Evidences of correspondences

Flipping Onsets to Enhances Syllabification

1. **First submission (10 April 2019)**
2. **LoA with Minor Revision (25 September 2019)**
3. **Respond to Reviewers, Final submission (25 September 2019)**
4. **LoA with Fully Accepted (25 September 2019)**

Telkom University

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

IJST - Submission Confirmation

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Wed, Apr 10, 2019 at 4:39 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, Flipping Onsets to Enhances 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

Flipping Onsets to Enhances Syllabification

--Manuscript Draft--

Manuscript Number:	
Full Title:	Flipping Onsets to Enhances Syllabification
Article Type:	Manuscript
Keywords:	bigram; consecutive syllables; flipping onsets; orthographic syllabification
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, Dr.
First Author Secondary Information:	
Order of Authors:	Suyanto Suyanto, Dr.
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	Two-year-old children who start learning to speak generally spell a polysyllabic word by flipping onsets of consecutive syllables. Sometimes they speak unclearly, hard to understand since the flipped onsets produce another word that has a much different meaning. For instance, two onsets in an English word "me.lon" (large round fruit of a plant of the gourd family) are flipped to produce another word "le.mon" (an acid fruit). In Bahasa Indonesia, such cases are quite common. For examples, two onsets in word "ba.tu" (stone) are swapped to be "ta.bu" (taboo), two onsets in "be.sar" (big) are flipped to be "se.bar" (spread), two onsets in "ru.mah" (house) are swapped to be "mu.rah" (cheap), etc. A preliminary study on 50k Indonesian formal words shows that the ratio between frequencies of the flipped-onset-bigrams and the 50 most frequent original syllable-bigrams is quite high, up to 13.09%. This research investigates the adoption of such phenomenon to enhances a bigram orthographic syllabification model that is commonly poor for out-of-vocabulary words. A 5-fold cross-validation on 50k Indonesian formal words proves that the flipping onsets enhances the bigram orthographic syllabification, where the syllable error rate (SER) is relatively reduced by 18.02%. The method is also capable of producing quite low SER for a tiny trainset of 1k words to generalize 10k unseen words. Besides, it can be simply generalized to be applied to other languages as well as name-entities using a few specific knowledge related to the sets of vowels, diphthongs, and consonants.

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

Flipping Onsets to Enhances Syllabification

Suyanto Suyanto

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)

Flipping Onsets to Enhances Syllabification

Abstract Two-year-old children who start learning to speak generally spell a polysyllabic word by flipping onsets of consecutive syllables. Sometimes they speak unclearly, hard to understand since the flipped onsets produce another word that has a much different meaning. For instance, two onsets in an English word "**m**e.**l**on" (large round fruit of a plant of the gourd family) are flipped to produce another word "**l**e.**m**on" (an acid fruit). In Bahasa Indonesia, such cases are quite common. For examples, two onsets in word "***b**a.**t**u*" (stone) are swapped to be "***t**a.**b**u*" (taboo), two onsets in "***b**e.**s**ar*" (big) are flipped to be "***s**e.**b**ar*" (spread), two onsets in "***r**u.**m**ah*" (house) are swapped to be "***m**u.**r**ah*" (cheap), etc. A preliminary study on 50k Indonesian formal words shows that the ratio between frequencies of the flipped-onset-bigrams and the 50 most frequent original syllable-bigrams is quite high, up to 13.09%. This research investigates the adoption of such phenomenon to enhances a bigram orthographic syllabification model that is commonly poor for out-of-vocabulary words. A 5-fold cross-validation on 50k Indonesian formal words proves that the flipping onsets enhances the bigram orthographic syllabification, where the syllable error rate (SER) is relatively reduced by 18.02%. The method is also capable of producing quite low SER for a tiny trainset of 1k words to generalize 10k unseen words. Besides, it can be simply generalized to be applied to other languages as well as name-entities using a few specific knowledge related to the sets of vowels, diphthongs, and consonants.

Keywords bigram · consecutive syllables · flipping onsets · orthographic syllabification

1 Introduction

A syllable is very important in phonology since it is relevant to the phonological rules. It is also a basis to describe stresses in a language. Breaking a word into syllables automatically, commonly called automatic syllabification, is one of the interesting researches since it relates to many other fields of studies as well as applications. The automatic syllabification is commonly used in both syllable-based speech recognition [30], [13], [19] and speech synthesis systems [25]. It is also widely exploited in spell-checking [4], [23], information retrieval [14], and statistical machine translation [18].

In general, an automatic syllabification can be applied to either grapheme or phoneme sequences. For many formal words, a phonemic syllabification commonly gives lower SER than an orthographic (also known as graphemic) syllabification, as described in [28] and [24]. However, the phonemic syllabification needs a perfect phoneme sequence that generally cannot be provided by any G2P model, but by a linguist. Moreover, it looks to be getting worse (produce a higher SER) when it is applied for a name-entity since this problem has a high dynamic and ambiguity. These facts make the graphemic syllabification is more widely used in practice than the phonemic one since it does not need any module of grapheme-to-phoneme conversion (G2P).

The researchers mostly prefer to develop automatic syllabifications using a statistical-based approach rather than the rule-based one [2], [26]. It can be implemented using two different techniques: supervised and unsupervised learning. The supervised learning is more popular and commonly implemented using either global or local classification method.

Some global classifications to develop a syllabification model are neural networks [8], [17], [29], decision tree [9], treebank [21], support vector machine [5], hidden Markov model [16], finite-state transducers [11], [15], context-free grammars [22], syllabification by analogy [1], unsupervised method [20], joint n-gram models [27], and segmental conditional random fields (SCRF) [26]. Meanwhile, the syllabifications based on local classifications are generally developed using nearest neighbour-based methods as described in [24] and [28]. The local classifications are quite accurate but time consuming.

Some methods are designed to be low-cost as well as language-independent. One of them is the bigram syllabification model. Although it is commonly poor for low-resource languages with so many out-of-vocabulary (OOV) words, some features can be added to improve its performance. One of the improved bigram syllabifications is SCRF [26]. The main advantage of SCRF is high generalization with limited trainset. But, this method is quite complex. It incorporates eight features produced by three general principles of syllabification (sonority, legality, and maximum onset) to calculate the syllable bigram probability. Besides, it is applied to a phoneme string that is relatively easier to solve than a grapheme string.

In this research, a new simpler scheme of flipping onsets contained in some consecutive syllables is incorporated into a bigram-based method. Its effect is investigated on an orthographic syllabification for Bahasa Indonesia as a low resource language. First, a standard bigram is implemented as a baseline model. A bigram with flipping onsets is then developed and then investigated to see if it is capable of reducing the SER significantly.

The proposed scheme is inspired by an interesting phenomenon of two-year-old children who start learning to speak. They generally spell a polysyllabic word by flipping onsets of two first consecutive syllables. Sometimes they speak unclearly, hard to understand, since flipping those onsets may produce another word that have much different meaning. For examples, an English word ”**b**a.**s**in” (a bowl for washing) is spelled as ”**s**a.**b**in” (a vaccine against poliomyelitis), ”**c**a.**p**o” (the head of a crime syndicate) is spoken as ”**p**a.**c**o” (alpaca), and ”**m**e.**l**on” (large round fruit of a plant of the gourd family) is said as ”**l**e.**m**on” (an acid fruit). In Bahasa Indonesia, such cases are quite common. For instance, onsets of two syllables in word ”**b**a.**w**ah” (under) are flipped to be ”**w**a.**b**ah” (epidemic), onsets in ”**b**e.**k**al” (stock) are swapped to be ”**k**e.**b**al” (immune), onsets in ”**b**e.**s**ar” (big) are flipped to be ”**s**e.**b**ar” (spread), onsets in ”**k**e.**s**al” (upset) are swapped to be ”**s**e.**k**al” (plump), etc.

A preliminary observation on 50k Indonesian formal words shows an interesting fact. Table 1 illustrates the top 50 most frequent syllable-bigrams from the 50k words, where their total frequency is 7,394. The interesting fact is the flipped-onset-bigrams corresponding to those syllable-bigrams have a relatively high frequency of 968. This produces a high ratio between the frequency of the flipped-onset-bigram and the original syllable-bigram, i.e. 13.09%. This ratio can be probably higher when the most frequent syllable-bigrams taken into account are more than 50. Hence, the proposed scheme of flipping onsets is expected to relatively reduce the SER by at least 13.09%.

