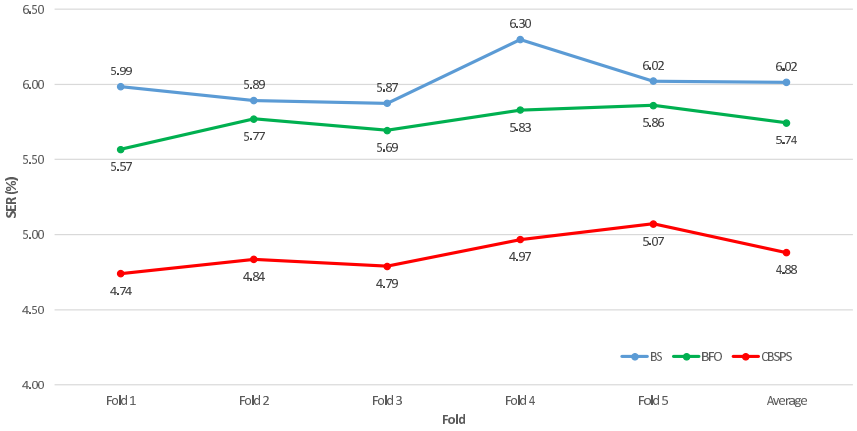


Fig. 11 SERs produced by BS, BFO, and CBSPS models for the mixed dataset of formal words and named-entities

as the fixed testing-set. Next, six training-sets of 1k, 5k, 10k, 20k, 30k, and 40k are then randomly generated five times from the remaining four folds (not in the testing-sets) so that each size of training-set contains five subsets. The comparisons of BS, BFO, and CBSPS are performed repeatedly five times (once for each subset), and the average SERs are then calculated.

The experimental results in Fig. 12 shows that CBSPS, for most training-set sizes, gives lower average SERs than both BS and BFO. For the training-set size of 1k, both CBSPS and BFO give the same average SER of 16.13%, which is smaller than BS (19.30%). For the training-set of 5k, CBSPS produces the lowest mean SER of 6.37% among BFO (7.01%) and BS (8.93%). The exciting results come from the training-set of 10k, where CBSPS yields impressively

lower mean SER (3.68%) than both BFO (4.96%) and BS (6.29%). For the training-set of 20k, CBSPS and BFO yield the same mean SER of 3.94% that is smaller than BS (4.68%). The most exciting results come from the training-set of 30k, where CBSPS reaches a much lower average SER (1.57%) than both BFO (3.32%) and BS (4.09%). It means that CBSPS gives the relative reductions of the average SER by up to 52.71% and 61.61%, respectively. Finally, for the training-set of 40k, CBSPS also gives the smallest mean SER of 2.61% among BFO (3.31%) and BS (3.80%).

These results are fascinating. Increasing the size of the training-set does not always decrease the SER. Sometimes it raises the SER. It can be said that CBSPS is not stable. This fact can be easily explained here that swapped-consonant words producing many illegal OOV bigrams and unigrams potentially increase the SER. In other words, the unstable SERs produced by CBSPS are caused by the illegal OOV swapped-bigrams and swapped-unigrams generated from the training-sets. Therefore, a scheme of filtering legal bigrams and unigrams can be introduced to enhance CBSPS.

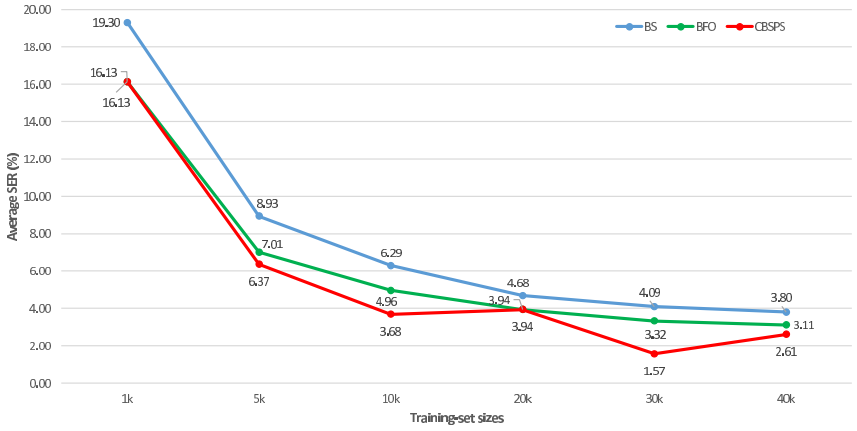


Fig. 12 Average SERs produced by BS, BFO, and CBSPS for six different sized training-sets taken from the 50k formal Indonesian words

4 Conclusion

The proposed CBSPS model is capable of significantly boosting the standard bigram-syllabification (BS) model for the dataset of 50k formal words with a relative reduction of SER up to 31.39%. It is also better than BFO by relatively reducing the mean SER by 16.08%. However, it is slightly worse than FkNNC, but it offers a much lower complexity. Nevertheless, CBSPS can give a relatively low SER, even for a tiny training-set, since it exploits

a swapping graphemes-based data augmentation that significantly increases the number of bigrams and unigrams. For the dataset of 15k named-entities, CBSPS is also better than both BS and BFO with relative reductions of SER by 9.53% and 4.83%, respectively. However, it is worse than FkNNC since it is hard to solve much ambiguity of vowel in the named-entities. In the future, a scheme of filtering legal bigrams and unigrams can be introduced to improve its performance.

Acknowledgements

I want to give a high appreciation to Muhammad Agha Ariyanto, a three-year son, for his great inspiration in speaking words by swapping one or more consonants into similar ones based on both place and manner of articulation.

References

1. Adsett, C.R., Marchand, Y., Kešelj, V.: Syllabification rules versus data-driven methods in a language with low syllabic complexity: The case of Italian. Computer Speech and Language **23**, 444–463 (2009). DOI 10.1016/j.csl.2009.02.004
2. Alwi, H., Lapoliwa, H., Darmowidjojo, S.: Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3 edn. Balai Pustaka (2003)
3. Aripin, Haryanto, H., Sumpeno, S.: 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 **13**(8), 19–37 (2018). DOI 10.3991/ijet.v13i08.8084
4. Balc, D., Beleiu, A., Potolea, R., Lemnaru, C.: A learning-based approach for Romanian syllabification and stress assignment. In: P. R. (ed.) Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015, pp. 37–42. Institute of Electrical and Electronics Engineers Inc. (2015). DOI 10.1109/ICCP.2015.7312603
5. Bartlett, S., Kondrak, G., Cherry, C.: 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, pp. 308–316. Boulder, Colorado (2009). DOI 10.3115/1620754.1620799
6. Ben Alex, S., Babu, B.P., Mary, L.: Utterance and syllable level prosodic features for automatic emotion recognition. In: 2018 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2018, pp. 31–35. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/RAICS.2018.8635059. URL https://ieeexplore.ieee.org/document/8635059
7. Bernard, A.: An onset is an onset: Evidence from abstraction of newly-learned phonotactic constraints. Journal of Memory and Language **78**, 18–32 (2015). DOI https://doi.org/10.1016/j.jml.2014.09.001. URL http://www.sciencedirect.com/science/article/pii/S0749596X1400103X
8. Brants, T., Popat, A.C., Och, F.J.: 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, pp. 858–867 (2007). URL https://www.aclweb.org/anthology/D07-1090
9. Daelemans, W., Bosch, A.V.D., Weijters, T.: IGTree: Using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review **11**(1-5), 407–423 (1997). DOI 10.1.1.29.4517

10. Faldessai, N., Pawar, J., Naik, G.: Syllabification: An effective approach for a TTS system for Konkani. In: 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, ICEECCOT 2016, pp. 161–167. Institute of Electrical and Electronics Engineers Inc. (2017). DOI 10.1109/ICEECCOT.2016.7955207. URL https://ieeexplore.ieee.org/document/7955207
11. Fallows, D.: Experimental evidence for English syllabification and syllable structure. Journal of Linguistics **17**(2), 309–317 (1981). DOI 10.1017/S0022226700007027
12. Feng, S., Lee, T.: Exploiting Cross-Lingual Speaker and Phonetic Diversity for Unsupervised Subword Modeling. IEEE/ACM Transactions on Audio Speech and Language Processing **27**(12), 2000–2011 (2019). DOI 10.1109/TASLP.2019.2937953. URL https://ieeexplore.ieee.org/document/8818297
13. Foster, C.C.: A Comparison of Vowel Identification Methods. Cryptologia **16**(3), 282–286 (1992). DOI 10.1080/0161-119291866955. URL https://doi.org/10.1080/0161-119291866955
14. Geeta, S., Muralidhara, B.L.: 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, pp. 1205–1208. Institute of Electrical and Electronics Engineers Inc. (2018). DOI 10.1109/WiSPNET.2017.8299954. URL https://ieeexplore.ieee.org/document/8299954
15. Hlaing, T.H., Mikami, Y.: Automatic Syllable Segmentation of Myanmar Texts using Finite State Transducer. International Journal on Advances in ICT for Emerging Regions (ICTer) **6**(2), 2–9 (2014). DOI 10.4038/icter.v6i2.7150
16. Johnson, D.O., Kang, O.: Comparison of algorithms to divide noisy phone sequences into syllables for automatic unconstrained English speaking proficiency scoring. Artificial Intelligence Review pp. 1–24 (2017). DOI 10.1007/s10462-017-9594-y
17. Kamper, H., Jansen, A., Goldwater, S.: A segmental framework for fully-unsupervised large-vocabulary speech recognition. Computer Speech & Language **46**, 154–174 (2017). DOI https://doi.org/10.1016/j.csl.2017.04.008. URL http://www.sciencedirect.com/science/article/pii/S0885230816301905
18. Krantz, J., Dulin, M., De Palma, P., VanDam, M.: Syllabification by Phone Categorization. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, pp. 47–48. ACM, New York, NY, USA (2018). DOI 10.1145/3205651.3208781. URL http://doi.acm.org/10.1145/3205651.3208781
19. Krisnawati, L.D., Mahastama, A.W.: A Javanese Syllabifier Based on its Orthographic System. In: S.H.R.A.B.M.A.N.E.A.L.R.N.A.B.P. Dong M. Ruskanda F.Z. (ed.) International Conference on Asian Language Processing, pp. 244–249. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/IALP.2018.8629173. URL https://ieeexplore.ieee.org/document/8629173
20. Kulju, P., Mäkinen, M.: Phonological strategies and peer scaffolding in digital literacy game-playing sessions in a Finnish pre-primary class. Journal of Early Childhood Literacy (2019). DOI 10.1177/1468798419838576. URL https://journals.sagepub.com/doi/pdf/10.1177/1468798419838576
21. Leemann, A., Kolly, M.J., Nolan, F., Li, Y.: The role of segments and prosody in the identification of a speaker's dialect. Journal of Phonetics **68**, 69–84 (2018). DOI https://doi.org/10.1016/j.wocn.2018.02.001. URL http://www.sciencedirect.com/science/article/pii/S0095447016300365
22. Magdum, D., Suman, M.: System for identifying and correcting invalid words in the devanagari script for text to speech engine. International Journal of Innovative Technology and Exploring Engineering **8**(6 Special Issue 4), 1001–1006 (2019). DOI 10.35940/ijitee.F1206.0486S419
23. Mayer, T.: 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, pp. 63–71 (2010)
24. Müller, K.: 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, pp. 11–20 (2006). DOI 10.3115/1622165.1622167

