Evidences of correspondences

Indonesian Syllabification Using
Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. **First Submission (07 January 2016)**
2. LoA with Major Revision (22 September 2016)
3. Respond to Reviewers (11 October 2019)
4. Second Submission (11 October 2019)
5. LoA with Minor Revision (21 October 2019)
6. Final Submission (24 October 2019)
7. LoA with Fully Accepted (28 October 2019)

Manuscript details

Manuscript number	SPECOM_2016_8
Title	Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge
Article type	Full Length Article
Abstract	This paper discusses a phonemic syllabification using pseudo nearest neighbour rule (PNNR) and phonotactic knowledge for the Indonesian language. In this data-driven model, a four-feature phoneme encoding and a phonotactic-based pre-syllabification are proposed. Evaluating on five datasets with trainset of 40K words and testset of 10K words each shows the proposed encoding significantly reduces the average syllable error rate (SER) by 13.90% relatively to the commonly used orthogonal binary encoding and the pre-syllabification also reduces the average SER up to 17.17% relatively to the PNNR without pre-syllabification. Five fold cross validating proofs that the proposed PNNR-based syllabification is stable by producing average SER of 0.64%. The most errors come from some derivatives with three prefixes /ber/, /per/, and /ter/ as well as some compound words.
Keywords	Indonesian syllabification; four-feature phoneme encoding; phonotactic knowledge; pseudo nearest neighbour rule
Manuscript category	Regular Paper

Corresponding Author Suyanto Suyanto

Corresponding Author's Institution Telkom University

Order of Authors Suyanto Suyanto, Sri Hartati, Agus Harjoko, Dirk Van Compernolle

Submission files included in this PDF

File Type	File Name
Manuscript File	SuyantoIndonesianSyllabification06.pdf
Cover Letter	Suyanto_cover_letter.pdf

Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

Suyanto[a,1,*], Sri Hartati[a], Agus Harjoko[a], Dirk Van Compernolle[b]

[a] *Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia*
[b] *Departement Elektrotechniek-ESAT, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium*

Abstract

This paper discusses a phonemic syllabification using pseudo nearest neighbour rule (PNNR) and phonotactic knowledge for the Indonesian language. In this data-driven model, a four-feature phoneme encoding and a phonotactic-based pre-syllabification are proposed. Evaluating on five datasets with trainset of 40K words and testset of 10K words each shows the proposed encoding significantly reduces the average syllable error rate (SER) by 13.90% relatively to the commonly used orthogonal binary encoding and the pre-syllabification also reduces the average SER up to 17.17% relatively to the PNNR without pre-syllabification. Five fold cross validating proofs that the proposed PNNR-based syllabification is stable by producing average SER of 0.64%. The most errors come from some derivatives with three prefixes /ber/, /per/, and /ter/ as well as some compound words.

Keywords: Indonesian syllabification, four-feature phoneme encoding, phonotactic knowledge, pseudo nearest neighbour rule

---

*Corresponding author

Email addresses: `suyanto@telkomuniversity.ac.id` (Suyanto), `shartati@ugm.ac.id` (Sri Hartati), `aharjoko@ugm.ac.id` (Agus Harjoko), `Dirk.VanCompernolle@esat.kuleuven.be` (Dirk Van Compernolle)

[1]Lecturer at School of Computing, Telkom University (former: Telkom Institute of Technology), Bandung, West Java, 40257 Indonesia

1. Introduction

There are two approaches to the automatic syllabification: rule-based and data-driven. The rule-based approach, using sonority sequencing principle, legality principle, and maximal onset principle, produces ambiguous syllabification that is just valid for several cases [1]. This approach gives high accuracy for some languages with few simple and consistent syllabification rules, such as Sinhala [2] and Spanish [3]. But, it performs poorly for a slightly more complex syllabic language, such as Malay [4]. Hence, some researchers develop many data-driven methods, such as IGTree learning algorithm [5], weighted finite-state transducers [6], combination of treebank and bracketed corpora training [7], neural network [8], [9], [10], probabilistic context-free grammars [11], joint n-gram models [12], combination of support vector machine and hidden Markov model [13], syllabification by analogy (SbA) [14], counting the actual syllables to determine the best split of word-medial consonant sequences [15], segmental conditional random fields [1].

The data-driven approach gives higher accuracy than the rule-based one for English, a complex syllabic language [16]. It also performs better for a language with low syllabic complexity, such as Italian, where SbA reaches a word accuracy of 97.70% but the best rule set (SYL-LABE) achieves only 89.77% word accuracy for 44K words [17]. Adsett and Marchand in [14] proved that it generally produces higher accuracy for nine European languages: Basque, Dutch, English, French, Frisian, German, Italian, Norwegian, and Spanish [14], where SbA gives the highest average word accuracy of 96.84% (standard deviation of 2.93) whereas Liang's algorithm produces a mean of 95.67% (standard deviation of 5.70).

Based on the method of classifying languages proposed by Dauer in [18], the Indonesian is categorized as a simple syllabic language since it has majority of CV syllables (where C is a single consonant and V is a single vowel) and more open syllables than closed. A study of a vocabulary of 50K words, collected from the great dictionary of the Indonesian language (*Kamus Besar Bahasa*

Indonesia, KBBI) developed by *Pusat Bahasa*, shows that the Indonesian has 50.63% CV syllables and 56.63% open syllables, as listed in table 1. In contrast, English, as a complex syllabic language, has about 35% of CV syllables and a wider variety of both open and closed syllables [18]. However, the Indonesian is a syllabically rich language. An observation of the 50K words shows that the Indonesian has 3.20 syllables per word and 7.64 phonemes per word in average. It has 98.30% polysyllabic words, much more than the monosyllabic ones that only account for 1.70%. It even has some very long words containing seven or more syllables, e.g. '*merestrukturisasi*' /mə.rɛ.struk.tu.ri.sɑ.si/ (restructure), '*semaksimal-maksimalnya*' /sə.mɑk.si.mɑɫ-mɑk.si.mɑɫ.ɲɑ/ (maximum), and '*pertelekomunikasian*' /pər.tɛ.ɫɛ.kɔ.mu.ni.kɑ.si.ɑn/ (related to telecommunication). These facts are extremely different from English that has more than 80% monosyllable words [19]. For example, all words in the English sentence 'Please come to my home' are monosyllabic. Translating that sentence into the Indonesian gives '*Silakan datang ke rumahku*', where only the word '*ke*' is monosyllabic, the others are disyllabic and trisyllabic.

A data-driven method called PNNR, a variant of k-nearest neighbour classification rule (kNN), gives low phoneme error rate and capability of disambiguating homographs for the Indonesian grapheme-to-phoneme (G2P) conversion [20]. Here PNNR is used to develop a new syllabification model sequentially integrated after a G2P model. It receives a phoneme sequence and produces its syllabification points. In this new model, a four-feature phoneme encoding and a phonotactic-based pre-syllabification procedure are proposed. This model will be cross validated using five datasets generated from 50K Indonesian words.