However, two-year-old children sometimes also flip onsets in the second and the third syllables. For instance, the onsets in an Indonesian formal word ”ke.**p**a.**l**a” (head) are flipped to be ”ke.**l**a.**p**a” (coconut). Hence, flipping onsets can be generalized into more than two consecutive syllables. This phenomenon is also investigated by doing some experiments with varying number of consecutive syllables from 2 to 7. This is motivated by a fact that Bahasa Indonesia has some long words containing up to 7 syllables or more, where on average it has 3.20 syllables in a word [28].

2 Flipping Onsets to Enhances a Bigram Syllabification

A word is composed of one or more syllables. A syllable contains a non-obligatory onset, followed by an obligatory nucleus, and ended by a non-obligatory coda [12]. The nucleus can be a single vowel, a diphthong or, in some languages such as English, a sonorant consonant. In Bahasa Indonesia, the nucleus is always either a single vowel or a diphthong while onset and coda are always consonants [3], [7]. For example, an Indonesian word ”*besar*” (big) is composed of two syllables, ⟨be⟩ and ⟨sar⟩. The first syllable contains a nucleus ⟨e⟩ that is preceded by an onset ⟨b⟩ but not followed by any coda. Meanwhile, the second syllable consists of a nucleus ⟨a⟩ with an onset ⟨s⟩ and a coda ⟨r⟩.

Hence, the number of syllables contained in an Indonesia word should equal to the number of vowels and/or diphthongs [3]. This simplifies the problem

Table 1 Top 50 frequent syllable-bigrams generated from 50k Indonesian formal words collected from KBBI

Rank	Syllable-bigram	Frequency	Flipped-onsets-bigram	Frequency
1	mem.per	382	pem.mer	0
2	me.nga	367	nge.ma	2
3	me.nge	330	nge.me	0
4	me.nye	295	nye.me	1
5	sa.si	251	sa.si	251
6	lo.gi	219	go.li	1
7	me.ma	181	me.ma	181
8	si.a	181	i.sa	4
9	pe.nga	174	nge.pa	3
10	me.ngu	172	nge.mu	8
11	ber.ke	162	ker.be	0
12	me.na	161	ne.ma	15
13	me.ne	161	ne.me	0
14	si.o	152	i.so	63
15	ka.si	150	si.ka	9
16	me.ngi	148	nge.mi	0
17	ta.si	145	sa.ti	5
18	me.ra	140	re.ma	12
19	me.ter	140	te.mer	0
20	ra.si	139	sa.ri	27
21	me.nya	138	nye.ma	7
22	si.kan	136	ki.san	9
23	me.la	132	le.ma	12
24	meng.ge	132	geng.me	0
25	ber.se	128	ser.be	0
26	pe.nye	128	nye.pe	2
27	li.sa	126	si.la	41
28	mem.be	124	bem.me	0
29	ber.a	120	er.ba	0
30	mem.ba	120	bem.ma	0
31	pe.ra	116	re.pa	6
32	me.le	115	le.me	3
33	na.si	115	sa.ni	13
34	me.me	114	me.me	114
35	pe.nge	113	nge.pe	0
36	men.de	112	den.me	0
37	me.re	110	re.me	6
38	la.si	106	sa.li	0
39	a.si	103	sa.i	18
40	a.li	101	la.i	9
41	me.nu	101	ne.mu	6
42	di.a	100	i.da	10
43	pe.la	99	le.pa	10
44	me.nyu	97	nye.mu	1
45	te.ra	97	re.ta	10
46	me.mu	96	me.mu	96
47	pa.ra	94	ra.pa	11
48	me.ngo	92	nge.mo	1
49	gra.fi	90	fa.gri	0
50	li.tas	89	ti.las	1
Total		7,394		968

of finding the correct syllabification since it limits the possible syllabification points or boundaries in between vowels (and/or diphthongs). For instance, the word "*besar*" produces only two candidate syllabifications, i.e. ⟨be.sar⟩ and ⟨bes.ar⟩, while two others ⟨b.esar⟩ and ⟨besa.r⟩ are illegal syllabifications.

Vowels and consonants. In a text, some vowels and diphthongs can be automatically detected using some methods, such as Sukhotin's algorithm and its combinations those are language-independent unsupervised learning [10]. Evaluations on 39 languages contain more than 100,000 symbols each show that the average accuracy of Sukhotin's algorithm is 95.66% [20]. Since Bahasa Indonesia has simple typological knowledge, this research does not exploit Sukhotin's algorithm. As described in [3], Bahasa Indonesia has five graphemes those are single vowels {⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩, ⟨u⟩}, four sequences of graphemes those form diphthongs {⟨ai⟩, ⟨au⟩, ⟨ei⟩, ⟨oi⟩}, and twenty one graphemes those are consonants {⟨b⟩, ⟨c⟩, ⟨d⟩, ⟨f⟩, ⟨g⟩, ⟨h⟩, ⟨j⟩, ⟨k⟩, ⟨l⟩, ⟨m⟩, ⟨n⟩, ⟨p⟩, ⟨q⟩, ⟨r⟩, ⟨s⟩, ⟨t⟩, ⟨v⟩, ⟨w⟩, ⟨x⟩, ⟨y⟩, ⟨z⟩}.

Monophthong and diphthong. One of challenges in automatic syllabification is how to distinguish monophthong and diphthong. Two consecutive graphemes ⟨ai⟩ in the word "*cintai*" (to love) are two monophthongs ⟨a⟩ and ⟨i⟩ so that the word is syllabified as ⟨cin.ta.i⟩ while ⟨ai⟩ in the word "*intai*" (to spy) is a diphthong so that it is segmented as ⟨in.tai⟩. In bigram syllabification, the problem regarding any word that probably contains a diphthong can be solved by maximizing the bigram probabilities of all possible syllabifications. The word "*cintai*" produces six candidates, i.e. ⟨ci.nta.i⟩, ⟨cin.ta.i⟩, and ⟨cint.a.i⟩ (where ⟨ai⟩ are assumed as two monophthongs) as well as ⟨ci.ntai⟩, ⟨cin.tai⟩, and ⟨cint.ai⟩ (where ⟨ai⟩ is hypothesized as a diphthong).

Bigram syllabification model. The bigram syllabification can be seen as finding the most likely syllable sequence that represents a word. The probability of a bigram syllabification of L tokens $P(w_1, w_2, ..., w_L)$ is calculated using the probability chain. In practice, there are so many smoothing methods to estimate the probability to solve the problem of sparsity affected by the OOV words. One of them is the *Stupid Backoff* that is introduced in [6]. In the *Stupid Backoff* smoothing, the probability is estimated using a score S that may have a value bigger than 1

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

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ denotes the frequencies of occurrences of both syllable-bigrams and syllable-unigrams in the trainset respectively, α is the backoff factor heuristically set to 0.4, and N is the trainset size [6]. In practical implementation, the score for an unseen syllable-unigram, a case where $f(w_i) = 0$, is approximated as the score for syllable-unigram seen once that produces $S(w_i|w_{i-1}) = \alpha\frac{1}{(N+1)}$. The final score is computed using a logarithmic formula $\frac{1}{-\log(S)}$ to avoid producing an underflow value.

Proposed model. In general, the bigram syllabification model is poor for a low trainset with so many OOV syllable-bigrams [26]. In case of Bahasa Indonesia, a relatively low trainset of 50k words contains a vocabulary $V = 2,267$ distinct syllable-unigrams and a total token $N = 33,147$ unique syllable-bigrams. This is only 0.64% ($33,147$ out of $2,267^2 = 5,139,289$ possible syllable-bigrams). It means there are 99.36% unseen syllable-bigrams. In other words, it has high OOV syllable-bigrams. To solve the problem, a simple modification of the bigram syllabification model is proposed by incorporating a procedure of flipping onsets of two or more consecutive syllables contained in a word. Pseudocode of the proposed method, called bigram with flipping onsets (BFO), can be briefly described in three steps below:

1. From a given grapheme sequence, detect all positions of vowels as well as possible diphthongs and do the procedure of bigram syllabification with *Stupid Backoff* smoothing, i.e. generate all C candidate syllabifications and compute their scores S_i, where $i = 1, 2, ..., C$, using Equation 1 and Equation 2;
2. For each candidate syllabification do flipping onsets in B consecutive syllables to generate new $(B-1)$ candidates, where B is an integer bigger than 2, and compute the average score $\bar{S}_i$ by taking into account the previous corresponding score in step 1; and
3. Finally, select the ith candidate with the highest average score $\bar{S}_i$ as the best syllabification.