25. Mulyanto, E., Yuniarno, E.M., Purnomo, M.H.: Adding an Emotions Filter to Javanese Text-to-Speech System. In: 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia, CENIM 2018 - Proceeding, pp. 142–146. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/CENIM.2018.8711229
26. Nayak, S., Bhati, S., Rama Murty, K.S.: 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, pp. 6590–6594. Institute of Electrical and Electronics Engineers Inc. (2019). DOI 10.1109/ICASSP.2019.8683462. URL https://ieeexplore.ieee.org/document/8683462
27. Ngo, G.H., Nguyen, M., Chen, N.F.: Phonology-Augmented Statistical Framework for Machine Transliteration Using Limited Linguistic Resources. IEEE/ACM Transactions on Audio Speech and Language Processing **27**(1), 199–211 (2019). DOI 10.1109/TASLP.2018.2875269. URL https://ieeexplore.ieee.org/document/8488567
28. Oncevay-marcos, A.: Spell-Checking based on Syllabification and Character-level Graphs for a Peruvian Agglutinative Language. In: The First Workshop on Subword and Character Level Models in NLP, pp. 109–116 (2017)
29. Pakoci, E., Popović, B., Pekar, D.: Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition. Computational Intelligence and Neuroscience **2019** (2019). DOI 10.1155/2019/5072918
30. Parande, E.A.: Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. International Journal of Speech Technology **22**(1), 13–20 (2019). DOI 10.1007/s10772-018-09569-3. URL https://link.springer.com/article/10.1007/s10772-018-09569-3
31. Ramli, I., Jamil, N., Seman, N., Ardi, N.: An Improved Syllabification for a Better Malay Language Text-to- Speech Synthesis (TTS). Procedia - Procedia Computer Science **76**(Iris), 417–424 (2015). DOI 10.1016/j.procs.2015.12.280. URL http://dx.doi.org/10.1016/j.procs.2015.12.280
32. Räsänen, O., Doyle, G., Frank, M.C.: Pre-linguistic segmentation of speech into syllable-like units. Cognition **171**, 130–150 (2018). DOI https://doi.org/10.1016/j.cognition.2017.11.003. URL http://www.sciencedirect.com/science/article/pii/S0010027717302901
33. Rogova, K., Demuynck, K., Compernolle, D.V.: Automatic syllabification using segmental conditional random fields. Computational Linguistics in the Netherlands Journal **3**, 34–48 (2013). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907935327&partnerID=40&md5=ec7496ee067bb9d2aca02f45d90d6bd0
34. Rugchatjaroen, A., Saychum, S., Kongyoung, S., Chootrakool, P., Kasuriya, S., Wutiwiwatchai, C.: Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion. Speech Communication **106**, 105–111 (2019). DOI https://doi.org/10.1016/j.specom.2018.12.003. URL http://www.sciencedirect.com/science/article/pii/S0167639317303965
35. Schmid, H., Möbius, B., Weidenkaff, J.: Tagging syllable boundaries with joint n-gram models. In: INTERSPEECH, vol. 1, pp. 49–52 (2007). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-56149127120&partnerID=40&md5=d6c048349e00f9fa7f7afec0dc34ea84
36. Segundo, E.S., Yang, J.: Formant dynamics of Spanish vocalic sequences in related speakers : A forensic-voice-comparison investigation. Journal of Phonetics **75**, 1–26 (2019). DOI 10.1016/j.wocn.2019.04.001. URL https://doi.org/10.1016/j.wocn.2019.04.001
37. Singh, L.G., Laitonjam, L., Singh, S.R.: Automatic Syllabification for Manipuri language. In: the 26th International Conference on Computational Linguistics, pp. 349–357 (2016). URL https://www.aclweb.org/anthology/papers/C/C16/C16-1034/
38. Sun, L., Fu, S., Wang, F.: Decision tree SVM model with Fisher feature selection for speech emotion recognition. Eurasip Journal on Audio, Speech, and Music Processing **2019**(1) (2019). DOI 10.1186/s13636-018-0145-5
39. Suyanto, Harjoko, A.: Nearest neighbour-based Indonesian G2P conversion. Telkomnika (Telecommunication, Computing, Electronics, and Control) **12**(2), 389–396 (2014). DOI http://dx.doi.org/10.12928/telkomnika.v12i2.57. URL http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/57/pdf_93

40. Suyanto, S.: Flipping onsets to enhance syllabification. International Journal of Speech Technology **22**(4), 1031–1038 (2019). DOI 10.1007/s10772-019-09649-y. URL https://link.springer.com/article/10.1007/s10772-019-09649-y
41. Suyanto, S.: Incorporating syllabification points into a model of grapheme-to-phoneme conversion. International Journal of Speech Technology **22**(2), 459–470 (2019). DOI 10.1007/s10772-019-09619-4. URL https://doi.org/10.1007/s10772-019-09619-4
42. Suyanto, S., Hartati, S., Harjoko, A., Compernolle, D.V.: Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge. Speech Communication **85**, 109–118 (2016). DOI 10.1016/j.specom.2016.10.009. URL https://www.sciencedirect.com/science/article/pii/S016763931630005X
43. Van Esch, D., Chua, M., Rao, K.: Predicting pronunciations with syllabification and stress with recurrent neural networks. In: M.N.N.S.M.F. Morgan N. Georgiou P. (ed.) Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 08-12-Sept, pp. 2841–2845. International Speech and Communication Association (2016). DOI 10.21437/Interspeech.2016-1419. URL https://www.isca-speech.org/archive/Interspeech_2016/pdfs/1419.PDF

Evidence of correspondence

Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection

1. First submission with title "Slur Words, Boost Indonesian Bigram-Syllabification" (11 August 2019)
2. LoA with Major Revision (12 December 2019)
3. Response to Reviewers, Final submission with revised title "Phonological Similarity-Based Backoff Smoothing to Boost a Bigram Syllable Boundary Detection" (04 January 2020)
4. LoA with Fully Accepted (07 January 2020)
5. **Final Proof Reading (22 January 2020)**

Query Details Back to Main Page

1. Figures 1, 2, 6, and 7 mismatch between source and reference pdf source pdf followed. Author confirm if processed figures are correct.

No, the processed Figures 1, 2, 6, and 7 are NOT correct. The Figures 1, 2, 6, and 7 should be replaced with the new figures from the files in the attachment (I attach), where the detail explanations are as follow:

1. Please replace Figure 1 with the new figure from the file "BlockTrainingCBSPS.eps" in the attachment because the new figure gives a more detail and clearer illustration for the fundamental concept of the proposed model.
2. Please replace Figure 2 with the new figure from the file "BlockTestingCBSPS.eps" in the attachment because the new figure provides a more detail and clearer illustration for the fundamental concept of the proposed model.
3. Please replace Figure 6 with the new figure from the file "SERJointlyUBAlphaFW.eps" in the attachment since both axis and axis-title in the old figure are not suitable for the text (the word "Alpha" should be changed into a symbol "α").
4. Please replace Figure 7 with the new figure from the file "SERComparisonFW.eps" in the attachment since there is a typo in the legend of the old figure ("FkKNC" should be changed into "FkNNC").

Phonological similarity-based backoff smoothing to boost a bigram syllable boundary detection

Suyanto Suyanto, [1]✉

Email suyanto@telkomuniversity.ac.id

[1] School of Computing, Telkom University, Bandung, West Java, 40257 Indonesia

Received: 11 August 2019 / Accepted: 13 January 2020

Abstract

Swapping one or more consonant-graphemes in a word into other phonologically similar ones, which based on both place and manner of articulation, interestingly produces some other words without shifting the syllable boundary (or point). For examples, in the Indonesian language, swapping consonant-graphemes in a word "***b****a.* ***r****a*" (embers) creates three new words: "***b****a.* ***l****a*" (disaster), "***p****a.* ***r****a*" (reference to a group), and "***p****a.* ***l****a*" (nutmeg) without changing the syllabification points since both graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while both ⟨r⟩ and ⟨l⟩ are thrill Please change "thrill" into "trill" /lateral-dental. An observation on 50k Indonesian words shows that replacing consonant-graphemes in those words impressively increases the number of unigrams by 16.52 times and significantly increases the number of bigrams by 14.12 times. Therefore, in this paper, a procedure of swapping consonant-graphemes based on phonological similarity is proposed to boost the standard bigram-based orthographic syllabification, which commonly has a low performance for a dataset with many out-of-vocabulary (OOV) bigrams. Some examinations on the 50k words using the *k*-fold cross-validation scheme, with $k = 5$, prove that the proposed procedure significantly boosts the standard bigram-syllabification, where it gives a relative reduction of mean syllable error rate (SER) up to 31.39%. It also shows an improvement for the dataset of 15k named-entities by relatively decreasing the average SER by 9.53%. It is better than a flipping onsets-based model for both datasets. Compared to a nearest neighbor-based model, its performance is a little worse, but it provides much lower complexity. Another important finding is that the proposed model can produce a relatively small SER, even for a tiny training-set.

Keywords

Backoff smoothing
Bigram
Indonesian language
Orthographic syllabification
Phonological similarity

1. Introduction

One of the important pronunciation units in a language is a syllable. It is completely relevant to the phonology rules. In Bernard (2015), the researcher states that a syllable is a representational unit used to learn the phonotactic constraints of speech-sounds. The syllable is generally believed to be central to the infant as well as the adult perception of speech (Räsänen et al. 2018). The syllable theories are based on much evidence. One of them is evidence from child language acquisition (Fallows 1981).

In linguistic theory, a syllable consists of an obligatory nucleus with or without non-obligatory surrounding consonants called onset and coda (Rogova et al. 2013). In the Indonesian language, a nucleus should be either vowel or diphthong (Alwi et al. 2003). Meanwhile, both onset and coda are consonants (Alwi et al. 2003). For instance, a word "*pantai*" (beach) contains two syllables: ⟨pan⟩ and ⟨tai⟩. The former consists of an onset ⟨p⟩, a nucleus of single vowel ⟨a⟩, and a coda ⟨n⟩. The later is composed of an onset ⟨t⟩, a nucleus of diphthong ⟨ai⟩, but no coda.

A model of syllable boundary detection, also known as automatic syllabification, is defined as a process of dividing a word into syllables. This model is urgent for some researches as well as application developments in the linguistics area, e.g. grapheme-to-phoneme conversion (G2P) (Rugchatjaroen et al. 2019; Suyanto 2019b), spelling-checker (Magdum and Suman 2019; Oncevay-Marcos 2017), machine transliteration (Ngo et al. 2019), speech synthesis (Aripin et al. 2018; Faldessai et al. 2017; Geeta and Muralidhara 2018; Mulyanto et al. 2019), speaking rate estimation (Nayak et al. 2019), speaking proficiency scoring (Johnson and Kang 2017), word count estimation (Räsänen et al. 2018), speech recognition (Feng and Lee 2019; Kamper et al. 2017; Nayak et al. 2019; Pakoci et al. 2019), dialect identification (Leemann et al. 2018), speech emotion recognition (Ben Alex et al. 2019; Sun et al. 2019), forensic-voice (Segundo and Yang 2019), early childhood digital literacy (Kulju and Mäkinen 2019), etc.

An automatic syllabification is commonly implemented using two different approaches: either orthographic or phonemic-based. The previous works show that the phonemic-syllabification (Suyanto et al. 2016) performs better than the orthographic one (Parande 2019), but it requires a linguist to provides perfect phoneme sequences. A G2P model can be created to replace the linguist role, but a low phoneme error rate produced by the G2P can significantly decrease its performance in terms of SER (Suyanto et al. 2016). Besides, it potentially performs much worse for named-entities having

many exceptions and ambiguities. Therefore, many researchers are interested in the orthographic (also known as graphemic) syllabification as it is much simpler and more flexible for any dataset, primarily when they use the statistical models.

In general, an automatic syllabification is implemented using statistical models, instead of rule-based ones, since they are easier to implement and give lower SER (Adsett et al. 2009). The statistical models usually use either supervised or unsupervised learning technique, such as Näive Bayes model (Balc et al. 2015), decision tree-based model (Daelemans et al. 1997), random forest-based model (Balc et al. 2015), recurrent networks-based model (Van Esch et al. 2016), support vector machine-based model (Bartlett et al. 2009), finite-state transducers model (Hlaing and Mikami 2014; Krisnawati and Mahastama 2019), context-free grammars model (Müller 2006), Hidden Markov Model (Krantz et al. 2018), syllabification by analogy (Adsett et al. 2009), dropped-and-matched model (Ramli et al. 2015), *n*-gram model (Schmid et al. 2007), conditional random fields model (Rogova et al. 2013; Singh et al. 2016), nearest neighbor-based model (Parande 2019; Suyanto et al. 2016), and unsupervised-syllabification model based on a classification of graphemic-symbols into two categories: consonants and vowels (Mayer 2010).

The nearest neighbor is one of the exciting models since it produces a low SER (Parande 2019; Suyanto et al. 2016). Unfortunately, it has a high complexity of computation. It also requires complex language-specific knowledge. Besides, a graphemic encoding proposed in Parande (2019) produces a relatively high SER since it does not accurately represent a language-specific knowledge.

Another attractive model is the *n*-gram syllabification since it gives both competitive SER and low complexity. Besides, it is simple to implement as well as language-independent that does not require any language-specific phonotactic knowledge. Unfortunately, it has a disadvantage for a small dataset with a high rate of OOV bigrams. Many researchers have proposed various procedures to make some improvements, such as the segmental conditional random fields (SCRF) (Rogova et al. 2013) and the bigram with flipping onsets (BFO) (Suyanto 2019a). The SCRF is a bigram-based syllabification smoothed by the Stupid Backoff scheme described in Brants et al. (2007). It performs excellent, with a high generalization, even for quite small training-sets. Unfortunately, it looks very complex since eight features generated by sonority, legality, and maximum onset should be taken into account in calculating the bigram-probability. Meanwhile, the BFO offers a simple computation, but it produces quite high SER for the Indonesian language (Suyanto 2019a).

Therefore, a new procedure of phonological similarity-based backoff smoothing is proposed in this paper to boost the Stupid Backoff smoothed bigram-syllabification. This procedure is inspired by a fact that replacing one or more consonant-graphemes in a word into other phonologically similar ones (based on both place and manner of articulations) may create other words without shifting the syllabification points. For instance, swapping two consonant-graphemes in an Indonesian word "***b****a*.***r****u*" (new) produces three new words: "***b****a*.***l****u*" (widower), "***p****a*.***r****u*" (lung), and "***p****a*.***l****u*" (hammer) without shifting the points of syllabifications since the graphemes ⟨b⟩ and ⟨p⟩ are in the same category of plosive-bilabial while ⟨r⟩ and ⟨l⟩ are thrill Please change "thrill" into "trill" /lateral-dental. This procedure increases the number of bigrams, which means that the OOV rate can be reduced.