2. Research Method

An automatic syllabification is commonly applied to a word (called hyphenation or orthographic syllabification) rather than a phoneme sequence (called phonemic syllabification). In English, orthographic syllabification is useful to improve the accuracy of G2P. Bartlett in [21] proved that the information of

Table 1: Frequency of syllable structures in the Indonesian language

Number	Syllable structure	Frequency	Percentage
1	V	6,606	4.08
2	CV	82,061	50.63
3	CCV	3,056	1.89
4	CCCV	44	0.03
5	VC	6,338	3.91
6	CVC	61,826	38.15
7	VCC	116	0.07
8	CVCC	252	0.16
9	CVCCC	6	0.00
10	CCVC	1,639	1.01
11	CCVCC	72	0.04
12	CCCVC	56	0.03

orthographic syllabification improves the accuracy of English G2P conversion. However, this fact can not be generalized to other languages.

The Indonesian language has some different characteristics compared to English. It has 29 affixes: 7 prefixes, 4 infixes, and 18 suffixes [22]. A prefix or an infix can be used individually or simultaneously with some suffixes to produce derivatives. Two prefixes, with a certain priority order, may be used simultaneously to build a derivative. Therefore, many derivatives can be derived from a root, as in table 2. These facts make the Indonesian has some ambiguous orthographic syllabification for some similar words, e.g. a root '*teror*' (terror) is syllabified into ⟨*te.ror*⟩ whereas a derivative '*terorak*', derived from '*orak*' (unravel), is syllabified into ⟨*ter.o.rak*⟩. This ambiguity can be solved if both words are converted into phoneme sequences first, where they will be syllabified into /tɛ.rɔr/ and /tər.ɔ.rɑk/ respectively. Syllabification and hyphenation in the Indonesian can be different for most derivatives [23]. For example, a root '*absah*' (valid) is syllabified into /ɑb.sɑh/ and hyphenated into ⟨*ab.sah*⟩, but its derivative '*keabsahan*' (validity) is syllabified into /kə.ɑb.sɑ.hɑn/ and hyphenated into ⟨*ke.ab.sah.an*⟩, where the grapheme ⟨h⟩ is in the syllable ⟨sah⟩ not in ⟨han⟩. Such case is called inside word resyllabification. The Indonesian does not have crossword resyllabification. A grapheme sequence in Indonesian can be converted into some ambiguous phonemes. For example, a grapheme sequence ⟨ng⟩ can be converted into a single phoneme /ŋ/, such as '*bunga*' (flower) that is phonemicized into /buŋɑ/, or two phonemes, /n/ and /g/, such as '*astringen*' (astringent) that is phonemicized into /ɑstringən/. Hence, syllabifying the phoneme sequences is easier than the graphemes since the ambiguity of grapheme sequences has been solved by converting them into single phonemic symbols (SPS). Therefore, it is better to perform G2P before syllabification since G2P ambiguity is easier solved at the word level than at the syllable level. The Indonesian has simple rules for G2P. According to [20], an Indonesian G2P can produce low phoneme error rate, around 1.07%.

Based on the above characteristics, the Indonesian syllabification is designed to be a phonemic syllabification as illustrated by figure 1 that consists of two subprocesses. But, here the G2P is excluded from the syllabification system in order to focus the discussion on phonemic syllabification. In the figure 1, a phoneme sequence is firstly parsed to define syllabification points based on phonotactic constraints as described in [22] and [23] that is generally applied to all Indonesian words without exception. For examples, the consecutive phonemes /mp/ in /empati/ and /kt/ in /strukturisasi/ should be split since there is no Indonesian syllable containing /mp/ nor /kt/. Secondly, a PNNR will find the remaining syllabification points. Since the Indonesian syllable should contain a vowel (nucleus) that can be preceded by one or more consonants (onset) and followed by one or more consonants (coda) [22], the syllabification points should be between two vowels. Hence, the missing syllabification point in /em.pati/ can be either between /a/ and /t/ or between /t/ and /i/.

Data preprocessing. Defining syllabification points in a phoneme sequence contextually depends on surrounding phonemes. The number of surrounding phonemes,

Table 2: Examples of the usage of Indonesian affixes

Root	Affixes	Derivative
beli (buy)	*meng-*	*membeli* (buy)
	meng-kan	*membelikan* (buy for)
	per-	*pembeli* (buyer)
	per-an	*pembelian* (purchasing)
	ber-an	*berbelian* (go shopping)
	ter-	*terbeli* (not deliberately bought)
	di-	*dibeli* (deliberately bought)
	di-kan	*dibelikan* (bought by someone)
	-kan	*belikan* (please buy)
	-an	*belian* (purchasing)
henti (stop)	*meng-kan*	*menghentikan* (to stop)
	meng-ber-kan	*memberhentikan* (to stop)
	per-an	*perhentian* (stopping point)
	per-ber-an	*pemberhentian* (stopping point)
	ber-	*berhenti* (to sop)
	ter-	*terhenti* (not deliberately stopped)
	di-kan	*dihentikan* (deliberately stopped)
	-kan	*hentikan* (please stop)

also known as contextual length L, varies depending on the language. Since the Indonesian has 7.64 phonemes per word, L is set to be 8 or more. Data preprocessing is started by converting a phoneme sequence into some patterns, as illustrated by figure 2, where '*' is a blank symbol (no phoneme), '|' is a syllabification boundary (B), L_1 and R_1 are the first phonemic context on the left and the right respectively, Class = 1 is a syllabification point, and Class = 0 is a not syllabification point. In the figure 2, using $L = 8$ the phoneme sequence /buŋɑ/ (flower) is converted into three patterns: two in class 0 and one in class 1. Next, all patterns are grouped into two classes. A trainset of 40K words produces 118K unique patterns in class 0 (63%) and 69K in class 1 (37%).

Four-feature phoneme encoding. The best encoding for neural network-based hyphenation and syllabification is orthogonal binary code [9], [10]. But, this encoding produces high SER since it sees graphemes or phonemes equally as independent inputs with same distances (has two different bits) without considering them contextually in a word. Therefore, in this research a four-feature encoding {consonant/vowel, manner of articulation, place of articulation, voiced/unvoiced} is proposed by considering the categorization of Indonesian phonemes in [22]. The four-feature codes for 38 Indonesian phonemes and three additional non-phonemic symbols (*, -, and space) are listed in table 3. The distance between two phonemes is defined as the number of different features. This encoding produces a small distance for two phonemes with similar features, such as two vowels or two similar consonants such as /b/ and /p/. This is motivated by some common cases in the Indonesian. For examples, in words '*sabda*' (word) and '*sapta*' (seven), the syllabification points are between those consonant sequences. Thus, /b/ and /p/ as well as /d/ and /t/ have very small distance. As an example, phoneme /b/ is encoded into CPBU {Consonant, Plosive, Bilabial, Unvoiced}.