The comparison of Bigram and BFO can be simply explained using five illustrations in Table 2 to Table 6, where both model are developed using a trainset of 40k words. In Table 2, the bigram syllabification segments the word containing possible diphthong "*hasai*" (fragile) into a wrong syllabification ⟨ha.sa.i⟩. In contrast, Table 3 shows that BFO correctly gives a true syllabification ⟨ha.sai⟩. A more complex example is syllabification of a word "*berani*" (brave) that has three syllables. In Table 4, the bigram syllabification segments the word "*berani*" into a wrong syllabification ⟨ber.a.ni⟩. In contrast, Table 5 and Table 6 shows that BFO with two and three flipped onsets successfully produces a true syllabification ⟨be.ra.ni⟩.

3 Result and Discussion

In this research, the dataset of 50k Indonesian formal words with their syllabification points is used. It is the same dataset as described in [24]. Two experiments are conducted to investigate the effects of both trainset sizes and

Table 2 Bigram syllabification breaks the word "*hasai*" into a wrong syllabification ⟨ha.sa.i⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
ha.sa.i	0.0677	has.a.i	0.0468	ha.sai	0.0660	has.ai	0.0533
$\bar{S}_1$	**0.0677**	$\bar{S}_2$	0.0468	$\bar{S}_3$	0.0660	$\bar{S}_4$	0.0533

Table 3 BFO correctly segments a word "*hasai*" into ⟨ha.sai⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
ha.sa.i	0.0677	has.a.i	0.0468	ha.sai	0.0660	has.ai	0.0533
sa.ha.i	0.0611	as.ha.i	0.0485	sa.hai	0.0639	as.hai	0.0571
$\bar{S}_1$	0.1288	$\bar{S}_2$	0.0953	$\bar{S}_3$	**0.1299**	$\bar{S}_4$	0.1104

Table 4 Bigram syllabification breaks the word "*berani*" into a wrong syllabification ⟨ber.a.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0776	be.ran.i	0.0634	ber.a.ni	0.0836	ber.an.i	0.0661
$\bar{S}_1$	0.0776	$\bar{S}_2$	0.0634	$\bar{S}_3$	**0.0836**	$\bar{S}_4$	0.0661

Table 5 BFO with two flipped onsets correctly syllabifies the word "*berani*" into ⟨be.ra.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0836	be.ran.i	0.0634	ber.a.ni	0.0776	ber.an.i	0.0661
re.**b**a.ni	0.0588	**r**e.**b**an.i	0.0549	er.**b**a.ni	0.0566	er.**b**an.i	0.0435
$\bar{S}_1$	**0.0712**	$\bar{S}_2$	0.0592	$\bar{S}_3$	0.0671	$\bar{S}_4$	0.0548

Table 6 BFO with three flipped onsets correctly segments a word "*berani*" into ⟨be.ra.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0776	be.ran.i	0.0634	ber.a.ni	0.0836	ber.an.i	0.0661
re.**b**a.ni	0.0588	**r**e.**b**an.i	0.0549	er.**b**a.ni	0.0566	er.**b**an.i	0.0435
be.**n**a.**r**i	0.0765	be.an.**r**i	0.0575	ber.**n**a.i	0.0658	ber.an.i	0.0661
$\bar{S}_1$	**0.0710**	$\bar{S}_2$	0.0586	$\bar{S}_3$	0.0687	$\bar{S}_4$	0.0586

number of flipped onsets. The performance is measured using a syllable error rate (the percentage of syllable error).

Trainset sizes. First, the 50k words in the dataset are randomly selected to produce six trainset sizes: 1k, 5k, 10k, 20k, 30, and 40k. The random selection is performed five times for each trainset size. The testset size is fixed to 10k. Next, the experiments to investigate the performance of both Bigram and BFO syllabifications are repeated five times for each trainset size to reduce the random effect. The results illustrated by Fig. 1 shows that BFO syllabification produces lower average SERs for all trainset sizes than the Bigram one. For the smallest trainset size of 1k, BFO produces lower average SER of 16.13%

than Bigram that reaches up to 19.30%. This result shows that BFO has a high generalization, where it gives a quite low SER for a much bigger testset of 10k unseen words based on a very small trainset of 1k words. For the trainset size of 10k, which is the same as the testset size, BFO produces much lower average SER of 4.96% than Bigram that reaches 6.29%. For the biggest trainset size of 40k, BFO gives the lowest average SER of 3.11% while Bigram reaches 3.80%. It means the proposed scheme relatively reduces the SER by 18.02% as expected. This result of BFO is slightly worse compared to the nearest neighbour-based syllabification in [24] that produces SER of 2.27%, but BFO is much more efficient in computation.

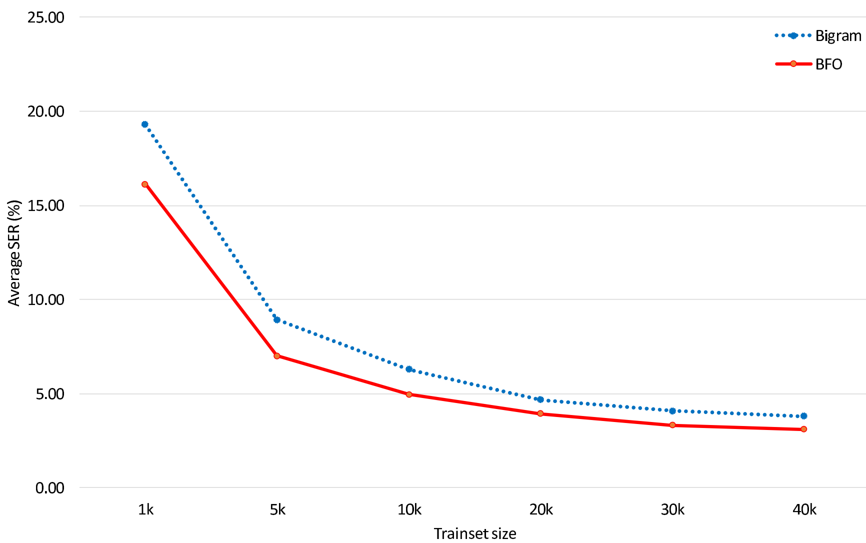

Fig. 1 Comparison of Bigram and BFO for some varying sizes of trainsets. BFO gives lower SERs for all trainset sizes, especially for the smallest one

Number of flipped onsets. In this experiment, the dataset is randomly divided into five distinct subsets to do the 5-fold cross-validation. The results in Fig. 2 shows that flipping two onsets produces the lowest average SER of 3.11%. The SER slightly increases with increasing number of flipped onsets. The detail observations show that taking into account more than two flipped onsets produces some biases in the average scores of the candidate syllabifications. This result follows a phenomenon that two-year-old children frequently flip the two first onsets, but rarely flip three or more onsets, contained in a word.

Hard-to-solve problems. The proposed method sometimes fails to syllabify some ambiguous words, which come from some roots those are similar to

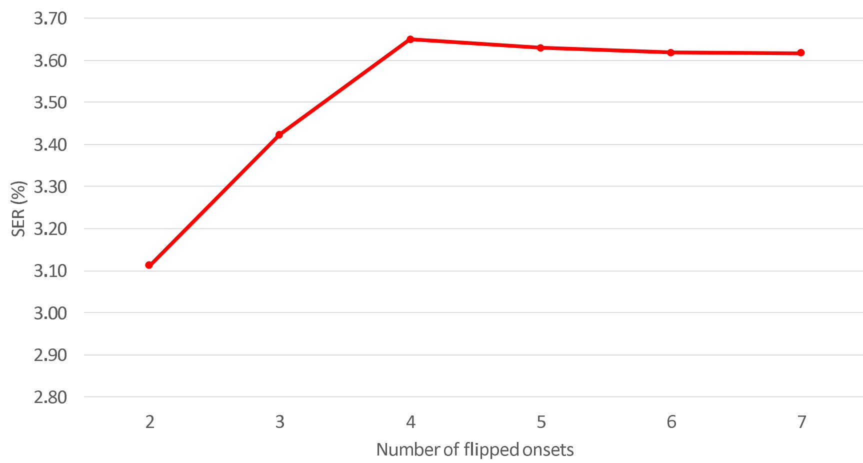

Fig. 2 SER of BFO for some varying number of flipped onsets

derivatives. For examples, a root "*beruju*" (youngest) is syllabified as ⟨be.ru.ju⟩ but a derivative "*berujud*" (tangible) is segmented into ⟨ber.u.jud⟩, where the prefix ⟨ber⟩ is split from the root ⟨u.jud⟩. The suffix ⟨i⟩ sometimes is also confused with the diphthong ⟨ai⟩. For instance, the word "*cintai*" (to love) is segmented into ⟨cin.ta.i⟩, where the suffix ⟨i⟩ is split from the root ⟨cin.ta⟩, while a root "*intai*" (to spy) is syllabified into ⟨in.tai⟩. A detailed observation shows that the syllabification errors are mostly dominated by these problems since Bahasa Indonesia has many affixes, i.e. seven prefixes, four infixes, and eighteen suffixes [3], those produce many derivatives with high similarity to some roots. The problem related to the suffix ⟨i⟩ and diphthong ⟨ai⟩ probably can be solved by adding a high accuracy preprocessing model of diphthong detection before developing the syllable-bigrams.