Indonesian language has eighteen prefixes (Alwi et al. 2003). An interesting phenomenon is that swapping one or more consonant-graphemes in a prefix generally not only produces another legal prefix but also a few illegal one (noise). For instance, swapping the grapheme ⟨b⟩ in a prefix ⟨ber⟩ in the word "*be.ra.tu.ran*" (regular) into ⟨p⟩ produces another legal prefix ⟨per⟩ in "*pe.ra.tu.ran*" (rules). Swapping a prefix ⟨pe⟩ in "*pe.nam.pi.lan*" (performance) produces an OOV word "*be.nam.pi.lan*". But, all syllables in the OOV word produce three legal bigrams come from other words, i.e. "*be.nam*" that come from "*mem.be.nam*" (to immerse); "*nam.pi*" that come from "*me.nam.pi*" (winnow), "*pe.nam.pi*" (shelter), "*pe.nam.pi.lan*" (performance), and other words; and "*pi.lan*" that come from "*a.pi.lan*" (breastwork), "*kam.pi.lan*" (appearance), "*pi.pi.lan*" (flat), "*pe.nam.pi.lan*" (performance), and many other words. Swapping a prefix ⟨ter⟩ in "*ter.ba.wa*" (not deliberately taken away) produces illegal prefix ⟨der⟩ in an OOV word "*der.ba.wa*" with a bigram "*der.ba*" that is never found in 50k words but, based on the Indonesian phonotactic rules, it is a legal bigram.

For English and other European languages, the procedure of swapping consonant-graphemes in many words may create huge illegal syllable-unigrams and syllable-bigrams. However, for the Indonesian language, the procedure generates more new legal syllable-unigrams and syllable-bigrams than the illegal ones. A preliminary study shows that 50k Indonesian words produce a total of 161,981 legal syllable-unigrams. Swapping those 50k words produces a total of 2,676,764 swapped syllable-unigrams, where 87.36% of them are legal and the rest 12.64% are unseen. It means that the swapping procedure significantly increases the number of unigrams by up to 16.52 times. Furthermore, those 50k words produce a total of 212,550 syllable-bigrams. Swapping them produces a total of 3,317,292 swapped syllable-bigrams, where 77.45% are legal and the rest 22.55% are unseen. It means that the swapping procedure impressively increases the number of

bigrams by 14.12 times. Those unseen syllable-unigrams and syllable-bigrams can be either legal or illegal based on the Indonesian phonotactic rules. However, it is not easy to classify them into both classes.

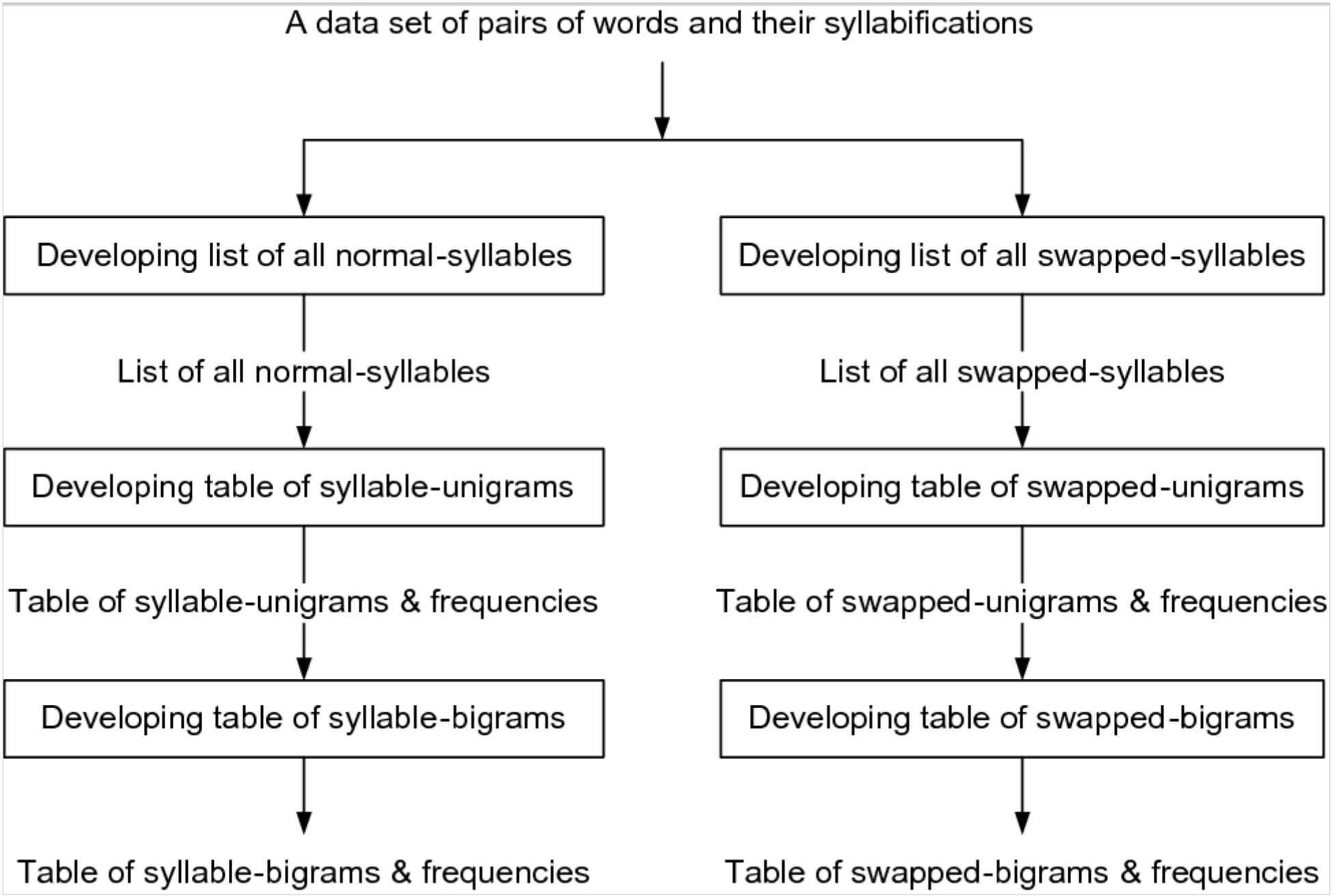
In this research, the impact of swapping consonant-graphemes in a word is investigated on an Indonesian orthographic syllabification. First, the standard bigram-syllabification (BS) smoothed by the Stupid Backoff scheme (Brants et al. 2007) is implemented. Next, the combination of standard bigram-syllabification and phonological similarity-based backoff smoothing (CBSPS) is developed, and its performance is then compared to BS. Since it is not easy to detect the unseen syllable-unigrams and bigrams as legal or illegal, CBSPS is implemented using all of them (not just the legal ones). Thus, this research focuses on examining whether CBSPS can enhance the performance of BS or not.

2. Research method

The training process of CBSPS is simply illustrated in Fig. 1. A tiny training-set is used here to make it easy to understand. Let the training-set contains only two words: "*pandai*" (smart) and "*pantai*" (beach), which are syllabified as "*pan.dai*" and "*pan.tai*", respectively. Training this dataset produces both tables of syllable-unigrams and syllable-bigrams with their frequencies (see the left side). Meanwhile, on the right side, it creates both tables of swapped syllable-unigrams (or swapped-unigrams) and swapped syllable-bigrams (or swapped-bigrams) with higher frequencies than those on the left side. Both tables swapped-unigrams and swapped-bigrams have higher frequencies since combinatorially swapping the consonant-graphemes ⟨p⟩, ⟨d⟩, and ⟨t⟩ in both original words in the training-set into ⟨b⟩, ⟨t⟩, and ⟨d⟩, respectively, produces three new words each. Combinatorially swapping the consonant-graphemes ⟨p⟩ and ⟨d⟩ in the word "*pandai*" (smart) into ⟨b⟩ and ⟨t⟩ creates three new words: "*pantai*" (beach), "*bandai*" (OOV word), and "*bantai*" (slaughter). Meanwhile, combinatorially swapping the consonant-graphemes ⟨p⟩ and ⟨t⟩ in the word "*pantai*" into ⟨b⟩ and ⟨d⟩ also produces three new words: "*pandai*", "*bantai*", and "*bandai*". Thus, there are four new unique words produced by the swapping procedure: two words "*bandai*" and "*bantai*" occur twice each while two others "*pandai*" and "*pantai*" appear once each. Those generated tables of both normal and swapped syllable-unigrams and bigrams are then exploited in the testing process.

Fig. 1

Training process of CBSPS model

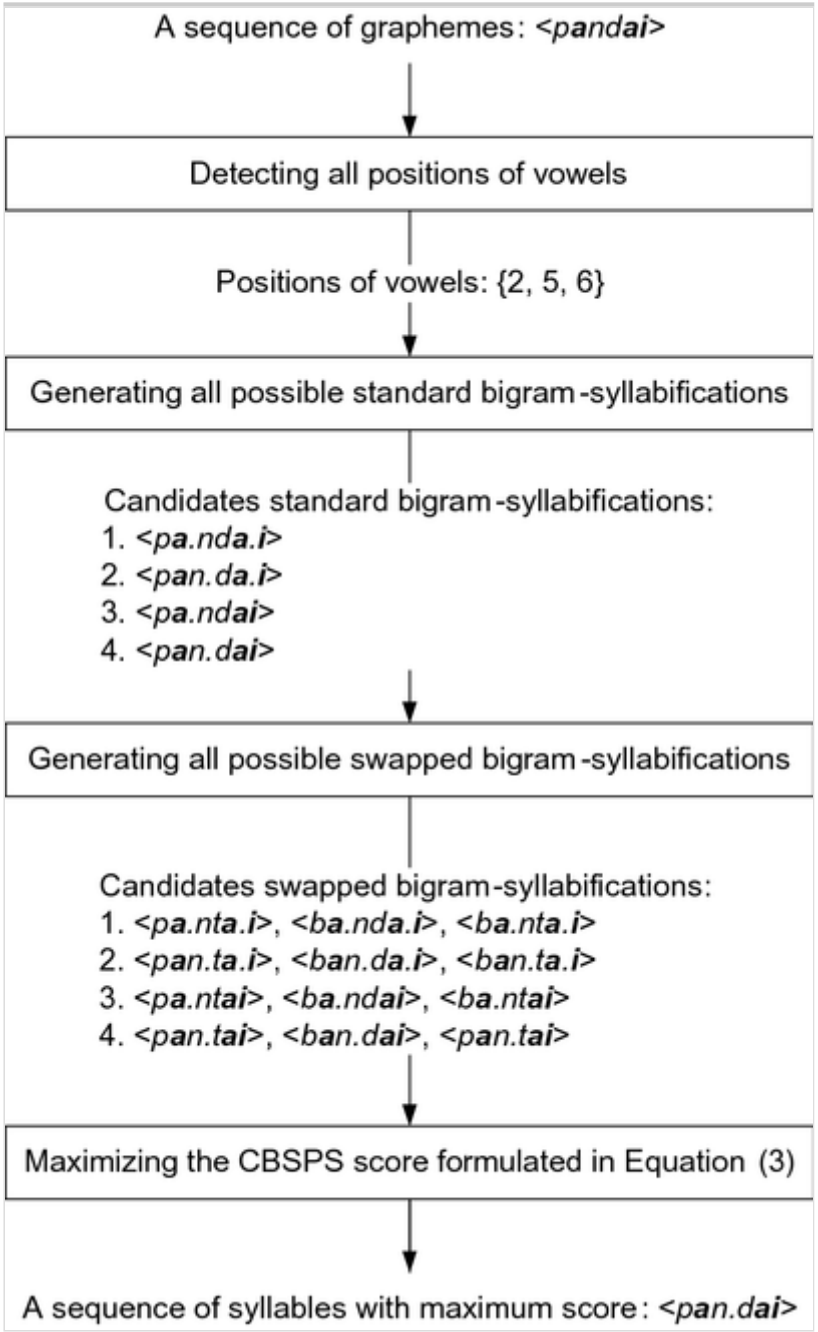

A data set of pairs of words and their syllabifications
Developing list of all normal-syllables
Developing list of all swapped-syllables
List of all normal-syllables
List of all swapped-syllables
Developing table of syllable-unigrams
Developing table of swapped-unigrams
Table of syllable-unigrams & frequencies
Table of swapped-unigrams & frequencies
Developing table of syllable-bigrams
Developing table of swapped-bigrams
Table of syllable-bigrams & frequencies
Table of swapped-bigrams & frequencies

AQ1

The testing process of CBSPS is described in Fig. 2. Let the input is an unseen word "*bantai*" (slaughter) that can be represented as a sequence of graphemes $\langle bantai \rangle$. This input is quite hard to be syllabified since $\langle \mathrm{ai} \rangle$ is a diphthong, not two separate vowels $\langle \mathrm{a} \rangle$ and $\langle \mathrm{i} \rangle$. First, three vowels $\{\langle \mathrm{a} \rangle, \langle \mathrm{a} \rangle,$ and $\langle \mathrm{i} \rangle\}$ contained in the grapheme sequence are detected in the positions $\{2, 5, 6\}$. A well known high accurate method called Sukhotin's algorithm proposed in Foster (1992) can be exploited to automatically detect vowels and diphthongs but it is not used here. Instead, this research just uses the simple Indonesian typological knowledge explained in Alwi et al. (2003), where five graphemes $\{\langle \mathrm{a} \rangle, \langle \mathrm{e} \rangle, \langle \mathrm{i} \rangle, \langle \mathrm{o} \rangle, \langle \mathrm{u} \rangle\}$ can be single vowels; four grapheme sequences $\{\langle \mathrm{ai} \rangle, \langle \mathrm{au} \rangle, \langle \mathrm{ei} \rangle, \langle \mathrm{oi} \rangle\}$ may produce diphthongs; and other graphemes are considered as consonants.