Phonemic contextual weight. In [20], an exponentially decaying contextual weight function is used for the Indonesian G2P which approaches the trend of the information gain (IG). In this research a similar phonemic contextual weight for syllabification is used as formulated by equation 1, where w_i is the weight for the i-th contextual phoneme, p is an exponential constant, and L is the phonemic contextual length distributed equally into left and right of the boundary. Thus, the first contextual phoneme has the maximum weight since it is the most

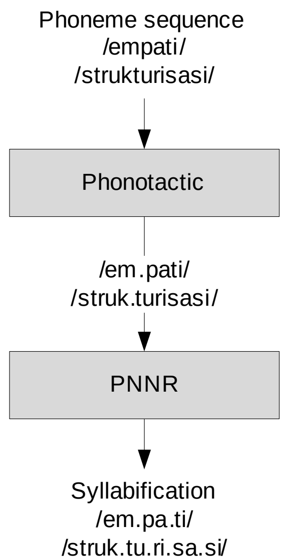

Figure 1: Design of syllabification using phonotactic knowledge and PNNR

$L_4\,L_3\,L_2\,L_1\,B\,R_1\,R_2\,R_3\,R_4$	Class
***b|uŋa*	0
bu|ŋa	1
*buŋ|a***	0

Figure 2: Converting a phoneme sequence into some patterns using $L = 8$

Table 3: Encoding for 38 single phonemic symbols and 3 non-phonemic symbols using the symbol set [{Vowel, Consonant}, {Low, Mid, High, Plosive, Affricative, Fricative, Nasal, Thrill, lateRal, Semivowel}, {Front, Central, bacK, Bilabial, Labiodental, Dental, Palatal, Velar, Glottal}, {Voiced, Unvoiced}]

Number	SPS	IPA	Code	Number	SPS	IPA	Code
1	a	ɑ	VLCV	22	g	g	CPVV
2	e	ɛ	VMFV	23	c	ʧ	CAPU
3	E	ə	VMFV	24	j	ʤ	CAPV
4	i	i	VHFV	25	f	f	CFTU
5	o	ɔ	VMKV	26	s	s	CFDU
6	u	u	VHKV	27	z	z	CFDV
7	A	aɪ	VMFV	28	m	m	CNBV
8	U	aʊ	VMKV	29	n	n	CNDV
9	Y	eɪ	VMFV	30	h	h	CFGU
10	O	ɔɪ	VHCV	31	r	r	CTDV
11	1	ɑ + ʔ	VLCV	32	l	ɫ	CRDV
12	2	ɛ + ʔ	VMFV	33	w	w	CSBV
13	3	ə + ʔ	VMFV	34	y	j	CSPV
14	4	i + ʔ	VHFV	35	K	x	CFVU
15	5	ɔ + ʔ	VMKV	36	G	ŋ	CNVV
16	6	u + ʔ	VHKV	37	N	ɲ	CNPV
17	b	b	CPBU	38	S	ʃ	CFPU
18	p	p	CPBV	39	*		****
19	t	t	CPDU	40	-		****
20	d	d	CPDV	41	space		****
21	k	k	CPVU				

important phoneme in deciding syllabification boundary, whereas the last one has the minimum.

$$w_i = p^{L/2-i+1} \tag{1}$$

PNNR-based syllabification. PNNR for syllabification needs to decide between two classes: syllabification boundary or not, and works by finding the minimum probabilistic nearest neighbour distance between the current pattern and both classes. Neighbourhood weight for the j-th neighbour, u_j, in equation 2 where c is an exponential constant, is used in the same way as in Indonesian G2P [20].

$$u_j = \frac{1}{j^c} \tag{2}$$

Total distance between the current pattern and a class taking into account the k closest patterns is calculated using equation 3, where u_j is the weight for the j-th neighbour, L is the contextual length, and d_{li} and d_{ri} are the distances of the i-th contextual phoneme on the left and right calculated using the four-feature phoneme encoding.

$$T = \sum_{j=1}^{k} u_j \sum_{i=1}^{L/2} (d_{li}w_i + d_{ri}w_i) \tag{3}$$

In the figure 3 a phoneme sequence /em.pati/ is converted into two patterns, (a) and (b), that correspond to the two possible syllabification boundaries in /pati/. Using $k = 3$, $L = 8$, $p = 2.0$, and $c = 1.0$, the first pattern is classified as syllabification point, but the second one is not. Thus, /em.pati/ is syllabified into /em.pa.ti/.

The example in figure 3 is an illustration of the common case in which one of the syllabification points is unambiguously chosen. However the application of (3) only may be insufficient as it may suggest zero or multiple syllabification points. These problems can be solved simply by maximizing the ratio of the total distance of class 1 and class 0. For example, if the pattern empa|ti** produces total distance of class 0 = 3 and class 1 = 7 and the pattern mpat|i***

Table 4: Comparison of phoneme encoding

Phoneme encoding	Average SER	Average WER
Orthogonal binary code	0.93%	1.54%
Four-feature code	0.80%	1.32%

gives total distance of class 0 = 21 and class 1 = 29, then maximizing the ratio of total distance of class 1 and class 0 shows that the pattern empa|ti** is the winner and thus the phoneme sequence /em.pati/ is syllabified into /em.pa.ti/.

3. Result and Discussion

The dataset used in this research is a set of 50k words with corresponding phoneme sequences and their syllabification points. First, the dataset is randomly split into five subsets of 10K different words each. In a five fold cross validation 40k words are used for parameter tuning and 10k for evaluation.

Phoneme encoding. Firstly the PNNR (without phonotactic knowledge) is evaluated to see the performance of the four-feature encoding. Here PNNR is tuned using some prospective values of parameters, i.e. k is set to 5, c (constant of ranking power) is set to 1.0, p (power of contextual weight) is set to 2.0 in order to ensure that phonemes closer to the syllable boundary are much more important than the further ones, and L (contextual length) is set to 8 since the dataset 50K shows 7.64 phonemes per word in average. Testing to five datasets shows that the four-feature encoding produces lower average syllable error rate (SER) as well as word error rate (WER) when compared to the orthogonal binary encoding as listed in table 4. It gives SER of 0.80%, significantly lower than the orthogonal binary encoding that produces 0.93%. It relatively reduces the SER by 13.90%. This result proofs that the proposed encoding, which produces shorter distances for patterns containing phonemes with some similar features, makes the PNNR capable of clustering the patterns more accurately.