Generalization to other languages as well as name-entities. English has some polysyllabic words those can produce another words if their onsets are flipped. For examples, flipping two onsets in a word "**b**a.**s**in" (a bowl for washing) produces another word "**s**a.**b**in" (a vaccine against poliomyelitis), flipping onsets in "**c**a.**p**o" (the head of a crime syndicate) produces "**p**a.**c**o" (alpaca). Since BFO just exploits both syllable-bigrams and flipping onsets, it obviously can be applied to any language. It does not need any specific knowledge except the sets of vowels, diphthongs, and consonants. Flipping onsets is also common in name-entities. For instances, flipping two onsets in a name-entity "**t**o.**ky**o" (the capital of Japan) produces another name-entity "**ky**o.**t**o" (the old capital of Japan), flipping two onsets in "**b**er.**l**in" (the capital of Germany) yields "**l**er.**b**in" (a name of person), flipping two onsets in "i.**r**an" (a country in Western Asia) produces "**r**i.an" (a name of person), etc. In cases of name-entities, BFO also can be applied easily by providing a trainset of name-entities and three sets of symbols for vowels, diphthongs, and consonants. This advantage makes the BFO much simpler than the nearest neighbour-based syllabification proposed in [24], which highly depends on the specific-language knowledge of both phonemic and phonotactic rules.

4 Conclusion

The proposed simple scheme of flipping onsets of consecutive syllables in a word significantly improves the bigram orthographic syllabification model, where the SER relatively decreases up to 18.02%. The method is capable of producing quite low SER for a limited trainset of 1k words to generalize 10k unseen words. It can be generalized to be applied to other languages using a few specific knowledge related to the sets of vowels, diphthongs, and consonants. It is also possible to be exploited to syllabify name-entities. Compared to the nearest neighbour-based syllabification, it is slightly worse in accuracy but faster in computation and simpler to be generalized to other languages and name-entities. In the future, a diphthong detection can be added as a preprocessing procedure to solve some errors regarding the diphthongs.

Acknowledgements I would like to thank Muhammad Agha Ariyanto for the inspiration and all colleagues in Telkom University for the supports.

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. 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. 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)
7. Chaer, A.: Fonologi Bahasa Indonesia (Indonesian Phonology). Rineka Cipta, Jakarta (2009)
8. 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
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. 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
11. 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
12. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development (2001)
13. 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
14. 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
15. 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)
16. 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
17. 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
18. Kunchukuttan, A., Bhattacharyya, P.: Orthographic Syllable as basic unit for SMT between Related Languages. CoRR (2016). URL http://arxiv.org/abs/1610.00634
19. 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)
20. 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)
21. 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)
22. 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
23. 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)
24. Parande, E.A., Suyanto, S.: Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. International Journal of Speech Technology (2018). DOI 10.1007/s10772-018-09569-3. URL https://doi.org/10.1007/s10772-018-09569-3
25. 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
26. Rogova, K., Demuynck, K., Compernolle, D.V.: Automatic syllabification using segmental conditional random fields. Computational Linguistics in the Netherlands Journal **3**, 34–48 (2013)
27. 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)
28. 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

29. 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)
30. 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

Evidences of correspondences

Flipping Onsets to Enhances Syllabification

1. First submission (10 April 2019)
2. LoA with Minor Revision (25 September 2019)
3. Respond to Reviewers, Final submission (25 September 2019)
4. LoA with Fully Accepted (25 September 2019)

Telkom University

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

Decision on your Manuscript #IJST-D-19-00058

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Wed, Sep 25, 2019 at 7:37 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, "Flipping Onsets to Enhances Syllabification", which you submitted to International Journal of Speech Technology.

Based on the advice received, the Editor feels that your manuscript could be accepted for publication should you be prepared to incorporate minor 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. I would like to have your fine article published in the upcoming issue of the IJST. To do so, I would need you to please upload your revised paper by October 5th. I thank you so much for understanding. I look forward to seeing your fine work appear in the fall 2019 issue.

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/.

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 subject is very important and the author(s) have made a significant contribution to flipping onsets to enhance syllabification. The flow of the paper as well as performance discussion is good.

Reviewer #4: The content of the paper is well-described but the paper needs further proof reading.

There is a claim that BFO performed slightly worse than the nearest neighbour-based syllabification but BFO is much more efficient in computation. How is the efficiency computed and where in the paper this claim is validated?

View Letter - Google Chrome

editorialmanager.com/ijst/ViewLetter.aspx?id=63966&lsid={F8D5295A-B2E0-4A8C-A03B-571DA37EBD1C}

View Letter

Close

Date: 24 Sep 2019
To: "Suyanto Suyanto" suyanto@telkomuniversity.ac.id
From: "International Journal of Speech Technology (IJST)" Ramya.Thulasingam@springer.com
Subject: Decision on your Manuscript #IJST-D-19-00058

Dear Dr. Suyanto:

We have received the reports from our advisors on your manuscript, "Flipping Onsets to Enhances Syllabification", which you submitted to International Journal of Speech Technology.

Based on the advice received, the Editor feels that your manuscript could be accepted for publication should you be prepared to incorporate minor 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. I would like to have your fine article published in the upcoming issue of the IJST. To do so, I would need you to please upload your revised paper by October 5th. I thank you so much for understanding. I look forward to seeing your fine work appear in the fall 2019 issue.

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

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

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 subject is very important and the author(s) have made a significant contribution to flipping onsets to enhance syllabification. The flow of the paper as well as performance discussion is good.

Reviewer #4: The content of the paper is well-described but the paper needs further proof reading.

There is a claim that BFO performed slightly worse than the nearest neighbour-based syllabification but BFO is much more efficient in computation. How is the efficiency computed and where in the paper this claim is validated?

Evidences of correspondences

Flipping Onsets to Enhances Syllabification

1. First submission (10 April 2019)
2. LoA with Minor Revision (25 September 2019)
3. **Respond to Reviewers, Final submission (25 September 2019)**
4. LoA with Fully Accepted (25 September 2019)

Reviewer #1: The subject is very important and the author(s) have made a significant contribution to flipping onsets to enhance syllabification. The flow of the paper as well as performance discussion is good.

>> Thank you very much

Reviewer #4: The content of the paper is well-described but the paper needs further proof reading.

There is a claim that BFO performed slightly worse than the nearest neighbour-based syllabification but BFO is much more efficient in computation. How is the efficiency computed and where in the paper this claim is validated?

>> Proof reading has been done: some typographic errors are now corrected. A new paragraph below is now added to give a validation that the proposed BFO is much more efficient in computation than the nearest neighbour-based syllabification.

Comparison to another model. In this experiment, BFO is compared to another model of syllabification based on Fuzzy *k*-Nearest Neighbour in every Class (FkNNC) using the same dataset of 50 k words with 5-folds cross-validation described in [24]. The results show that BFO is slightly worse than FkNNC, where BFO produces higher averaged SER of 3.11% than FkNNC that gives averaged SER of 2.27%. But, BFO is much more efficient in computation since it just calculates the probabilities of candidate syllables taking into account a few (tens or less) possible bigrams with flipping onsets to define the syllabification points, as illustrated in both Table 3 and Table 5. In contrast, FkNNC has to calculate the distances between a candidate pattern of syllabification and all (250 thousands) patterns in the trainset, then select the *k* nearest neighbour patterns in both class of syllabification-point and not-syllabification-point, and finally find the lowest total fuzzy-distance to decide if the candidate is a syllabification point or not.