Fig. 2

Testing process of CBSPS model


A sequence of graphemes: <pandai>
Detecting all positions of vowels
Positions of vowels: {2, 5, 6}
Generating all possible standard bigram-syllabifications
Candidates standard bigram-syllabifications:
1. <pa.nda.i>
2. <pan.da.i>
3. <pa.ndai>
4. <pan.dai>
Generating all possible swapped bigram-syllabifications
Candidates swapped bigram-syllabifications:
1. <pa.nta.i>, <ba.nda.i>, <ba.nta.i>
2. <pan.ta.i>, <ban.da.i>, <ban.ta.i>
3. <pa.ntai>, <ba.ndai>, <ba.ntai>
4. <pan.tai>, <ban.dai>, <pan.tai>
Maximizing the CBSPS score formulated in Equation (3)
A sequence of syllables with maximum score: <pan.dai>

All possible syllabifications (candidates) are then generated. In this case, there are six candidates: $\langle ba.ntai \rangle$, $\langle ba.nta.i \rangle$, $\langle ban.tai \rangle$, $\langle ban.ta.i \rangle$, $\langle bant.ai \rangle$, and $\langle bant.a.i \rangle$, where two graphemes $\langle ai \rangle$ may produce either a diphthong or two single vowels. After that, search each candidate in both tables of syllable-bigrams and syllable-unigrams to calculate the S_{bs} score using Eq. (1) as well as in both tables of swapped-bigrams and swapped-unigrams to calculate the S_{ps} score using Eq. (2). Finally, maximize the S_{cbsps} score formulated in Eq. (3) to decide the best sequence of syllables, which has the highest score. In this case, the third candidate $\langle ban.tai \rangle$ has the highest score since it may come from swapping consonant-graphemes in two other bigrams $\langle pan.tai \rangle$ (beach) and $\langle pan.dai \rangle$ (smart) while all the five rest candidates cannot come from any other bigram. Thus, CBSPS is capable of syllabifying the input sequence of graphemes $\langle bantai \rangle$ into $\langle ban.tai \rangle$, where two graphemes $\langle ai \rangle$ is correctly detected as a diphthong.

2.1. Standard bigram-syllabification

The standard bigram-syllabification (BS) model works by maximizing the likelihood of syllable sequences for an input word. The likelihood can be estimated using a probability chain, which is commonly smoothed by the Stupid Backoff scheme to produce a more accurate probability, which is here called *score* since its value can be more than 1, for a training-set with many OOV words (Brants et al. 2007). In this method, the score of bigram-syllabification S_{bs} is calculated as

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ are the frequencies of syllable bigrams and unigrams appear in the training-set, w_i is the *i*th syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both $(i-1)$th and *i*th syllables, *N* is the training-set size, and α is the factor of backoff smoothing that is generally tuned as 0.4 for many applications (Brants et al. 2007). The model of BS commonly gives a low performance for a small training-set that has a high rate of OOV syllable (Rogova et al. 2013). Therefore, a procedure of decreasing the OOV rate can be introduced to improve the performance of BS.

2.2. Combination of standard and phonological similarity-based bigram

A procedure of swapping consonant-graphemes in the training-set is proposed here to decrease the OOV rate in the BS model. This procedure forms a new model called phonological similarity-based bigram-syllabification, which has a score S_{ps} formulated as

$$S_{ps}(w_i|w_{i-1}) = \begin{cases} B\frac{f_s(w_{i-1}w_i)}{f_s(w_{i-1})} & \text{if } f_s(w_{i-1}w_i) > 0 \\ U\alpha\frac{f_s(w_i)}{N_s} & \text{otherwise} \end{cases} \quad (2)$$

where $f_s(w_{i-1}w_i)$ and $f_s(w_i)$ are the frequencies of both swapped-bigram and swapped-unigram appear in the training-set, w_i is the *i*th syllable contained in a word that can be seen as a unigram while $w_{i-1}w_i$ is a bigram containing both ($i-1$)th and *i*th syllables, N_s is the swapped training-set size, *B* is a weight of swapped-bigram, *U* is a weight of swapped-unigram, and α is the backoff factor as used in Eq. 1. Both weights *B* and *U* are introduced here to smooth the score since the swapped-consonant words may produce some illegal bigrams and/or illegal unigrams. Hence, the value of *B* should be less than 1.0 because swapping procedure on 50k formal words creates only 77.45% legal bigrams (the rest 22.55% are illegal bigrams). Meanwhile, the value of *U* is probably much lower than *B* since a unigram is less important than a bigram in deciding the score of syllabification.

Finally, the proposed CBSPS model uses the combined score S_{cbsps} that is simply calculated as

$$S_{cbsps} = S_{bs} + S_{ps} \quad (3)$$

where S_{bs} is the score of bigram-syllabification in Eq. (1) and S_{ps} is the score of phonological similarity-based model in Eq. (2).

2.3. Phonological similarity-based swapping consonant-graphemes

Table 1 illustrates 14 graphemes in the Indonesian language with their phonological similarities, which based on the categorization of phonemes described in Alwi et al. (2003), as well as their examples in some formal Indonesian words.

Here, the graphemes and their swaps are simply mapped to those phoneme categorizations since they are strongly related to the corresponding phonemes (Alwi et al. 2003; Suyanto and Harjoko 2014). A formal word containing one of those 14 graphemes, which are grouped into seven categories, can be swapped to produce another formal word, as shown in the last column (Example). In Alwi et al. (2003), both phonemes /g/ and /k/ are in the same category (plosive-velar). But, they are not used here since swapping grapheme ⟨g⟩ into ⟨k⟩ commonly produces many illegal syllable-unigrams and bigrams, such as swapping consonant-graphemes in the word "*me.mang.sa*" (prey on) generates "*me.mank.sa*" (OOV) with an illegal syllable-unigram "*mank*" and two illegal bigrams "*me.mank*" and "*mank.sa*". Instead, the grapheme ⟨q⟩ is used here since it is always pronounced as a phoneme /k/ (Alwi et al. 2003; Suyanto et al. 2016).

Table 1

Consonant-graphemes and their swaps as well as the example of the swapped-consonant words without shifting their points or boundaries of syllabifications

Grapheme category	Graph.	Swap	Example
Plosive-Bilabial: {b, p}	b	p	***b****a.ru* (new) → ***p****a.ru* (lung)
	p	b	***p****a.du* (intact) → ***b****a.du* (checkered)
Plosive-Dental: {d, t}	d	t	***d****a.ri* (from) → ***t****a.ri* (dance)
	t	d	***t****a.hi* (feces) → ***d****a.hi* (forehead)
Plosive-Velar: {k, q}	k	q	***k****a.ri* (curry) → ***q****a.ri* (reciter)
	q	k	*a.* ***q****i.dah* (creed) → *a.* ***k****i.dah* (creed)
Affricative-Palatal: {c, j}	c	j	***c****a.ri* (find) → ***j****a.ri* (finger)
	j	c	***j****an.da* (widow) → ***c****an.da* (joke)
Fricative-Labiodental: {f, v}	f	v	***f****i.si* (fission) → ***v****i.si* (vision)
	v	f	***v****o.li* (volley) → ***f****o.li* (thin metal)
Fricative-Dental: {s, z}	s	z	***s****a.man* (indict) → ***z****a.man* (era)

Grapheme category	Graph.	Swap	Example
	z	s	*a* ***.z****am* (aim) → *a.****s****am* (acid)
Thrill Please change "Thrill" into "Trill" /Lateral-Dental: {l, r}	l	r	***l****i.ma* (five) → ***r****i.ma* (rhyme)
	r	l	***r****a.bu* (Wednesday) → ***l****a.bu* (pumpkin)

Meanwhile, Table 2 illustrates the examples of some swapped-consonant words generated from the original words containing two or more possible swapping-graphemes without changing the points (or boundaries) of syllabifications. A word "***b****a. **r****a*" (embers), which has two possible swapping graphemes ⟨b⟩ and ⟨r⟩, can be combinatorially swapped to produce three new words: "***b****a. **l****a*" (disaster), "***p****a. **r****a*" (rubber), and "***p****a. **l****a*" (nutmeg) without shifting the syllabification points. A word "***b****i. **r****u*" (blue), which also has two possible swapping graphemes, can be combinatorially swapped into three new words: "***b****i. **l****u*", "***p****i. **r****u*", and "***p****i* ***.l****u*" without changing the syllabification points. There is no formal word "***b****i. **l****u*" in Indonesian language, which means that "***b****i. **l****u*" is an OOV word. But, it can be a sub-word for some other words, such as "*sem. **b****i. **l****u*" (sharp reed skin like a knife). The word "***p****i. **r****u*" is also an OOV word, but it is a sub-word for the word "***p****i. **r****u.et*" (one of ballet dance styles). In contrast, "***p****i. **l****u*" (really sad) is a formal word. Meanwhile, the words "***b****a. **r****a*** ***t***" (west) and "***c****e. **r****i. **t****a*" (story), which have three possible swapping graphemes, can be combinatorially swapped into seven new words each. No doubt, such swapped-consonant words increase the number of unigrams as well as bigrams. Hence, swapping one or more consonant-graphemes in a word can be seen as a method of data augmentation. This is expected to produce a more accurate S_{cbsps} score in Eq. (3) so that a better syllabification can be achieved.

Table 2

Examples of some new words produced by swapping consonants-graphemes in the original words without shifting the point or boundary of syllabification

Original word	Swapped-consonant words
ba.ra (embers)	*ba.la* (disaster), *pa.ra* (reference to a group), *pa.la* (nutmeg)

Original word	Swapped-consonant words
ba.ru (new)	*ba.lu* (widower), *pa.ru* (lung), *pa.lu* (hammer)
bi.ru (blue)	*pi.ru* (OOV), *bi.lu* (OOV), *pi.lu* (really sad)
ba.rat (west)	*ba.rad* (OOV), *ba.lat* (OOV), *ba.lad* (city), *pa.rat* (OOV), *pa.rad* (OOV), *pa.lat* (penis), *pa.lad* (OOV)
ce.ri.ta (story)	*ce.ri.da* (OOV), *ce.li.ta* (OOV), *ce.li.da* (OOV), *je.ri.ta* (OOV), *je.ri.da* (OOV), *je.li.ta* (very beautiful), *je.li.da* (OOV)

3. Result and discussion

There are three datasets used here: formal Indonesian words, named-entities, and the mixture of both datasets, where the first two datasets are the same as used in Parande (2019). The formal word dataset consists of 50k words equipped with boundaries (or points) of syllabifications. It is equally divided into five subsets (folds), each consists of 10k words, to do the five-fold cross-validation. The dataset of named-entities contains 15k entries with their syllable boundaries. It is also equally divided into five folds; each contains 3k words. The mixed dataset consists of 65k entries and their syllable boundaries. It is also equally divided into five folds; each contains 13k entries.