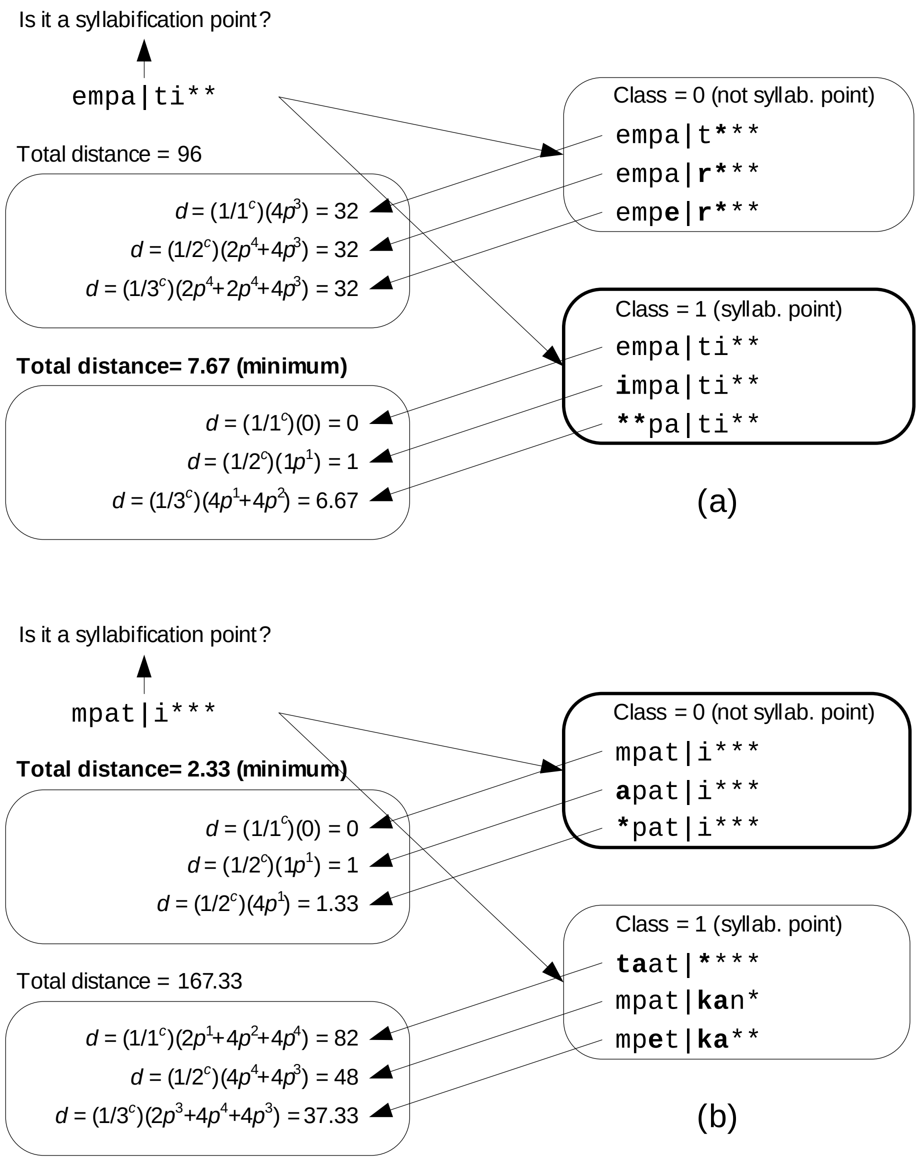

Figure 3: PNNR-based syllabification, using $k = 3$, $L = 8$, $p = 2.0$, and $c = 1.0$, syllabifies a phoneme sequence /em.pati/ into /em.pa.ti/

Table 5: Comparison of phonotactic knowledge

PNNR-based syllabification	Average SER	Average WER
PNNR without phonotactic knowledge	0.80%	1.32%
PNNR with phonotactic knowledge	0.66%	1.11%

Phonotactic knowledge. Secondly, PNNR with phonotactic knowledge is evaluated using five datasets. Both the PNNR without and with phonotactic knowledge use the same values of parameters, i.e. $k = 5$, $c = 1.0$, $p = 2.0$, and $L = 8$. The result in table 5 shows that PNNR with phonotactic knowledge produces average SER of 0.66%, lower than the PNNR without phonotactic knowledge that gives 0.80%. It relatively reduces the SER by 17.17%.

Based on both results above, the PNNR is designed to use both four-feature encoding and phonotactic knowledge. Next, the paramaters of PNNR are sequentially tuned on five datasets from the hardest (no knowledge to predict) to the easiest.

Neighbourhood size k. The number of neighbour, also called neighbourhood size, k in the PNNR is so varying based on the problem that it is difficult to be predicted. Hence, the PNNR is evaluated for varying k with $c = 1.0$, $p = 2.0$, and $L = 8$. The result, as illustrated by figure 4, shows that when $k = 1$ PNNR produces the highest SER since considering only one neighbour can lead it to be a too general clustering. It also gives high SER when considers so many neighbours that make it be a too specific clustering. It produces the lowest SER on $k = 3$.

Power constant for neighbourhood weight c. Next, the PNNR with $k = 3$, $p = 2.0$, and $L = 8$ is evaluated using varying c. The result, as illustrated by figure 5, shows that when c is less than 1.0 the PNNR yields high SER since the closer neighbour has quite similar distance to the further one. It also produces high SER when c is 1.4 or more since the closer neighbour has too high distance and the further is too low. It produces the lowest SER when $c = 1.3$.