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

Acknowledgement of Receipt of #IJST-D-19-00058R1

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Wed, Sep 25, 2019 at 7:58 PM
Reply-To: "International Journal of Speech Technology (IJST)" <ramya.thulasingam@springer.com>
To: Suyanto Suyanto <suyanto@telkomuniversity.ac.id>

Dear Dr. Suyanto:

We acknowledge, with thanks, receipt of the revised version of your manuscript, "Flipping Onsets to Enhance Syllabification", submitted to International Journal of Speech Technology. The manuscript number is IJST-D-19-00058R1.

You may check the status of your manuscript through 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/.

We will inform you of the Editor's decision as soon as possible.

Best regards,
Springer Journals Editorial Office
International Journal of Speech Technology

International Journal of Speech Technology

Flipping Onsets to Enhance Syllabification

--Manuscript Draft--

Manuscript Number:	IJST-D-19-00058R1
Full Title:	Flipping Onsets to Enhance Syllabification
Article Type:	Manuscript
Keywords:	bigram; consecutive syllables; flipping onsets; orthographic syllabification
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, Dr.
First Author Secondary Information:	
Order of Authors:	Suyanto Suyanto, Dr.
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	Two-year-old children who start learning to speak generally spell a polysyllabic word by flipping onsets of consecutive syllables. Sometimes they speak unclearly, hard to understand since the flipped onsets produce another word that has a much different meaning. For instance, two onsets in an English word "me.lon" (large round fruit of a plant of the gourd family) are flipped to produce another word "le.mon" (an acid fruit). In Bahasa Indonesia, such cases are quite common. For examples, two onsets in word "ba.tu" (stone) are swapped to be "ta.bu" (taboo), two onsets in "be.sar" (big) are flipped to be "se.bar" (spread), two onsets in "ru.mah" (house) are swapped to be "mu.rah" (cheap), etc. A preliminary study on 50k Indonesian formal words shows that the ratio between frequencies of the flipped-onset-bigrams and the 50 most frequent original syllable-bigrams is quite high, up to 13.09%. This research investigates the adoption of such phenomenon to enhances a bigram orthographic syllabification model that is commonly poor for out-of-vocabulary words. A 5-fold cross-validation on 50k Indonesian formal words proves that the flipping onsets enhances the bigram orthographic syllabification, where the syllable error rate (SER) is relatively reduced by 18.02%. The method is also capable of producing quite low SER for a tiny trainset of 1k words to generalize 10k unseen words. Besides, it can be simply generalized to be applied to other languages as well as name-entities using a few specific knowledge related to the sets of vowels, diphthongs, and consonants.

Reviewer #1: The subject is very important and the author(s) have made a significant contribution to flipping onsets to enhance syllabification. The flow of the paper as well as performance discussion is good.

>> Thank you very much

Reviewer #4: The content of the paper is well-described but the paper needs further proof reading.

There is a claim that BFO performed slightly worse than the nearest neighbour-based syllabification but BFO is much more efficient in computation. How is the efficiency computed and where in the paper this claim is validated?

>> Proof reading has been done: some typographic errors are now corrected. A new paragraph below is now added to give a validation that the proposed BFO is much more efficient in computation than the nearest neighbour-based syllabification.

Comparison to another model. In this experiment, BFO is compared to another model of syllabification based on Fuzzy *k*-Nearest Neighbour in every Class (FkNNC) using the same dataset of 50 k words with 5-folds cross-validation described in [24]. The results show that BFO is slightly worse than FkNNC, where BFO produces higher averaged SER of 3.11% than FkNNC that gives averaged SER of 2.27%. But, BFO is much more efficient in computation since it just calculates the probabilities of candidate syllables taking into account a few (tens or less) possible bigrams with flipping onsets to define the syllabification points, as illustrated in both Table 3 and Table 5. In contrast, FkNNC has to calculate the distances between a candidate pattern of syllabification and all (250 thousands) patterns in the trainset, then select the *k* nearest neighbour patterns in both class of syllabification-point and not-syllabification-point, and finally find the lowest total fuzzy-distance to decide if the candidate is a syllabification point or not.

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

Flipping Onsets to Enhance Syllabification

Suyanto Suyanto

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)

Flipping Onsets to Enhance Syllabification

Received: date / Accepted: date

Abstract Two-year-old children who start learning to speak generally spell a polysyllabic word by flipping onsets of consecutive syllables. Sometimes they speak unclearly, hard to understand since the flipped onsets produce another word that has a much different meaning. For instance, two onsets in an English word ”**m**e.**l**on” (large round fruit of a plant of the gourd family) are flipped to produce another word ”**l**e.**m**on” (an acid fruit). In Bahasa Indonesia, such cases are quite common. For examples, two onsets in word ”**b**a.**t**u” (stone) are swapped to be ”**t**a.**b**u” (taboo), two onsets in ”**b**e.**s**ar” (big) are flipped to be ”**s**e.**b**ar” (spread), two onsets in ”**r**u.**m**ah” (house) are swapped to be ”**m**u.**r**ah” (cheap), etc. A preliminary study on 50k Indonesian formal words shows that the ratio between frequencies of the flipped-onset-bigrams and the 50 most frequent original syllable-bigrams is quite high, up to 13.09%. This research investigates the adoption of such phenomenon to enhances a bigram orthographic syllabification model that is commonly poor for out-of-vocabulary words. A 5-fold cross-validation on 50k Indonesian formal words proves that the flipping onsets enhances the bigram orthographic syllabification, where the syllable error rate (SER) is relatively reduced by 18.02%. The method is also capable of producing quite low SER for a tiny trainset of 1k words to generalize 10k unseen words. Besides, it can be simply generalized to be applied to other languages as well as named-entities using a few specific knowledge related to the sets of vowels, diphthongs, and consonants.

Keywords bigram · consecutive syllables · flipping onsets · orthographic syllabification

1 Introduction

A syllable is very important in phonology since it is relevant to the phonological rules. It is also a basis to describe stresses in a language. Breaking a word into syllables automatically, commonly called automatic syllabification, is one of the interesting researches since it relates to many other fields of studies as well as applications. The automatic syllabification is commonly used in both syllable-based speech recognition [30], [13], [19] and speech synthesis systems [25]. It is also widely exploited in spell-checking [4], [23], information retrieval [14], and statistical machine translation [18].

In general, an automatic syllabification can be applied to either grapheme or phoneme sequences. For many formal words, a phonemic syllabification commonly gives lower SER than an orthographic (also known as graphemic) syllabification, as described in [28] and [24]. However, the phonemic syllabification needs a perfect phoneme sequence that generally cannot be provided by any G2P model, but by a linguist. Moreover, it looks to be getting worse (produce a higher SER) when it is applied for a named-entity since this problem has a high dynamic and ambiguity. These facts make the graphemic syllabification is more widely used in practice than the phonemic one since it does not need any module of grapheme-to-phoneme conversion (G2P).

The researchers mostly prefer to develop automatic syllabifications using a statistical-based approach rather than the rule-based one [2], [26]. It can be implemented using two different techniques: supervised and unsupervised learning. The supervised learning is more popular and commonly implemented using either global or local classification method.

Some global classifications to develop a syllabification model are neural networks [8], [17], [29], decision tree [9], treebank [21], support vector machine [5], hidden Markov model [16], finite-state transducers [11], [15], context-free grammars [22], syllabification by analogy [1], unsupervised method [20], joint n-gram models [27], and segmental conditional random fields (SCRF) [26]. Meanwhile, the syllabifications based on local classifications are generally developed using nearest neighbour-based methods as described in [24] and [28]. The local classifications are quite accurate but time consuming.

Some methods are designed to be low-cost as well as language-independent. One of them is the bigram syllabification model. Although it is commonly poor for low-resource languages with so many out-of-vocabulary (OOV) words, some features can be added to improve its performance. One of the improved bigram syllabifications is SCRF [26]. The main advantage of SCRF is high generalization with limited trainset. But, this method is quite complex. It incorporates eight features produced by three general principles of syllabification (sonority, legality, and maximum onset) to calculate the syllable bigram probability. Besides, it is applied to a phoneme string that is relatively easier to solve than a grapheme string.