3.1. Evaluation on the dataset of formal words

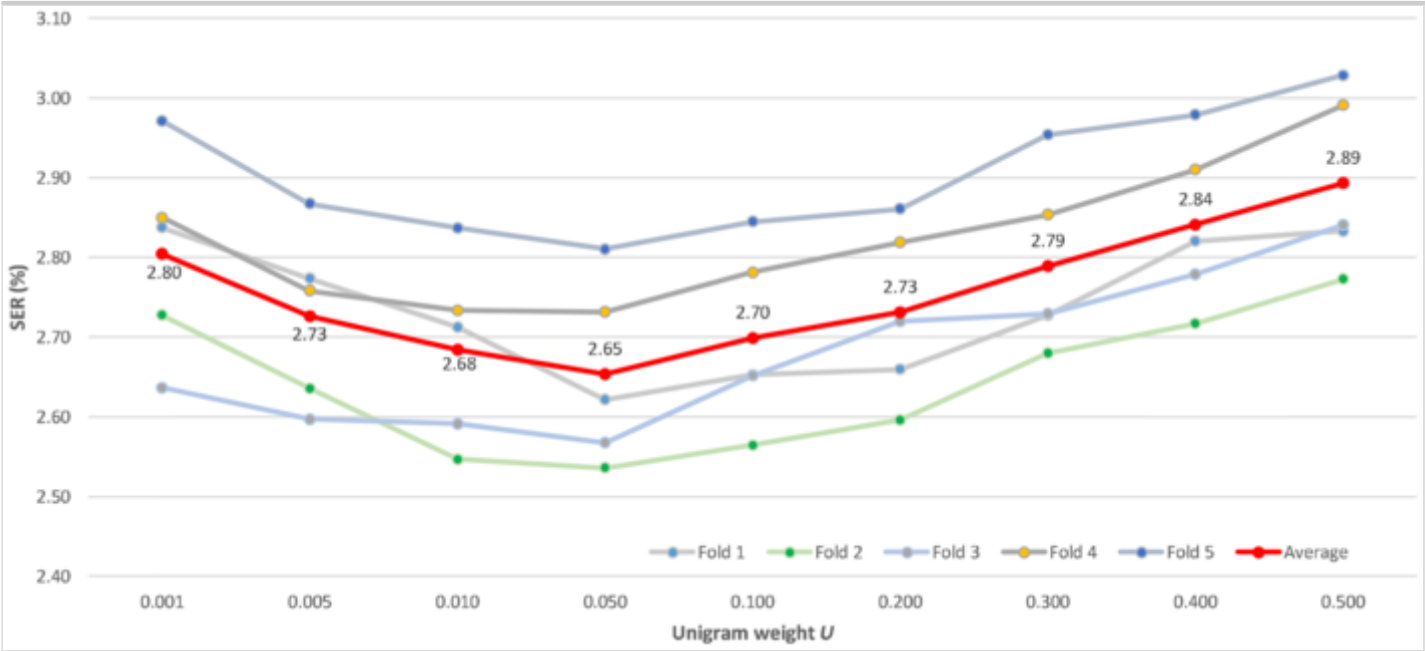
In this evaluation, five experiments are conducted to tune the parameters sequentially. Firstly, the optimum unigram weight U is searched using $\alpha = 0.4$ as suggested in Brants et al. (2007) and $B = 1.0$ based on the assumption that the swapped-bigrams have the same importance as the normal-bigrams. Secondly, the bigram weight B is then optimized using the found optimum U and $\alpha = 0.4$. Thirdly, the backoff factor α is verified using both optimum values of U and B. Fourthly, the three parameters are jointly optimized using the potential values resulted from the previous three experiments. Finally, CBSPS is fairly compared to other syllabification models. Here, a percentage of errors in the syllable level, which is commonly known as SER, is used to measure all performances in those experiments.

3.1.1. Optimizing unigram weight U

The proposed CBSPS is firstly evaluated using $\alpha = 0.4$ and $B = 1.0$ to find the optimum unigram weight U. The results illustrated in Fig. 3 informs that U is very sensitive. A very small $U = 0.001$ produces high SERs for all folds. A big $U = 0.1$ or bigger also gives higher SERs. The unigram weight U reaches the optimum value of 0.05 that produces the lowest SERs for all folds with the average SER of 2.65%. As hypothesized, the optimum value of this parameter is pretty low of 0.05 (much lower than B), which means that the impact of the swapped unigrams is just 5% in calculating the S_{cbsps} score in Eq. (3).

Fig. 3

SERs produced by CBSPS using $\alpha = 0.4,\ B = 1.0,$ and varying unigram weight U

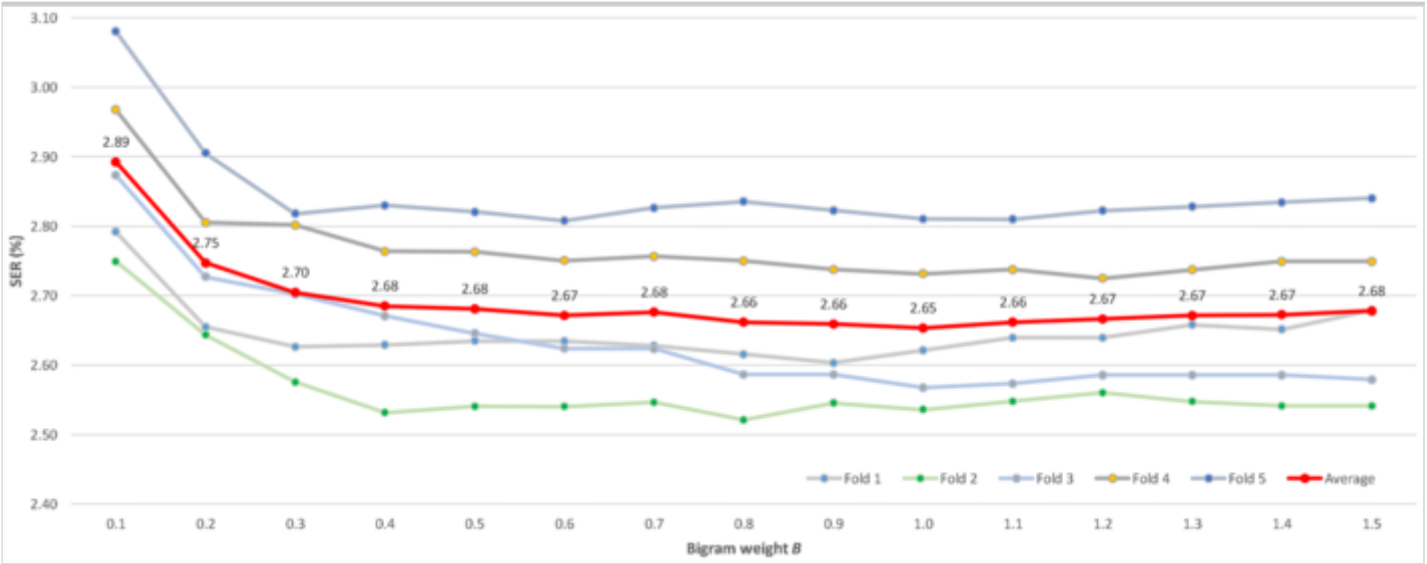


3.1.2. Optimizing bigram weight B

The proposed CBSPS is then evaluated using $\alpha = 0.4$ and $U = 0.05$ to optimize the bigram weight B. The results in Fig. 4 shows that B is not sensitive. It is quite stable to produce low SERs for all folds when the value is in the interval of 0.8 to 1.1. It reaches the optimum value of 1.0 that produces the lowest average SER of 2.65%.

Fig. 4

SERs produced by CBSPS using $\alpha = 0.4,\ U = 0.05,$ and varying bigram weight B

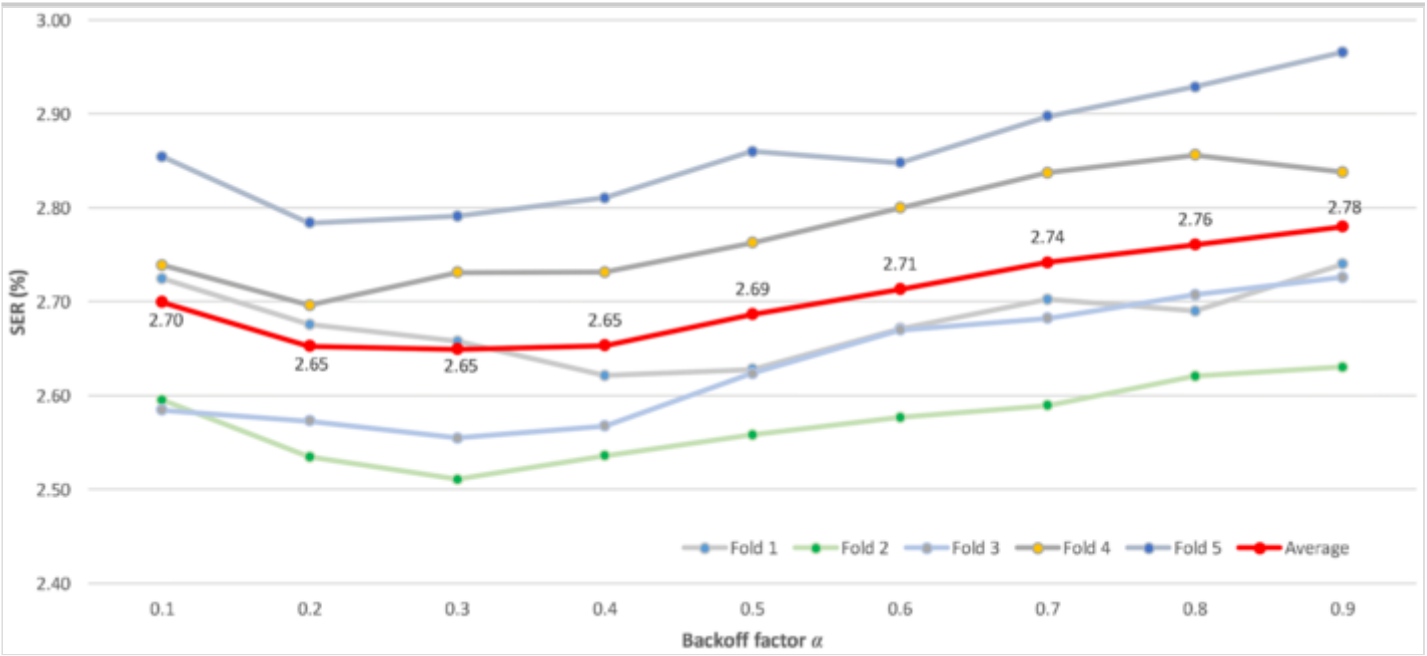


3.1.3. Verifying backoff factor α

Next, the use of $\alpha = 0.4$ suggested in Brants et al. (2007) is verified using both optimum values $U = 0.05$ and $B = 1.0$. Here, nine experiments are performed using $\alpha = 0.1$ to 0.9. The results in Fig. 5 informs that α is an easily tuned parameter. It gives the lowest average SER of 2.65% when the value is in the interval of 0.2 to 0.4. It means that the $\alpha = 0.4$ suggested in Brants et al. (2007) is also suitable for CBSPS.

Fig. 5

SERs produced by CBSPS using $U = 0.05,\ B = 1.0,$ and varying α

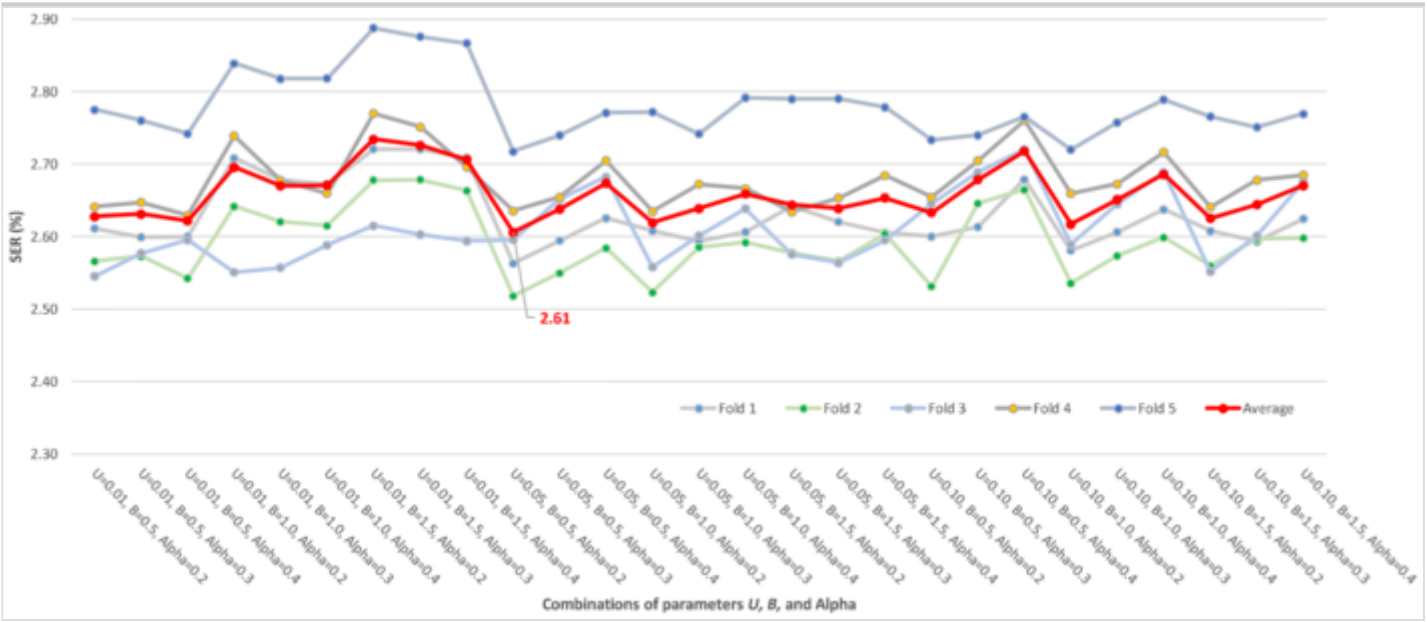


3.1.4. Jointly parameters optimization

Next, the three parameters are then jointly optimized using the potential values resulted from the previous sequential tunings, i.e. $U = \{0.05, 0.10, 0.15\},\ B = \{0.5, 1.0, 1.5\},$ and $\alpha = \{0.2, 0.3, 0.4\}.$ The results in Fig. 6 shows that the optimum combination of the three parameters is $U = 0.05,\ B = 0.5,$ and $\alpha = 0.2$ that gives the lowest average SER of 2.61%.