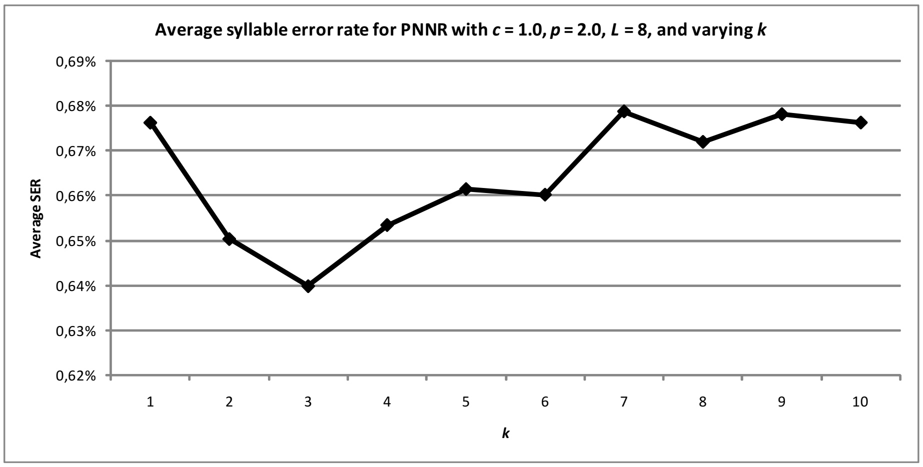

Figure 4: Performance of PNNR-based syllabification for varying k

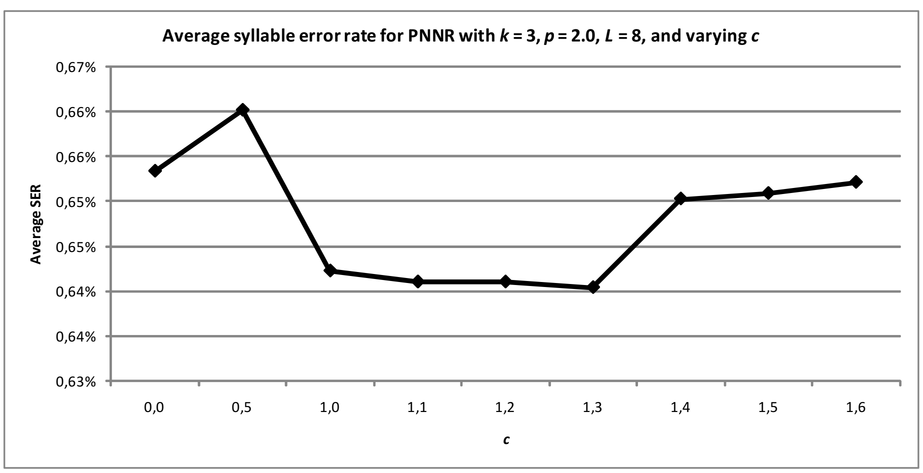

Figure 5: Performance of PNNR-based syllabification for varying c

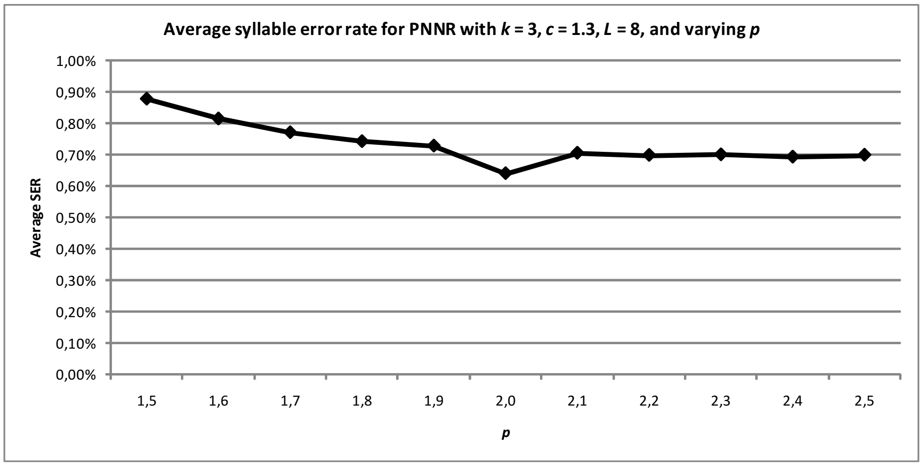

Figure 6: Performance of PNNR-based syllabification for varying p

Exponential constant for contextual weight p. The PNNR with $k = 3$, $c = 1.3$, and $L = 8$ is then evaluated using varying p. The result, as illustrated by figure 6, shows that when p is so small, less than 2.0, the PNNR yields high SER because the importance of closer contextual phonemes is quite similar to the further ones. It gives the lowest SER of 0.64% when $p = 2.0$.

Contextual length L. The PNNR with $k = 3$, $c = 1.3$, and $p = 2.0$ is then evaluated using varying L. The result, as illustrated by figure 7, shows that when L is 6 or less, the PNNR yields high SER since considering few contextual phonemes will lead to many ambiguous syllabification patterns. It gives stable low SER when L is 8 or more.

Syllabification errors. The most syllabification errors, about 60%, come from some derivatives with three prefixes /ber/, /per/, and /ter/, where the PNNR can not distinguish them to the roots beginning with those such phoneme sequences. For example, /beragam/ (diverse) is syllabified into /be.ra.gam/ but /beragama/ (religious) should be split into /ber.a.ga.ma/. The second errors, around 20%, are from some compound words, such as /anorganik/ (inorganic) that should be syllabified into /an.or.ga.nik/ but the PNNR produces

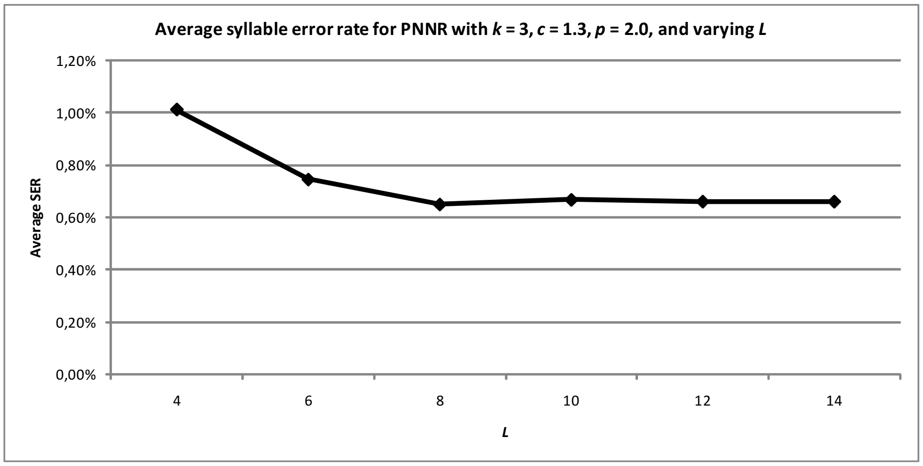

Figure 7: Performance of PNNR-based syllabification for varying L

/a.nor.ga.nik/. The others come from some roots and derivatives with prefix 'meng' as well as suffixes 'an' and 'i'.

4. Conclusion

The proposed four-feature phoneme encoding significantly reduces the SER by 13.90% relatively to the commonly used orthogonal binary encoding. The phonotactic-based pre-syllabification reduces the SER up to 17.17% relatively to the PNNR without pre-syllabification. Five fold cross validating proofs that the PNNR-based syllabification is stable by producing average SER of 0.64%. The most errors come from some derivatives with three prefixes /ber/, /per/, and /ter/ as well as some compound words. As a data-driven method, the PNNR can be applied to other languages, but the four-feature phoneme encoding and the phonotactic-based pre-syllabification should be slightly modified based on the phoneme categorization and the phonotactic constraints in those languages.

Acknowledgment

We would like to thank Telkom RDC and *Balai Pustaka* for the text corpus as well as the directorate general of higher eduction (DIKTI) for funding

the sandwich-like program in Departement Elektrotechniek-ESAT, KU Leuven, Belgium.

References

[1] K. Rogova, K. Demuynck, D. Van Compernolle, Automatic syllabification using segmental conditional random fields, Computational Linguistics in the Netherlands Journal 3 (2013) 34–48.

[2] R. Weerasinghe, A. Wasala, K. Gamage, A rule based syllabification algorithm for Sinhala, in: Proceedings of the second International Joint Conference on Natural Language Processing, 2005, pp. 438–449. doi:10.1007/11562214_39.

[3] Z. Hernández-Figueroa, F. J. Carreras-Riudavets, G. Rodríguez-Rodríguez, Automatic syllabification for Spanish using lemmatization and derivation to solve the prefix's prominence issue, Expert Systems with Applications 40 (2013) 7122–7131. doi:10.1016/j.eswa.2013.06.056.

[4] M. Hafiz, K. Rabiah, A, A. Azreen, A. M Taufik, Syllabification algorithm based on syllable rules matching for Malay language, in: Proceedings of The 10th WSEAS International Conference on Applied Computer and Applied Computational Science, 2011, pp. 279–286.