In this research, a new simpler scheme of flipping onsets contained in some consecutive syllables is incorporated into a bigram-based method. Its effect is investigated on an orthographic syllabification for Bahasa Indonesia as a low resource language. First, a standard bigram is implemented as a baseline model. A bigram with flipping onsets is then developed and then investigated to see if it is capable of reducing the SER significantly.

The proposed scheme is inspired by an interesting phenomenon of two-year-old children who start learning to speak. They generally spell a polysyllabic word by flipping onsets of two first consecutive syllables. Sometimes they speak unclearly, hard to understand, since flipping those onsets may produce another word that have much different meaning. For examples, an English word "**b**a.**s**in" (a bowl for washing) is spelled as "**s**a.**b**in" (a vaccine against poliomyelitis), "**c**a.**p**o" (the head of a crime syndicate) is spoken as "**p**a.**c**o" (alpaca), and "**m**e.**l**on" (large round fruit of a plant of the gourd family) is said as "**l**e.**m**on" (an acid fruit). In Bahasa Indonesia, such cases are quite common. For instance, onsets of two syllables in word "**b**a.**w**ah" (under) are flipped to be "**w**a.**b**ah" (epidemic), onsets in "**b**e.**k**al" (stock) are swapped to be "**k**e.**b**al" (immune), onsets in "**b**e.**s**ar" (big) are flipped to be "**s**e.**b**ar" (spread), onsets in "**k**e.**s**al" (upset) are swapped to be "**s**e.**k**al" (plump), etc.

A preliminary observation on 50k Indonesian formal words shows an interesting fact. Table 1 illustrates the top 50 most frequent syllable-bigrams from the 50k words, where their total frequency is 7,394. The interesting fact is the flipped-onset-bigrams corresponding to those syllable-bigrams have a relatively high frequency of 968. This produces a high ratio between the frequency of the flipped-onset-bigram and the original syllable-bigram, i.e. 13.09%. This ratio can be probably higher when the most frequent syllable-bigrams taken into account are more than 50. Hence, the proposed scheme of flipping onsets is expected to relatively reduce the SER by at least 13.09%.

However, two-year-old children sometimes also flip onsets in the second and the third syllables. For instance, the onsets in an Indonesian formal word "ke.**p**a.**l**a" (head) are flipped to be "ke.**l**a.**p**a" (coconut). Hence, flipping onsets can be generalized into more than two consecutive syllables. This phenomenon is also investigated by doing some experiments with varying number of consecutive syllables from 2 to 7. This is motivated by a fact that Bahasa Indonesia has some long words containing up to 7 syllables or more, where on average it has 3.20 syllables in a word [28].

2 Flipping Onsets to Enhances a Bigram Syllabification

A word is composed of one or more syllables. A syllable contains a non-obligatory onset, followed by an obligatory nucleus, and ended by a non-obligatory coda [12]. The nucleus can be a single vowel, a diphthong or, in some languages such as English, a sonorant consonant. In Bahasa Indonesia, the nucleus is always either a single vowel or a diphthong while onset and coda are always consonants [3], [7]. For example, an Indonesian word "*besar*" (big) is composed of two syllables, ⟨be⟩ and ⟨sar⟩. The first syllable contains a nucleus ⟨e⟩ that is preceded by an onset ⟨b⟩ but not followed by any coda. Meanwhile, the second syllable consists of a nucleus ⟨a⟩ with an onset ⟨s⟩ and a coda ⟨r⟩.

Hence, the number of syllables contained in an Indonesia word should equal to the number of vowels and/or diphthongs [3]. This simplifies the problem

Table 1 Top 50 frequent syllable-bigrams generated from 50k Indonesian formal words collected from KBBI

Rank	Syllable-bigram	Frequency	Flipped-onsets-bigram	Frequency
1	mem.per	382	pem.mer	0
2	me.nga	367	nge.ma	2
3	me.nge	330	nge.me	0
4	me.nye	295	nye.me	1
5	sa.si	251	sa.si	251
6	lo.gi	219	go.li	1
7	me.ma	181	me.ma	181
8	si.a	181	i.sa	4
9	pe.nga	174	nge.pa	3
10	me.ngu	172	nge.mu	8
11	ber.ke	162	ker.be	0
12	me.na	161	ne.ma	15
13	me.ne	161	ne.me	0
14	si.o	152	i.so	63
15	ka.si	150	si.ka	9
16	me.ngi	148	nge.mi	0
17	ta.si	145	sa.ti	5
18	me.ra	140	re.ma	12
19	me.ter	140	te.mer	0
20	ra.si	139	sa.ri	27
21	me.nya	138	nye.ma	7
22	si.kan	136	ki.san	9
23	me.la	132	le.ma	12
24	meng.ge	132	geng.me	0
25	ber.se	128	ser.be	0
26	pe.nye	128	nye.pe	2
27	li.sa	126	si.la	41
28	mem.be	124	bem.me	0
29	ber.a	120	er.ba	0
30	mem.ba	120	bem.ma	0
31	pe.ra	116	re.pa	6
32	me.le	115	le.me	3
33	na.si	115	sa.ni	13
34	me.me	114	me.me	114
35	pe.nge	113	nge.pe	0
36	men.de	112	den.me	0
37	me.re	110	re.me	6
38	la.si	106	sa.li	0
39	a.si	103	sa.i	18
40	a.li	101	la.i	9
41	me.nu	101	ne.mu	6
42	di.a	100	i.da	10
43	pe.la	99	le.pa	10
44	me.nyu	97	nye.mu	1
45	te.ra	97	re.ta	10
46	me.mu	96	me.mu	96
47	pa.ra	94	ra.pa	11
48	me.ngo	92	nge.mo	1
49	gra.fi	90	fa.gri	0
50	li.tas	89	ti.las	1
Total		7,394		968

of finding the correct syllabification since it limits the possible syllabification points or boundaries in between vowels (and/or diphthongs). For instance, the word "*besar*" produces only two candidate syllabifications, i.e. ⟨be.sar⟩ and ⟨bes.ar⟩, while two others ⟨b.esar⟩ and ⟨besa.r⟩ are illegal syllabifications.

Vowels and consonants. In a text, some vowels and diphthongs can be automatically detected using some methods, such as Sukhotin's algorithm and its combinations those are language-independent unsupervised learning [10]. Evaluations on 39 languages contain more than 100,000 symbols each show that the average accuracy of Sukhotin's algorithm is 95.66% [20]. Since Bahasa Indonesia has simple typological knowledge, this research does not exploit Sukhotin's algorithm. As described in [3], Bahasa Indonesia has five graphemes those are single vowels {⟨a⟩, ⟨e⟩, ⟨i⟩, ⟨o⟩, ⟨u⟩}, four sequences of graphemes those form diphthongs {⟨ai⟩, ⟨au⟩, ⟨ei⟩, ⟨oi⟩}, and twenty one graphemes those are consonants {⟨b⟩, ⟨c⟩, ⟨d⟩, ⟨f⟩, ⟨g⟩, ⟨h⟩, ⟨j⟩, ⟨k⟩, ⟨l⟩, ⟨m⟩, ⟨n⟩, ⟨p⟩, ⟨q⟩, ⟨r⟩, ⟨s⟩, ⟨t⟩, ⟨v⟩, ⟨w⟩, ⟨x⟩, ⟨y⟩, ⟨z⟩}.

Monophthong and diphthong. One of challenges in automatic syllabification is how to distinguish monophthong and diphthong. Two consecutive graphemes ⟨ai⟩ in the word "*cintai*" (to love) are two monophthongs ⟨a⟩ and ⟨i⟩ so that the word is syllabified as ⟨cin.ta.i⟩ while ⟨ai⟩ in the word "*intai*" (to spy) is a diphthong so that it is segmented as ⟨in.tai⟩. In bigram syllabification, the problem regarding any word that probably contains a diphthong can be solved by maximizing the bigram probabilities of all possible syllabifications. The word "*cintai*" produces six candidates, i.e. ⟨ci.nta.i⟩, ⟨cin.ta.i⟩, and ⟨cint.a.i⟩ (where ⟨ai⟩ are assumed as two monophthongs) as well as ⟨ci.ntai⟩, ⟨cin.tai⟩, and ⟨cint.ai⟩ (where ⟨ai⟩ is hypothesized as a diphthong).