Fig. 6

SERs produced by CBSPS using jointly parameters optimization for the dataset of formal words



3.1.5. Comparison to other models

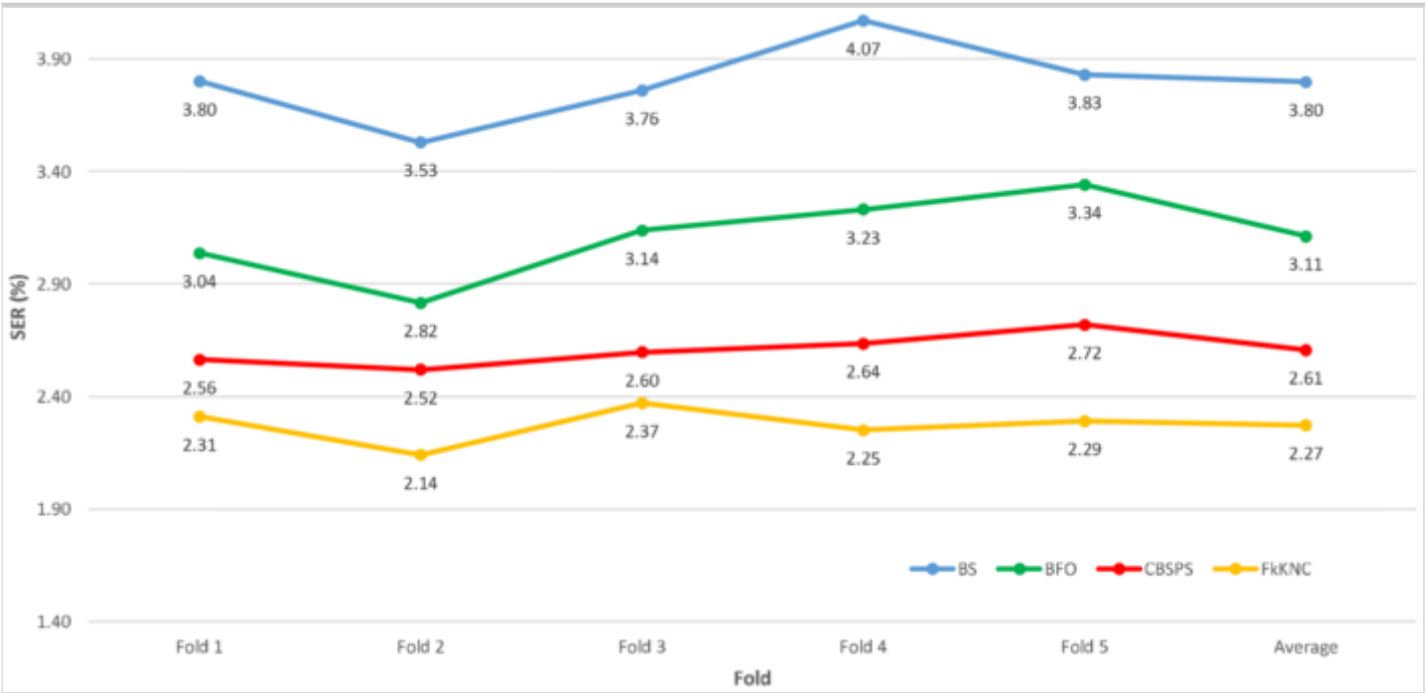
Finally, the best performance of CBSPS is compared to three other syllabification models: BS, BFO, and the fuzzy-based *k*-nearest neighbor model called FkNNC described in Parande (2019). All models are compared in their best performances for the same dataset of 50k formal Indonesian words in Parande (2019) to get fairness. The results in Fig. 7 inform that CBSPS produces a lower average SER of 2.61% than BS with an average SER of 3.80%. Hence, CBSPS gives a relative reduction of mean SER up to 31.39%. It proves that CBSPS is capable of significantly boosting the BS model. CBSPS is also better than BFO (with average SER of 3.11%), which means it decreases the mean SER by 16.08%. However, it is slightly worse than FkNNC, which reaches the lowest mean SER of 2.27%.

Nevertheless, CBSPS has a much lower complexity than FkNNC since it just calculates the probabilities of bigrams while FkNNC should find the *k* nearest neighbors to define the syllabification points. CBSPS just searches tens or fewer bigrams as well as unigrams that are taken into account to calculate the S_{cbsps} score, where the searching can be very fast using both indexed-sorted bigrams and unigrams. Meanwhile, FkNNC (Parande 2019) should compute up to 250 thousand

distances between a candidate syllabification and all impossible indexed-sorted patterns in the training-set, choose the *k* closest patterns in both classes (a point and not a point of syllabification), and eventually select the class with the lowest total fuzzy-distance as the decision.

Fig. 7

SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of formal words



3.2. Evaluation on the dataset of named-entities

The proposed CBSPS just uses both syllable bigrams and unigrams as well as their phonological similarities to maximize the scores of syllabifications. Hence, it can be applied to a dataset of named-entities since the phonological similarities are very common in this dataset. For example, mapping two graphemes ⟨b⟩ and ⟨d⟩ in a named-entity "**b**an.**d**ung" (the capital city in West Java) into their phonological similarities ⟨p⟩ and ⟨t⟩ produces three other named-entities: "**b**an.**t**ung" (a resort

in Sukhothai, Thailand), “**p**an.**d**ung” (a village in Special Region of Yogyakarta), and “**p**an.**t**ung” (a folk song from Bolaang Mongondow, North Sulawesi).

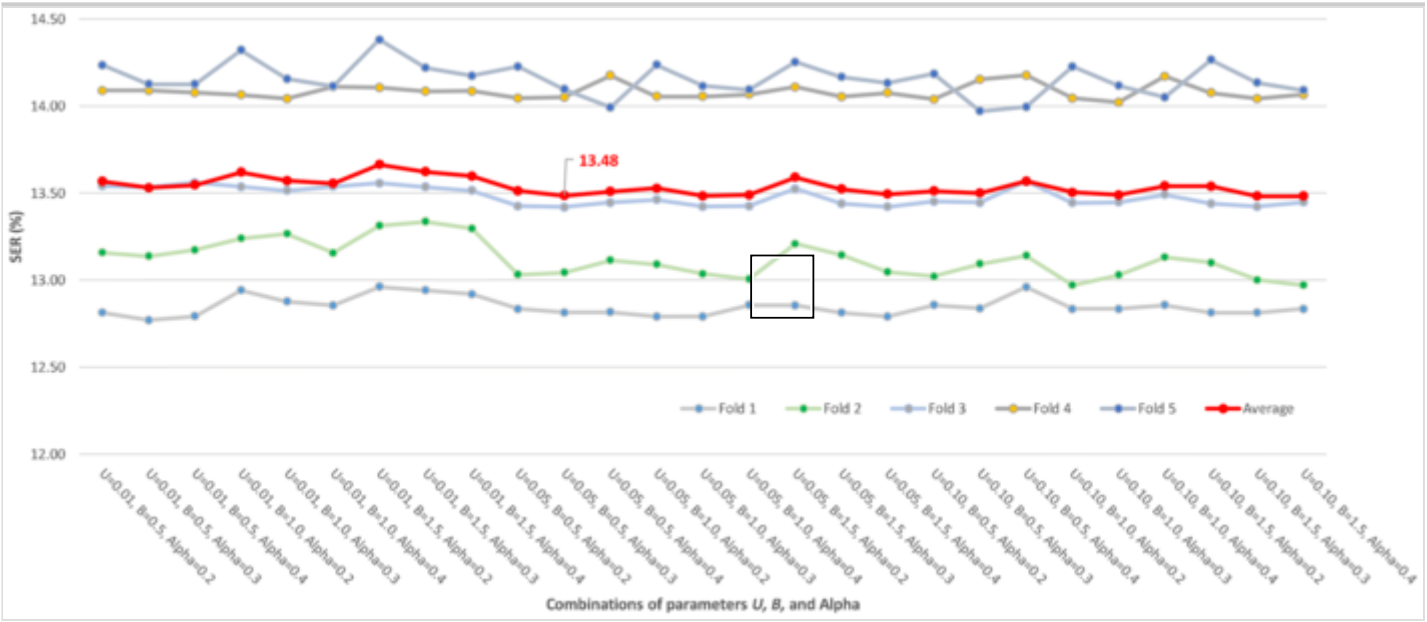
Careful observation on the dataset of 15k named-entities informs that it produces a total of 45,799 legal syllable-unigrams. Swapping procedure on those 15k named-entities produces a total of 385,850 swapped syllable-unigrams, where 90.92% of them are legal and the rest 9.08% are unseen. Hence, the swapping procedure significantly increases the number of unigrams by up to 8.43 times. Furthermore, those 15k named-entities create a total of 30,516 syllable-bigrams. Swapping them produces a total of 275,472 swapped syllable-bigrams, where 84.61% of them are legal and the rest 15.39% are unseen. It means that the swapping procedure impressively increases the number of bigrams by up to 9.03 times. These facts imply that the proposed CBSPS will be better than BS in syllabifying the named-entities. Therefore, CBSPS is evaluated here using this dataset of named-entities in a 5-fold cross-validation scheme. First, the three parameters *U*, *B*, and α are jointly optimized. The SER produced by the optimum values of those parameters is then compared to three other models: BS, BFO, and FkNNC.

3.2.1. Jointly parameters optimization

The three parameters are jointly optimized using the potential values resulted from the previous experiment on the dataset of formal words, where $U = \{0.05, 0.10, 0.15\},\ B = \{0.5, 1.0, 1.5\},$ and $\alpha = \{0.2, 0.3, 0.4\}.$ The results in Fig. 8 concludes that the optimum combination of the three parameters is $U = 0.05,\ B = 0.5,\ \text{and}\ \alpha = 0.3$ that gives the lowest average SER of 13.48%.

Fig. 8

SERs produced by CBSPS using jointly parameters optimization for the dataset of named-entities

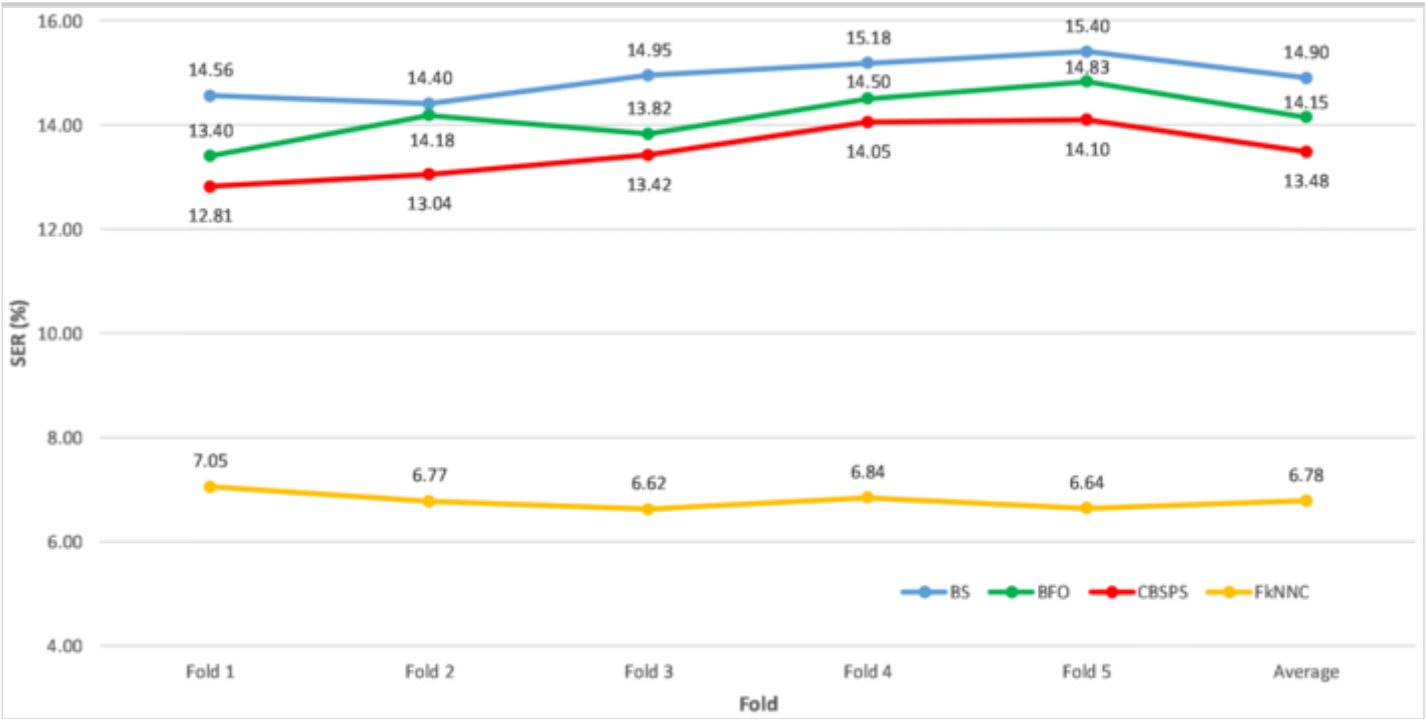


3.2.2. Comparison to other models

The best performance of CBSPS is then compared to three other syllabification models BS, BFO, and FkNNC using the same dataset of 15k named-entities described in Parande (2019). All models are compared in their best performances to get fairness. The results in Fig. 9 show that CBSPS produces a lower average SER (13.48%) than BS (14.90%), which means that it relatively decreases the mean SER by 9.53%. It is also better than BFO (14.15%) by relatively reducing the average SER by 4.83%. However, it is much worse than FkNNC, which reaches the lowest mean SER of 6.78%. This result is caused by much vowel ambiguity in the named-entities. For instances, the person names “A.dy”, “A.di”, and “A.dhie” are pronounced as /ɑ.di/ and the person names “Bu.dy”, “Bu.di”, and “Bu.dhie” are pronounced as /bu.di/. Since CBSPS always considers the grapheme ⟨y⟩ as a consonant, not a semi-vowel nor a vowel, it fails to syllabify “Budy” into “Bu.dy”.