[5] W. Daelemans, A. van den Bosch, T. Weijters, IGTree: Using trees for compression and classification in lazy learning algorithms, Artificial Intelligence Review 11 (1-5) (1997) 407–423. doi:10.1.1.29.4517.

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

[7] K. Müller, Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpora training, in: Proceedings of the

39th Annual Meeting on Association for Computational Linguistics, ACL, 2001, pp. 410–417.

[8] W. Daelemans, A. van den Bosch, A neural network for hyphenation, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN92), Brighton, United Kingdom, 1992, pp. 1647–1650 (vol. 2). doi:10.1016/B978-0-444-89488-5.50176-7.

[9] T. Kristensen, A neural network approach to hyphenating Norwegian, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IEEE, 2000, pp. 148–153 vol.2. doi:10.1109/IJCNN.2000.857889.

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

[11] K. Müller, Improving syllabification models with phonotactic knowledge, in: Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology - SIGPHON '06, 2006, pp. 11–20. doi:10.3115/1622165.1622167.

[12] H. Schmid, B. Möbius, J. Weidenkaff, Tagging syllable boundaries with joint n-gram models, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 1 (1) (2007) 49–52.

[13] S. Bartlett, G. Kondrak, C. Cherry, On the syllabification of phonemes, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 308–316. doi:10.3115/1620754.1620799.

[14] C. R. Adsett, Y. Marchand, A comparison of data-driven automatic syllabification methods, in: Proceedings of the 16th International Symposium

on String Processing and Information Retrieval (SPIRE), Springer Berlin Heidelberg, 2009, pp. 174–181. doi:10.1007/978-3-642-03784-9_17.

[15] T. Mayer, Toward a totally unsupervised, language-independent method for the syllabification of written texts, in: Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, no. July, 2010, pp. 63–71.

[16] Y. Marchand, C. R. Adsett, R. I. Damper, Evaluating automatic syllabification algorithms for English, in: Proceedings of the 6th International Speech Communication Association ISCA Workshop on Speech Synthesis, 2007, pp. 316–321.

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

[18] R. M. Dauer, Stress-timing and syllable-timing reanalyzed, Journal of Phonetics 11 (1) (1983) 51–62.

[19] S.-L. W. S.-L. Wu, M. Shire, S. Greenberg, N. Morgan, Integrating syllable boundary information into speech recognition, in: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 1997, pp. 987–990. doi:10.1109/ICASSP.1997.596105.

[20] Suyanto, A. Harjoko, Nearest neighbour-based Indonesian G2P conversion, Telkomnika (Telecommunication, Computing, Electronics, and Control) 12 (2) (2014) 389–396.

[21] S. Bartlett, G. Kondrak, C. Cherry, Automatic syllabification with structured SVMs for letter-to-phoneme conversion., in: Proceedings of Human Language Technologies: The 2008 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 568–576.

[22] H. Alwi, S. Dardjowidjojo, H. Lapoliwa, A. M. Moeliono, Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3rd Edition, Balai Pustaka, Jakarta, 1998.

[23] A. Chaer, Fonologi Bahasa Indonesia (Indonesian Phonology), Rineka Cipta, Jakarta, 2009.

January 06, 2016

Dear Bernd Möbius,

We wish to submit a new manuscript entitled "Indonesian Syllabification Using Pseudo Nearest Neighbor Rule and Phonotactic Knowledge" for consideration by the Speech Communication, Elsevier.

We confirm that this work is original and has not been submitted/published earlier in any journal and is not being considered for publication elsewhere. All authors have seen and approved the manuscript and have contributed significantly for the paper.

In this paper, we report on phonemic syllabification for the Indonesian language using a simple data-driven method exploiting a new four-feature phoneme encoding and a phonotactic-based pre-syllabication. Five fold cross validating proofs that the new encoding significantly reduces the syllable error rate (SER) by 13.90% relatively to the commonly used orthogonal binary encoding and the method is stable by producing very low SER up to 0.64%. This paper should be of interest to readers in the areas of phonetics and phonology.

Thank you for your consideration of this manuscript. Please address all correspondence concerning this manuscript to me at suyanto.s3.ilkomp@mail.ugm.ac.id.

Sincerely,

Suyanto
Gadjah Mada University
Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia

Evidences of correspondences

Indonesian Syllabification Using
Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. **LoA with Major Revision (22 September 2016)**
3. Respond to Reviewers (11 October 2019)
4. Second Submission (11 October 2019)
5. LoA with Minor Revision (21 October 2019)
6. Final Submission (24 October 2019)
7. LoA with Fully Accepted (28 October 2019)

Ref: SPECOM_2016_8
Title: Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge
Journal: Speech Communication

Dear Mr. Suyanto,

Thank you for submitting your manuscript to Speech Communication. I have completed the review of your manuscript and a summary is appended below. The reviewers recommend reconsideration of your paper following major revision. I invite you to resubmit your manuscript after addressing all reviewer comments.

When resubmitting your manuscript, please carefully consider all issues mentioned in the reviewers' comments, outline every change made point by point, and provide suitable rebuttals for any comments not addressed.

To submit your revised manuscript:

- Log into EVISE® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=SPECOM
- Locate your manuscript under the header 'My Submissions that need Revisions' on your 'My Author Tasks' view
- Click on 'Agree to Revise'
- Make the required edits
- Click on 'Complete Submission' to approve

What happens next?

After you approve your submission preview you will receive a notification that the submission is complete. To track the status of your paper throughout the editorial process, log in to Evise® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=SPECOM.

Enrich your article to present your research with maximum impact. This journal supports the following Content Innovations:

- Interactive Plots: Interactive plot viewer providing easy access to the data behind plots. Please prepare a .CSV file with your plot data and test it online here before submitting as supplementary material.

I look forward to receiving your revised manuscript as soon as possible.

Kind regards,

Paul Foulkes
Subject Editor
Speech Communication

Comments from the editors and reviewers:
-**Reviewer 1**

-

This work aims to tackle the problem of automatic syllabification for Indonesian language. The paper is pleasant to read. However, I have 2 major concerns:

1. It is difficult to know if the proposed method is performing well due to the fact that there is no direct comparison with other studies/methods. A benchmark should have not been too difficult to conduct with the existence of numerous techniques that try to deal with the same task (i.e. look-up table, SVM approach, IB1-IG from Daelemans et al [1997], etc...). I believe that this type of methodology (i.e. benchmark) is mandatory in the field of speech technology and machine learning to assess the performance/relevance of algorithm(s) under investigation. The authors could compare their method with simple methods to reimplement (i.e look-up table and IB1-IG) or - even better - with the best current methods in the field.

2. The method that is proposed depends on several parameters. Unfortunately, no justification and/or reference are provided concerning the values that were assigned to these parameters (see Phoneme encoding section: "Here PNNR is tuned using some prospective values of parameters, i.e. ..."). It seems that only the contextual length (L) appears to have a justification for the set up. Nevertheless, this justification is not really convincing because it is only based on the fact that there are 7.64 phonemes per word in average. However, average can be misleading in the case where there is a lot of variance in the data. If the method is fast to run, why not using other values for k? Along the same line of thought, I was wondering if the authors run other settings (i.e. others than varying k) to see if their method is robust as well as to discover an optimized combination of values for the parameters.