Bigram syllabification model. The bigram syllabification can be seen as finding the most likely syllable sequence that represents a word. The probability of a bigram syllabification of L tokens $P(w_1, w_2, ..., w_L)$ is calculated using the probability chain. In practice, there are so many smoothing methods to estimate the probability to solve the problem of sparsity affected by the OOV words. One of them is the *Stupid Backoff* that is introduced in [6]. In the *Stupid Backoff* smoothing, the probability is estimated using a score S that may have a value bigger than 1

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

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

where $f(w_{i-1}w_i)$ and $f(w_i)$ denotes the frequencies of occurrences of both syllable-bigrams and syllable-unigrams in the trainset respectively, α is the backoff factor heuristically set to 0.4, and N is the trainset size [6]. In practical implementation, the score for an unseen syllable-unigram, a case where $f(w_i) = 0$, is approximated as the score for syllable-unigram seen once that produces $S(w_i|w_{i-1}) = \alpha \frac{1}{(N+1)}$. The final score is computed using a logarithmic formula $\frac{1}{-\log(S)}$ to avoid producing an underflow value.

Proposed model. In general, the bigram syllabification model is poor for a low trainset with so many OOV syllable-bigrams [26]. In case of Bahasa Indonesia, a relatively low trainset of 50k words contains a vocabulary $V = 2,267$ distinct syllable-unigrams and a total token $N = 33,147$ unique syllable-bigrams. This is only 0.64% ($33,147$ out of $2,267^2 = 5,139,289$ possible syllable-bigrams). It means there are 99.36% unseen syllable-bigrams. In other words, it has high OOV syllable-bigrams. To solve the problem, a simple modification of the bigram syllabification model is proposed by incorporating a procedure of flipping onsets of two or more consecutive syllables contained in a word. Pseudocode of the proposed method, called bigram with flipping onsets (BFO), can be briefly described in three steps below:

1. From a given grapheme sequence, detect all positions of vowels as well as possible diphthongs and do the procedure of bigram syllabification with *Stupid Backoff* smoothing, i.e. generate all C candidate syllabifications and compute their scores S_i, where $i = 1, 2, ..., C$, using Equation 1 and Equation 2;
2. For each candidate syllabification do flipping onsets in B consecutive syllables to generate new $(B-1)$ candidates, where B is an integer bigger than 2, and compute the average score $\bar{S}_i$ by taking into account the previous corresponding score in step 1; and
3. Finally, select the ith candidate with the highest average score $\bar{S}_i$ as the best syllabification.

The comparison of Bigram and BFO can be simply explained using five illustrations in Table 2 to Table 6, where both model are developed using a trainset of 40k words. In Table 2, the bigram syllabification segments the word containing possible diphthong "*hasai*" (fragile) into a wrong syllabification $\langle$ha.sa.i$\rangle$. In contrast, Table 3 shows that BFO correctly gives a true syllabification $\langle$ha.sai$\rangle$. A more complex example is syllabification of a word "*berani*" (brave) that has three syllables. In Table 4, the bigram syllabification segments the word "*berani*" into a wrong syllabification $\langle$ber.a.ni$\rangle$. In contrast, Table 5 and Table 6 shows that BFO with two and three flipped onsets successfully produces a true syllabification $\langle$be.ra.ni$\rangle$.

3 Result and Discussion

In this research, the dataset of 50k Indonesian formal words with their syllabification points is used. It is the same dataset as described in [24]. Two experiments are conducted to investigate the effects of both trainset sizes and

Table 2 Bigram syllabification breaks the word "*hasai*" into a wrong syllabification ⟨ha.sa.i⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
ha.sa.i	0.0677	has.a.i	0.0468	ha.sai	0.0660	has.ai	0.0533
$\bar{S}_1$	**0.0677**	$\bar{S}_2$	0.0468	$\bar{S}_3$	0.0660	$\bar{S}_4$	0.0533

Table 3 BFO correctly segments a word "*hasai*" into ⟨ha.sai⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
ha.sa.i	0.0677	has.a.i	0.0468	ha.sai	0.0660	has.ai	0.0533
sa.ha.i	0.0611	as.ha.i	0.0485	sa.hai	0.0639	as.hai	0.0571
$\bar{S}_1$	0.1288	$\bar{S}_2$	0.0953	$\bar{S}_3$	**0.1299**	$\bar{S}_4$	0.1104

Table 4 Bigram syllabification breaks the word "*berani*" into a wrong syllabification ⟨ber.a.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0776	be.ran.i	0.0634	ber.a.ni	0.0836	ber.an.i	0.0661
$\bar{S}_1$	0.0776	$\bar{S}_2$	0.0634	$\bar{S}_3$	**0.0836**	$\bar{S}_4$	0.0661

Table 5 BFO with two flipped onsets correctly syllabifies the word "*berani*" into ⟨be.ra.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0836	be.ran.i	0.0634	ber.a.ni	0.0776	ber.an.i	0.0661
re.**b**a.ni	0.0588	**re**.**b**an.i	0.0549	er.**b**a.ni	0.0566	er.**b**an.i	0.0435
$\bar{S}_1$	**0.0712**	$\bar{S}_2$	0.0592	$\bar{S}_3$	0.0671	$\bar{S}_4$	0.0548

Table 6 BFO with three flipped onsets correctly segments a word "*berani*" into ⟨be.ra.ni⟩

Candidate 1		Candidate 2		Candidate 3		Candidate 4	
Syllabific.	S_1	Syllabific.	S_2	Syllabific.	S_3	Syllabific.	S_4
be.ra.ni	0.0776	be.ran.i	0.0634	ber.a.ni	0.0836	ber.an.i	0.0661
re.**b**a.ni	0.0588	**re**.**b**an.i	0.0549	er.**b**a.ni	0.0566	er.**b**an.i	0.0435
be.**na**.**r**i	0.0765	be.an.**r**i	0.0575	ber.**n**a.i	0.0658	ber.an.i	0.0661
$\bar{S}_1$	**0.0710**	$\bar{S}_2$	0.0586	$\bar{S}_3$	0.0687	$\bar{S}_4$	0.0586

number of flipped onsets. The performance is measured using a syllable error rate (the percentage of syllable error).

Trainset sizes. First, the 50k words in the dataset are randomly selected to produce six trainset sizes: 1k, 5k, 10k, 20k, 30, and 40k. The random selection is performed five times for each trainset size. The testset size is fixed to 10k. Next, the experiments to investigate the performance of both Bigram and BFO syllabifications are repeated five times for each trainset size to reduce the random effect. The results illustrated by Fig. 1 shows that BFO syllabification produces lower average SERs for all trainset sizes than the Bigram one. For the smallest trainset size of 1k, BFO produces lower average SER of 16.13%

than Bigram that reaches up to 19.30%. This result shows that BFO has a high generalization, where it gives a quite low SER for a much bigger testset of 10k unseen words based on a very small trainset of 1k words. For the trainset size of 10k, which is the same as the testset size, BFO produces much lower average SER of 4.96% than Bigram that gives 6.29%. For the biggest trainset size of 40k, BFO also produces lower average SER of 3.11% than Bigram that gives 3.80%. It means the proposed scheme relatively reduces the SER by 18.02% as expected.