Fig. 9

SERs produced by BS, BFO, CBSPS, and FkNNC models for the dataset of named-entities



3.3. Evaluation on the mixed dataset of formal words and named-entities

The proposed CBSPS is finally evaluated using the mixed dataset of 50k formal words and 15k named-entities to see its generalization. This dataset of 65k entries is equally divided into five folds; each contains 13k entries. First, the three parameters of CBSPS are jointly optimized. Its best performance is then compared to both BS and BFO models. Unfortunately, it cannot be compared to the FkNNC since there is no experimental result for this mixed dataset provided in Parande (2019).

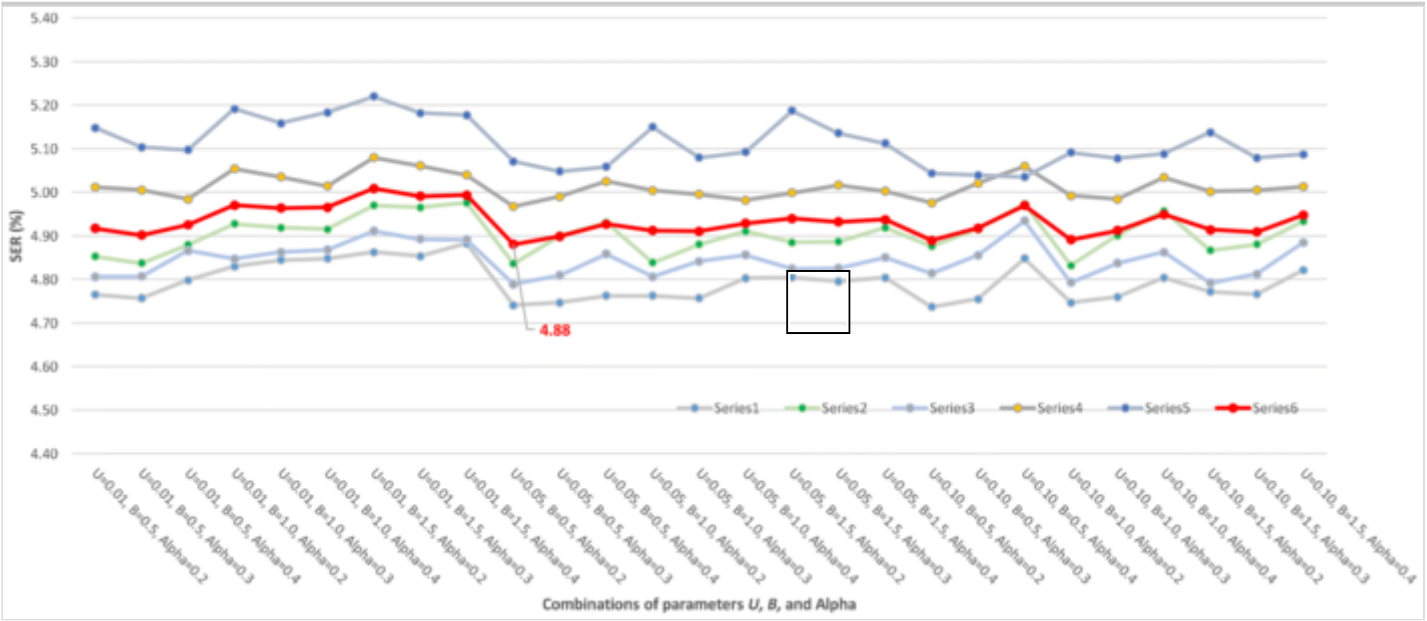
3.3.1. Jointly parameters optimization

The three parameters are jointly optimized using the potential values resulted from the previous experiments, i.e. $U = \{0.05, 0.10, 0.15\}$, $B = \{0.5, 1.0, 1.5\}$, and $\alpha = \{0.2, 0.3, 0.4\}$. The results in Fig. 10 show that the best combination

of the three parameters is $U = 0.05,\ B = 0.5,$ and $\alpha = 0.2,$ which gives the lowest average SER of 4.88%. Further investigation indicates that the SER produced by the named-entities is slightly lower than the previous model (trained using the named-entities only), but the SER from the formal words does not decrease at all. It means that CBSPS is capable of generalizing bigrams from the formal words into the named-entities, but not vice versa.

Fig. 10

SERs produced by CBSPS using jointly parameters optimization for the mixed dataset of formal words and named-entities

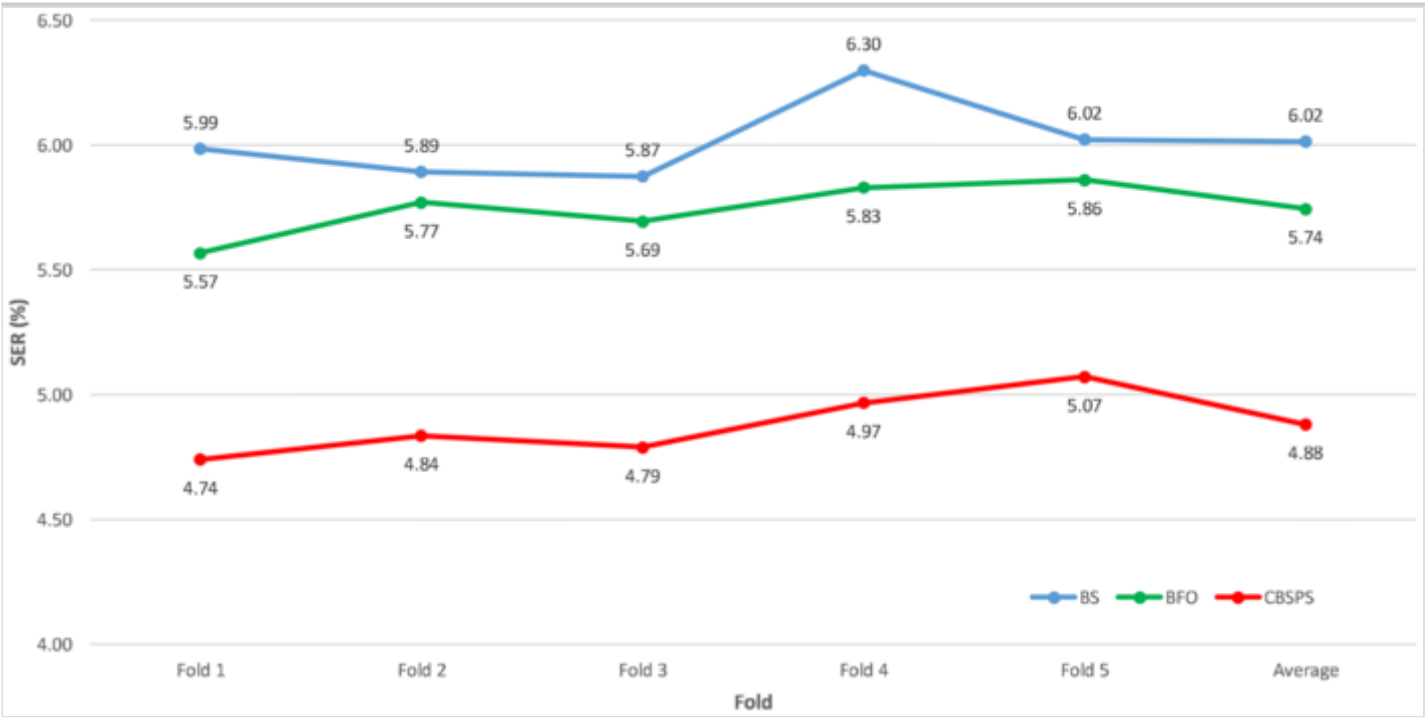


3.3.2. Comparison to other models

The best performance of CBSPS is then compared to both BS and BFO using the mixed dataset of 65k words. The results in Fig. 11 show that CBSPS produces smaller average SER (4.88%) than both BS (6.02%) and BFO (5.74%), which means that it gives relative reductions of 18.94% and 14.98%, respectively. The results also show that the performance of CBSPS is stable for all five folds.

Fig. 11

SERs produced by BS, BFO, and CBSPS models for the mixed dataset of formal words and named-entities



3.4. Evaluation on the training-set sizes

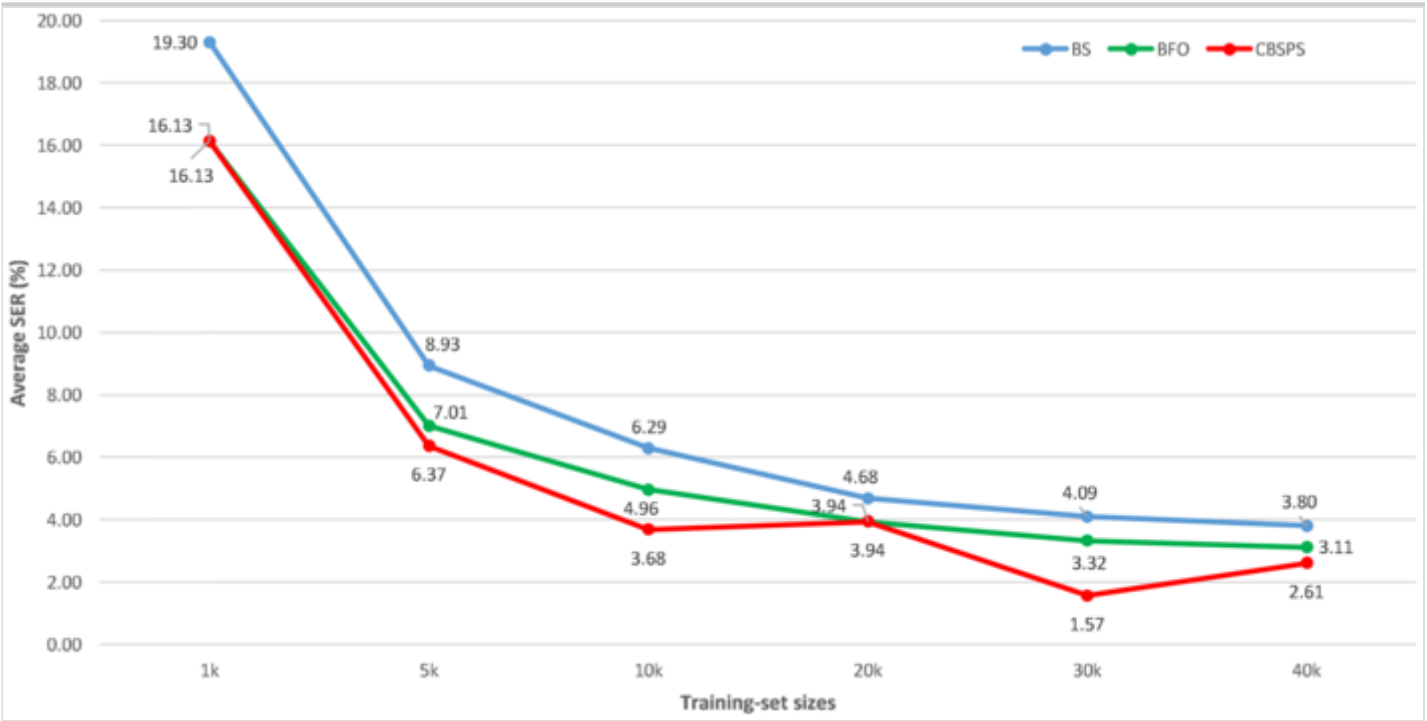
First, some different sized training-sets are developed by randomly selecting words from the five folds in the dataset of formal words. Each fold is defined as the fixed testing-set. Next, six training-sets of 1k, 5k, 10k, 20k, 30k, and 40k are then randomly generated five times from the remaining four folds (not in the testing-sets) so that each size of training-set contains five subsets. The comparisons of BS, BFO, and CBSPS are performed repeatedly five times (once for each subset), and the average SERs are then calculated.