I do believe that these 2 main points should be addressed for this paper to be accepted for publication.

Minor points/typos:

Abstract:

"with trainset of 40K words and testset of 10K words" -> "with a training set of 40K words and a test set of
10K words"

"shows the proposed" -> "shows that the proposed"

"Five fold cross validating proofs that the proposed..." -> "Five fold cross validating proves that the proposed..."

Footnote 1: the title (i.e. lecturer) should me removed. Also the affiliation "School of Computing, Telkom University (former: Telkom Institute of Technology), Bandung, West Java, 40257 Indonesia" should be located under the names of the authors.

Line 20: "Adsett and Marchand in [14] proved..." -> "Adsett and Marchand proved..."

Line 26: "proposed by Dauer in [18], the..." -> "proposed by Dauer [18], the..."

Lines 28/29: "more open syllables than closed": define the notions of open and closed syllables.

Line 31: Pusat Bahasa: who is she/he? Information should be provided about this person.

Lines 38/39: "It even has some very long words containing seven or more syllables", it would be helpful to provide the distribution of the number of syllables per word for the dataset of 50K Indonesian words that you are using.

Lines 48/49/50: "capability of disambiguating homographs for the Indonesian grapheme-to-phoneme (G2P)conversion": this sentence is not clear. Please rephrase it and give an example.

Lines 56/57: "(called hyphenation or orthographic syllabification)" -> "(called orthographic syllabification)", hyphenation is not the same as orthographic syllabification.

Lines 59/60: "Bartlett in [21] proved that the information of orthographic syllabification improves the accuracy of English G2P conversion." -> "Bartlett proved that the information of orthographic syllabification improves the accuracy of English G2P conversion [21]."

Lines 67/68: "has some ambiguous orthographic syllabification" -> "has some ambiguous orthographic syllabifications"

Line 78: "crossword resyllabification": please explain.

Line 116: "is orthogonal binary code [9], [10]" -> "is orthogonal binary code ([9], [10])" + give an example.

Table 3: Define the acronym IPA. IPA symbols with a + should be explained.

Equation 1 after line 140: the equation is not well motivated. Give more detail about it.

-**Reviewer 2**

- This paper addresses an interesting topic, the automatic syllabification of Indonesian, using what seems to be an original approach - pseudo nearest neighbour rule (PNNR) and phonotactic knowledge - which, it is claimed, significantly reduces the syllable error rate as compared to other approaches.

In my opinion the paper is not yet ready for publication and the authors need to address the following points before the paper can be published.

- The authors assume that there is a correct syllabification for every word - this is seen in the fact that they frequently make statements such as:

"a root 'terror' (terror) is syllabified into (te.ror) whereas a derivative 'terorak', derived from 'orak' (unravel) is syllabified into (ter.o.rak)."

No reference is given as to what authority has been used to establish the "correct" syllabification. This question needs to be addressed since in a language like English, for example, a simple word like 'city' has been claimed by some to be syllabified as [ci][ty], by others as [cit][y] and by yet others as [ci[t]y] where the medial consonant is analysed as belonging to both the first and the second syllable (ambisyllabicity). Still other phonologists have even argued that the syllable is not an appropriate unit for phonological representations at all. Without getting into detailed phonological arguments, the authors should at least explain on what authority they base their analysis, otherwise the whole issue of syllable error rate is meaningless.

- The PNNR is described as being based on an earlier study by two of the same authors but the rule itself is described extremely briefly. Since it is quite possible that readers of the article may not have access to the earlier study it would be appropriate to expand the explanation with detailed examples.

- The English of the article needs revision by a native speaker - there are a large number of errors, particularly in the use of articles ("the Indonesian", "it has majority of CV syllables", etc.).

Evidences of correspondences

Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. LoA with Major Revision (22 September 2016)
3. **Respond to Reviewers (11 October 2019)**
4. Second Submission (11 October 2019)
5. LoA with Minor Revision (21 October 2019)
6. Final Submission (24 October 2019)
7. LoA with Fully Accepted (28 October 2019)

Comments from the editors and reviewers:
-Reviewer 1

This work aims to tackle the problem of automatic syllabification for Indonesian language. The paper is pleasant to read. However, I have 2 major concerns:

1. It is difficult to know if the proposed method is performing well due to the fact that there is no direct comparison with other studies/methods. A benchmark should have not been too difficult to conduct with the existence of numerous techniques that try to deal with the same task (i.e. look-up table, SVM approach, IB1-IG from Daelemans et al [1997], etc...). I believe that this type of methodology (i.e. benchmark) is mandatory in the field of speech technology and machine learning to assess the performance/relevance of algorithm(s) under investigation. The authors could compare their method with simple methods to reimplement (i.e look-up table and IB1-IG) or - even better - with the best current methods in the field.

We did a benchmark to look-up procedure (LUP) as decribed by Marchand 2009. The concept of LUP is explained in Section 2 and the benchmark result is listed in Table 6.

2. The method that is proposed depends on several parameters. Unfortunately, no justification and/or reference are provided concerning the values that were assigned to these parameters (see Phoneme encoding section: "Here PNNR is tuned using some prospective values of parameters, i.e. ..."). It seems that only the contextual length (L) appears to have a justification for the set up. Nevertheless, this justification is not really convincing because it is only based on the fact that there are 7.64 phonemes per word in average. However, average can be misleading in the case where there is a lot of variance in the data. If the method is fast to run, why not using other values for k? Along the same line of thought, I was wondering if the authors run other settings (i.e. others than varying k) to see if their method is robust as well as to discover an optimized combination of values for the parameters.

Each parameter of PNNR is explained more detail in Section 2.

The following sentence is added on line 248:

“For computational reasons each parameter is sequentially tuned from the hardest (no knowledge to predict) to the easiest using the described five fold cross validation procedure.”

I do believe that these 2 main points should be addressed for this paper to be accepted for publication.

Minor points/typos:

Abstract:

"with trainset of 40K words and testset of 10K words" -> "with a training set of 40K words and a test set of 10K words"

"with trainset of 40K words and testset of 10K words" → "with a training set of 40K words and a test set of 10K words"

"shows the proposed" -> "shows that the proposed"

"shows the proposed" → "shows that the proposed"

"Five fold cross validating proofs that the proposed..." -> "Five fold cross validating proves that the proposed..."

"Five fold cross validating proofs that the proposed..." → " Five-fold cross-validating proves that the proposed..."

Footnote 1: the title (i.e. lecturer) should be removed. Also the affiliation "School of Computing, Telkom University (former: Telkom Institute of Technology), Bandung, West Java, 40257 Indonesia" should be located under the names of the authors.