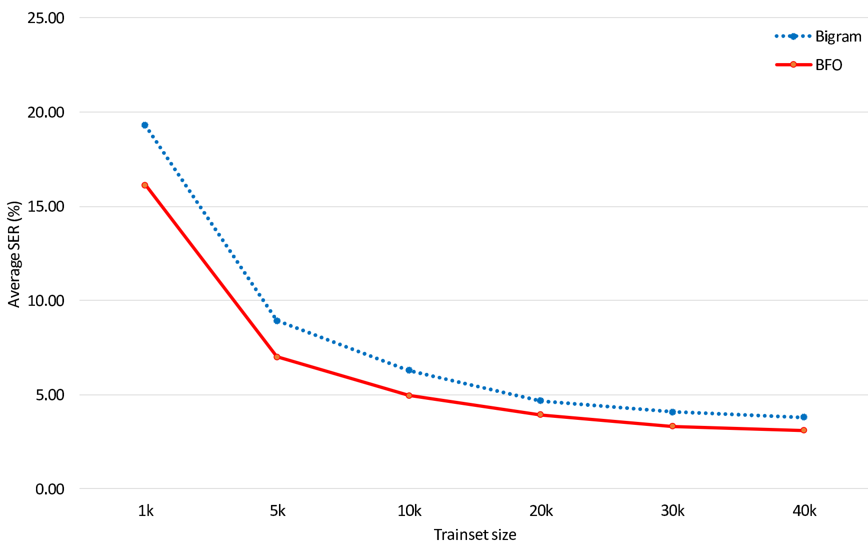

Fig. 1 Comparison of Bigram and BFO for some varying sizes of trainsets. BFO gives lower SERs for all trainset sizes, especially for the smallest one

Comparison to another model. In this experiment, BFO is compared to another model of syllabification based on Fuzzy k-Nearest Neighbour in every Class (FkNNC) using the same dataset of 50 k words with 5-folds cross-validation described in [24]. The results show that BFO is slightly worse than FkNNC, where BFO produces higher averaged SER of 3.11% than FkNNC that gives averaged SER of 2.27%. But, BFO is much more efficient in computation since it just calculates the probabilities of candidate syllables taking into account a few (tens or less) possible bigrams with flipping onsets to determine the syllabification points, as illustrated in both Table 3 and Table 5. In contrast, FkNNC has to calculate the distances between a candidate pattern of syllabification and all (250 thousands) patterns in the trainset, then select the k nearest neighbour patterns in both class of syllabification point and not syllabification point, and finally find the lowest total fuzzy-distance to decide if the candidate is a syllabification point or not.

Number of flipped onsets. In this experiment, the dataset is randomly divided into five distinct subsets to do the 5-fold cross-validation. The experimental results illustrated by Fig. 2 shows that flipping two onsets produces the lowest average SER of 3.11%. The SER slightly increases with increasing number of flipped onsets. The detail observations show that taking into account more than two flipped onsets produces some biases in the average scores of the candidate syllabifications. This result follows a phenomenon that two-year-old children frequently flip the two first onsets, but rarely flip three or more onsets, contained in a word.

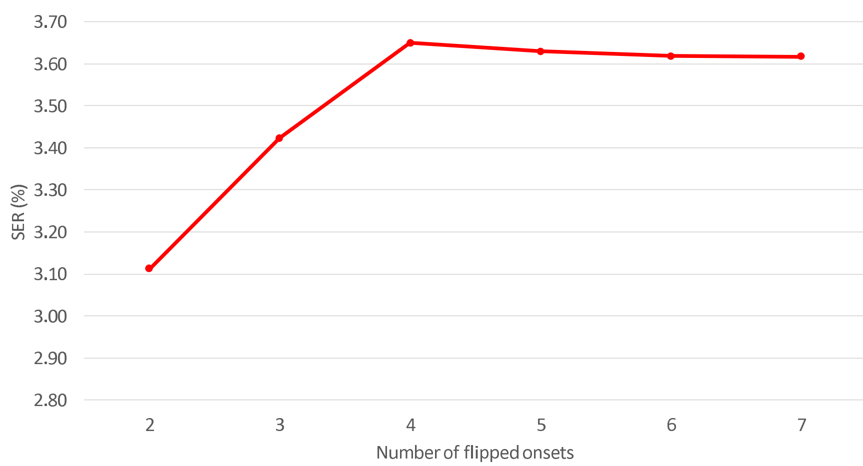

Fig. 2 SER of BFO for some varying number of flipped onsets

Hard-to-solve problems. The proposed method sometimes fails to syllabify some ambiguous words, which come from some roots those are similar to derivatives. For examples, a root "*beruju*" (youngest) is syllabified as ⟨be.ru.ju⟩ but a derivative "*berujud*" (tangible) is segmented into ⟨ber.u.jud⟩, where the prefix ⟨ber⟩ is split from the root ⟨u.jud⟩. The suffix ⟨i⟩ sometimes is also confused with the diphthong ⟨ai⟩. For instance, the word "*cintai*" (to love) is segmented into ⟨cin.ta.i⟩, where the suffix ⟨i⟩ is split from the root ⟨cin.ta⟩, while a root "*intai*" (to spy) is syllabified into ⟨in.tai⟩. A detailed observation shows that the syllabification errors are mostly dominated by these problems since Bahasa Indonesia has many affixes, i.e. seven prefixes, four infixes, and eighteen suffixes [3], those produce many derivatives with high similarity to some roots. The problem related to the suffix ⟨i⟩ and diphthong ⟨ai⟩ probably can be solved by adding a high accuracy preprocessing model of diphthong detection before developing the syllable-bigrams.

Generalization to other languages as well as named-entities. English has some polysyllabic words those can produce another words if their onsets are flipped. For examples, flipping two onsets in a word "**b**a.**s**in" (a bowl for washing) produces another word "**s**a.**b**in" (a vaccine against poliomyelitis), flipping onsets in "**c**a.**p**o" (the head of a crime syndicate) produces "**p**a.**c**o" (alpaca). Since BFO just exploits both syllable-bigrams and flipping onsets, it obviously can be applied to any language. It does not need any specific knowledge except the sets of vowels, diphthongs, and consonants. Flipping onsets is also common in named-entities. For instances, flipping two onsets in a named-entity "**t**o.**ky**o" (the capital of Japan) produces another named-entity "**ky**o.**t**o" (the old capital of Japan), flipping two onsets in "**b**er.**l**in" (the capital of Germany) yields "**l**er.**b**in" (a name of person), flipping two onsets in "i.**r**an" (a country in Western Asia) produces "**r**i.an" (a name of person), etc. In cases of named-entities, BFO also can be applied easily by providing a trainset of named-entities and three sets of symbols for vowels, diphthongs, and consonants. This advantage makes the BFO much simpler than the nearest neighbour-based syllabification proposed in [24], which highly depends on the specific-language knowledge of both phonemic and phonotactic rules.

4 Conclusion

The proposed simple scheme of flipping onsets of consecutive syllables in a word significantly improves the bigram orthographic syllabification model, where the SER relatively decreases up to 18.02%. The method is capable of producing quite low SER for a limited trainset of 1k words to generalize 10k unseen words. It can be generalized to be applied to other languages using a few specific knowledge related to the sets of vowels, diphthongs, and consonants. It is also possible to be exploited to syllabify named-entities. Compared to the nearest neighbour-based syllabification, it is slightly worse in accuracy but faster in computation and simpler to be generalized to other languages and named-entities. In the future, a diphthong detection can be added as a preprocessing procedure to solve some errors regarding the diphthongs.

Acknowledgements I would like to thank my wife, Ari Virgandini, as well as my beloved sons, Muhammad Arkan Ariyanto and Muhammad Agha Ariyanto, for the great inspiring ideas of flipping your onsets, and also all colleagues in Telkom University for the supports.

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. 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. 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)
7. Chaer, A.: Fonologi Bahasa Indonesia (Indonesian Phonology). Rineka Cipta, Jakarta (2009)
8. 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
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. 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
11. 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
12. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development (2001)
13. 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
14. 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
15. 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)
16. 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
17. 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
18. Kunchukuttan, A., Bhattacharyya, P.: Orthographic Syllable as basic unit for SMT between Related Languages. CoRR (2016). URL http://arxiv.org/abs/1610.00634
19. 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)
20. 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)
21. 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)
22. 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

23. 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)
24. Parande, E.A., Suyanto, S.: Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure. International Journal of Speech Technology (2018). DOI 10.1007/s10772-018-09569-3. URL https://doi.org/10.1007/s10772-018-09569-3
25. 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
26. Rogova, K., Demuynck, K., Compernolle, D.V.: Automatic syllabification using segmental conditional random fields. Computational Linguistics in the Netherlands Journal **3**, 34–48 (2013)
27. 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)
28. 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
29. 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)
30. 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

Evidences of correspondences

Flipping Onsets to Enhances Syllabification

1. First submission (10 April 2019)
2. LoA with Minor Revision (25 September 2019)
3. Respond to Reviewers, Final submission (25 September 2019)
4. **LoA with Fully Accepted** (25 September 2019)

Telkom University

SUYANTO SUYANTO <suyanto@telkomuniversity.ac.id>

Decision on your manuscript #IJST-D-19-00058R1

1 message

International Journal of Speech Technology (IJST) <em@editorialmanager.com> Wed, Sep 25, 2019 at 10:48 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, "Flipping Onsets to Enhance Syllabification" 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-19-00058R1, whenever inquiring about your manuscript.
With best regards,
Amy Neustein, Ph.D.
Editor-in-Chief
International Journal of Speech Technology