The experimental results in Fig. 12 shows that CBSPS, for most training-set sizes, gives lower average SERs than both BS and BFO. For the training-set size of 1k, both CBSPS and BFO give the same average SER of 16.13%, which is smaller than BS (19.30%). For the training-set of 5k, CBSPS produces the lowest mean SER of 6.37% among BFO (7.01%) and BS (8.93%). The exciting results come from the training-set of 10k, where CBSPS yields impressively lower mean SER (3.68%) than both BFO (4.96%) and BS (6.29%). For the training-set of 20k, CBSPS and BFO yield the same mean SER of 3.94% that is smaller than BS (4.68%). The most exciting results come from the training-set of 30k, where CBSPS reaches a much lower average SER (1.57%) than both BFO (3.32%) and BS (4.09%). It means that CBSPS gives the relative reductions of the average SER by up to 52.71% and 61.61%, respectively. Finally, for the training-set of 40k, CBSPS also gives the smallest mean SER of 2.61% among BFO (3.31%) and BS (3.80%).

These results are fascinating. Increasing the size of the training-set does not always decrease the SER. Sometimes it raises the SER. It can be said that CBSPS is not stable. This fact can be easily explained here that swapped-consonant words producing many illegal OOV bigrams and unigrams potentially increase the SER. In other words, the unstable SERs produced by CBSPS are caused by the illegal OOV swapped-bigrams and swapped-unigrams generated from the training-sets. Therefore, a scheme of filtering legal bigrams and unigrams can be introduced to enhance CBSPS.

Fig. 12

Average SERs produced by BS, BFO, and CBSPS for six different sized training-sets taken from the 50k formal Indonesian words



4. Conclusion

The proposed CBSPS model is capable of significantly boosting the standard bigram-syllabification (BS) model for the dataset of 50k formal words with a relative reduction of SER up to 31.39%. It is also better than BFO by relatively reducing the mean SER by 16.08%. However, it is slightly worse than FkNNC, but it offers a much lower complexity. Nevertheless, CBSPS can give a relatively low SER, even for a tiny training-set, since it exploits a swapping graphemes-based data augmentation that significantly increases the number of bigrams and unigrams. For the dataset of 15k named-entities, CBSPS is also better than both BS and BFO with relative reductions of SER by 9.53% and 4.83%, respectively. However, it is worse than FkNNC since it is hard to solve much ambiguity of vowel in the named-entities. In the future, a scheme of filtering legal bigrams and unigrams can be introduced to improve its performance.

Publisher's Note

Acknowledgements

I want to give a high appreciation to Muhammad Agha Ariyanto, a 3-year son, for his great inspiration in speaking words by swapping one or more consonants into similar ones based on both place and manner of articulation.

References

Adsett, C. R., Marchand, Y., & Kešelj, V. (2009). Syllabification rules versus data-driven methods in a language with low syllabic complexity: the case of Italian. *Computer Speech and Language*, *23*, 444–463. https://doi.org/10.1016/j.csl.2009.02.004.

Alwi, H., Lapoliwa, H., & Darmowidjojo, S. (2003). *Tata Bahasa Baku Bahasa Indonesia* [The standard Indonesian grammar] (3rd ed.). Jakarta: Balai Pustaka.

Aripin, Haryanto, H., & Sumpeno, S. (2018). 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*, *13*(8), 19–37. https://doi.org/10.3991/ijet.v13i08.8084.

Balc, D., Beleiu, A., Potolea, R., & Lemnaru, C. (2015). A learning-based approach for Romanian syllabification and stress assignment. In *Proceedings—2015 IEEE 11th international conference on intelligent computer communication and processing, ICCP 2015* (pp. 37–42). Institute of Electrical and Electronics Engineers, Cluj-Napoca, Romania. https://doi.org/10.1109/ICCP.2015.7312603.

Bartlett, S., Kondrak, G., & Cherry, C. (2009). 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* (pp. 308–316). Boulder, CO. https://doi.org/10.3115/1620754.1620799.

Ben Alex, S., Babu, B. P., & Mary, L. (2019). Utterance and syllable level prosodic features for automatic emotion recognition. In *2018 IEEE recent advances in intelligent computational systems, RAICS 2018* (pp. 31–35). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/RAICS.2018.8635059. https://ieeexplore.ieee.org/document/8635059

Bernard, A. (2015). An onset is an onset: Evidence from abstraction of newly-learned phonotactic constraints. *Journal of Memory and Language*, *78*, 18–32. https://doi.org/10.1016/j.jml.2014.09.001.

Brants, T., Popat, A. C., & Och, F. J. (2007). 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, pp. 858–867). https://www.aclweb.org/anthology/D07-1090

Daelemans, W., Bosch, A. V. D., & Weijters, T. (1997). IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, *11*(1–5), 407–423. https://doi.org/10.1023/A:1006506017891.

Faldessai, N., Pawar, J., & Naik, G. (2017). Syllabification: An effective approach for a TTS system for Konkani. In *2016 International conference on electrical, electronics, communication, computer and optimization techniques, ICEECCOT 2016* (pp. 161–167). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICEECCOT.2016.7955207.

Fallows, D. (1981). Experimental evidence for English syllabification and syllable structure. *Journal of Linguistics*, *17*(2), 309–317. https://doi.org/10.1017/S0022226700007027.

Feng, S., & Lee, T. (2019). Exploiting cross-lingual speaker and phonetic diversity for unsupervised subword modeling. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *27*(12), 2000–2011. https://doi.org/10.1109/TASLP.2019.2937953.

Foster, C. C. (1992). A comparison of vowel identification methods. *Cryptologia*, *16*(3), 282–286. https://doi.org/10.1080/0161-119291866955.

Geeta, S., & Muralidhara, B. L. (2018). 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, pp. 1205–1208). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/WiSPNET.2017.8299954. https://ieeexplore.ieee.org/document/8299954

Hlaing, T. H., & Mikami, Y. (2014). Automatic syllable segmentation of Myanmar texts using finite state transducer. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, *6*(2), 2–9. https://doi.org/10.4038/icter.v6i2.7150.

Johnson, D. O., & Kang, O. (2017). Comparison of algorithms to divide noisy phone sequences into syllables for automatic unconstrained English speaking proficiency scoring. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-017-9594-y.

Kamper, H., Jansen, A., & Goldwater, S. (2017). A segmental framework for fully-unsupervised large-vocabulary speech recognition. *Computer Speech & Language*, *46*, 154–174. https://doi.org/10.1016/j.csl.2017.04.008.

Krantz, J., Dulin, M., De Palma, P., & VanDam, M. (2018). Syllabification by phone categorization. In *Proceedings of the genetic and evolutionary computation conference companion, GECCO '18* (pp. 47–48). ACM, New York. https://doi.org/10.1145/3205651.3208781.

Krisnawati, L. D., & Mahastama, A. W. (2019). A Javanese syllabifier based on its orthographic system. In M. Dong & F. Z. Ruskanda (Eds.), *International conference on Asian Language processing* (pp. 244–249). Piscataway: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IALP.2018.8629173.

Kulju, P., & Mäkinen, M. (2019). Phonological strategies and peer scaffolding in digital literacy game-playing sessions in a Finnish pre-primary class. *Journal of Early Childhood Literacy*. https://doi.org/10.1177/1468798419838576.

Leemann, A., Kolly, M. J., Nolan, F., & Li, Y. (2018). The role of segments and prosody in the identification of a speaker's dialect. *Journal of Phonetics*, *68*, 69–84. https://doi.org/10.1016/j.wocn.2018.02.001.

Magdum, D., & Suman, M. (2019). System for identifying and correcting invalid words in the devanagari script for text to speech engine. *International Journal of Innovative Technology and Exploring Engineering*, *8*(6 Special Issue 4), 1001–1006. https://doi.org/10.35940/ijitee.F1206.0486S419.

Mayer, T. (2010). 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* (pp. 63–71).

Müller, K. (2006). 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* (pp. 11–20). https://doi.org/10.3115/1622165.1622167.

Mulyanto, E., Yuniarno, E. M., & Purnomo, M. H. (2019). Adding an emotions filter to Javanese text-to-speech system. In *2018 International conference on computer engineering, network and intelligent multimedia, CENIM 2018—Proceeding* (pp. 142–146). Piscataway: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/CENIM.2018.8711229.

Nayak, S., Bhati, S., & Rama Murty, K. S. (2019). Zero resource speaking rate estimation from change point detection of syllable-like units. In *IEEE International conference on acoustics, speech and signal processing—proceedings (ICASSP)* (Vol. 2019, pp. 6590–6594). Piscataway: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICASSP.2019.8683462. https://ieeexplore.ieee.org/document/8683462

Ngo, G. H., Nguyen, M., & Chen, N. F. (2019). Phonology-augmented statistical framework for machine transliteration using limited linguistic resources. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *27*(1), 199–211. https://doi.org/10.1109/TASLP.2018.2875269.

Oncevay-Marcos, A. (2017). Spell-checking based on syllabification and character-level graphs for a peruvian agglutinative language. In *The First workshop on subword and character level models in NLP* (pp. 109–116).

Pakoci, E., Popović, B., & Pekar, D. (2019). Using morphological data in language modeling for serbian large vocabulary speech recognition. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2019/5072918.

Parande, E. A. (2019). Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. *International Journal of Speech Technology*, *22*(1), 13–20. https://doi.org/10.1007/s10772-018-09569-3.

Ramli, I., Jamil, N., Seman, N., & Ardi, N. (2015). An improved syllabification for a better malay language text-to-speech synthesis (TTS). *Procedia—Computer Science*, *76*(Iris), 417–424. https://doi.org/10.1016/j.procs.2015.12.280.

Räsänen, O., Doyle, G., & Frank, M. C. (2018). Pre-linguistic segmentation of speech into syllable-like units. *Cognition*, *171*, 130–150. https://doi.org/10.1016/j.cognition.2017.11.003.

Rogova, K., Demuynck, K., & Compernolle, D. V. (2013). Automatic syllabification using segmental conditional random fields. *Computational Linguistics in the Netherlands Journal*, *3*, 34–48.

Rugchatjaroen, A., Saychum, S., Kongyoung, S., Chootrakool, P., Kasuriya, S., & Wutiwiwatchai, C. (2019). Efficient two-stage processing for joint sequence model-based Thai grapheme-to-phoneme conversion. *Speech Communication*, *106*, 105–111. https://doi.org/10.1016/j.specom.2018.12.003.

Schmid, H., Möbius, B., & Weidenkaff, J. (2007). Tagging syllable boundaries with joint n-gram models. In *INTERSPEECH* (Vol. 1, pp. 49–52). https://www.scopus.com/inward/record.uri?eid=2-s2.0-

56149127120&partnerID=40&md5=d6c048349e00f9fa7f7afec0dc34ea84.

Segundo, E. S., & Yang, J. (2019). Formant dynamics of Spanish vocalic sequences in related speakers : A forensic-voice-comparison investigation. *Journal of Phonetics*, *75*, 1–26. https://doi.org/10.1016/j.wocn.2019.04.001.

Singh, L. G., Laitonjam, L., & Singh, S. R. (2016). Automatic Syllabification for Manipuri language. In *the 26th International conference on computational linguistics* (pp. 349–357). https://www.aclweb.org/anthology/papers/C/C16/C16-1034/

Sun, L., Fu, S., & Wang, F. (2019). Decision tree SVM model with Fisher feature selection for speech emotion recognition. *Eurasip Journal on Audio, Speech, and Music Processing*, *2019*(1), 2. https://doi.org/10.1186/s13636-018-0145-5.

Suyanto, S. (2019a). Flipping onsets to enhance syllabification. *International Journal of Speech Technology*, *22*(4), 1031–1038. https://doi.org/10.1007/s10772-019-09649-y.

Suyanto, S. (2019b). Incorporating syllabification points into a model of grapheme-to-phoneme conversion. *International Journal of Speech Technology*, *22*(2), 459–470. https://doi.org/10.1007/s10772-019-09619-4.

Suyanto, S., & Harjoko, A. (2014). Nearest neighbour-based Indonesian G2P conversion. *Telkomnika (Telecommunication, Computing, Electronics, and Control)*, *12*(2), 389–396. https://doi.org/10.12928/telkomnika.v12i2.57.

Suyanto, S., Hartati, S., Harjoko, A., & Compernolle, D. V. (2016). Indonesian syllabification using a pseudo nearest neighbour rule and phonotactic knowledge. *Speech Communication*, *85*, 109–118. https://doi.org/10.1016/j.specom.2016.10.009.

Van Esch, D., Chua, M., & Rao, K. (2016). Predicting pronunciations with syllabification and stress with recurrent neural networks. In N. Morgan & P. Georgiou (Eds.), *Proceedings of the annual conference of the international speech communication association, INTERSPEECH* (Vol. 08, pp. 2841–2845). Baixas: International Speech and Communication Association. https://doi.org/10.21437/Interspeech.2016-1419. https://www.isca-speech.org/archive/Interspeech_2016/pdfs/1419.PDF.