Revised footnote:

*Corresponding author
Email addresses: suyanto@telkomuniversity.ac.id (Suyanto), shartati@ugm.ac.id (Sri Hartati), aharjoko@ugm.ac.id (Agus Harjoko), Dirk.VanCompernolle@esat.kuleuven.be (Dirk Van Compernolle)

Revised affiliation:

Suyanto[a,*], Sri Hartati[b], Agus Harjoko[b], Dirk Van Compernolle[c]

[a]*Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia. School of Computing, Telkom University, Bandung, West Java 40257, Indonesia*
[b]*Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia*
[c]*Departement Elektrotechniek-ESAT, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven,Belgium*

Line 20: "Adsett and Marchand in [14] proved..." -> "Adsett and Marchand proved..."

"Adsett and Marchand in [14] proved..." → "Adsett and Marchand proved..."

Line 26: "proposed by Dauer in [18], the..." -> "proposed by Dauer [18], the..."

"proposed by Dauer in [18], the..." → "proposed by Dauer [18], the..."

Lines 28/29: "more open syllables than closed": define the notions of open and closed syllables.

more open syllables (ended with a vowel) than the closed ones (ended with a consonant)

Line 31: Pusat Bahasa: who is she/he? Information should be provided about this person.

Tim Penyusun Kamus Pusat Bahasa, Kementerian Pendidikan dan Kebudayaan (The Lexicographer Team of Language Center, Indonesian Ministry of Education and Culture)

Lines 38/39: "It even has some very long words containing seven or more syllables", it would be helpful to provide the distribution of the number of syllables per word for the dataset of 50K Indonesian words that you are using.

The distributions of the number of syllables per word and the number of phonemes per word in the 50K words from KBBI are illustrated by figure 1 and 2.

Lines 48/49/50: "capability of disambiguating homographs for the Indonesian grapheme-to-phoneme (G2P) conversion": this sentence is not clear. Please rephrase it and give an example.

It also has a capability of disambiguating homographs, such as a word 'apel' where the grapheme <e> could be pronounced as either /\textipa{\textepsilon}/ like in 'apel pagi' (morning ceremony) or /\textipa{@}/ like in 'apel hijau' (green apple), simply by using a longer graphemic contextual length.

Lines 56/57: "(called hyphenation or orthographic syllabification)" -> "(called orthographic syllabification)", hyphenation is not the same as orthographic syllabification.

"(called hyphenation or orthographic syllabification)" → "(called orthographic syllabification)"

Lines 59/60: "Bartlett in [21] proved that the information of orthographic syllabification improves the accuracy of English G2P conversion." -> "Bartlett proved that the information of orthographic syllabification improves the accuracy of English G2P conversion [21]."

"Bartlett in [21] proved that the information of orthographic syllabification improves the accuracy of English G2P conversion." → "Bartlett proved that the information of orthographic syllabification improves the accuracy of English G2P conversion [21]."

Lines 67/68: "has some ambiguous orthographic syllabification" -> "has some ambiguous orthographic syllabifications"

"has some ambiguous orthographic syllabification" → "has some ambiguous orthographic syllabifications"

Line 78: "crossword resyllabification": please explain.

“Indonesian does not have crossword resyllabification.” is deleted.

Line 116: "is orthogonal binary code [9], [10]" -> "is orthogonal binary code ([9], [10])" + give an example.

An example is illustrated by figure 6.

Table 3: Define the acronym IPA. IPA symbols with a + should be explained.

The IPA is defined on the caption of Table 3:

Table 3: Four-feature codes for 38 Single Phonemic Symbols (SPS) with the corresponding **International Phonetic Alphabet** (IPA) and 3 non-phonemic symbols using the symbol set [Vowel, Consonantg, Low, Mid, High, Plosive, Affricative, Fricative, Nasal, Thrill, lateRal,Semivowelg, Front, Central, bacK, Bilabial, Labiodental, Dental, Palatal, Velar, Glottal, Voiced, Unvoiced]

The /a + ʔ/ is explained as follow:

There are six double phonemes, simbolized as /1/ to /6/, that contain a glottal, such as **/a + ʔ/ from a word '*saat*' (time) that is pronounced as /saʔat/**.

Equation 1 after line 140: the equation is not well motivated. Give more detail about it.

The Equation 1 is explained on line 147-155 and illustrated by figure 5.

-Reviewer 2

- This paper addresses an interesting topic, the automatic syllabification of Indonesian, using what seems to be an original approach - pseudo nearest neighbour rule (PNNR) and phonotactic knowledge - which, it is claimed, significantly reduces the syllable error rate as compared to other approaches.
In my opinion the paper is not yet ready for publication and the authors need to address the following points before the paper can be published.
- The authors assume that there is a correct syllabification for every word - this is seen in the fact that they frequently make statements such as:
"a root 'terror' (terror) is syllabified into (te.ror) whereas a derivative 'terorak', derived from 'orak' (unravel) is syllabified into (ter.o.rak)."

No reference is given as to what authority has been used to establish the "correct" syllabification. This question needs to be addressed since in a language like English, for example, a simple word like 'city' has been claimed by some to be syllabified as [ci][ty], by others as [cit][y] and by yet others as [ci[t]y] where the medial consonant is analysed as belonging to both the first and the second syllable (ambisyllabicity). Still other phonologists have even argued that the syllable is not an appropriate unit for phonological representations at all. Without getting into detailed phonological arguments, the authors should at least explain on what authority they base their analysis, otherwise the whole issue of syllable error rate is meaningless.

"You are correct that syllabification ambiguity exists and the degree of ambiguity is language dependent. For Indonesian there is no established syllabification. We relied on the Indonesian syllabification rule developed by Alwi et al. [20] as described in section 2, line 86-101.

- The PNNR is described as being based on an earlier study by two of the same authors but the rule itself is described extremely briefly. Since it is quite possible that readers of the article may not have access to the earlier study it would be appropriate to expand the explanation with detailed examples.

The PNNR is explained more detail in Section 2, line 204-218 as well as illustrated by Figure 7 and 8.

- The English of the article needs revision by a native speaker - there are a large number of errors, particularly in the use of articles ("the Indonesian", "it has majority of CV syllables", etc.).

"the Indonesian" → “Indonesian”

"it has majority of CV syllables" → “it has the most CV syllables”

Evidences of correspondences

Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. LoA with Major Revision (22 September 2016)
3. Respond to Reviewers (11 October 2019)
4. **Second Submission (11 October 2019)**
5. LoA with Minor Revision (21 October 2019)
6. Final Submission (24 October 2019)
7. LoA with Fully Accepted (28 October 2019)

Indonesian Syllabification Using a Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

Suyanto[a,*], Sri Hartati[b], Agus Harjoko[b], Dirk Van Compernolle[c]

[a] *Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia. School of Computing, Telkom University, Bandung, West Java 40257, Indonesia*
[b] *Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara, Bulaksumur, Yogyakarta 55281, Indonesia*
[c] *Departement Elektrotechniek-ESAT, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium*

Abstract

This paper discusses phonemic syllabification using a pseudo nearest neighbour rule (PNNR) and phonotactic knowledge for Indonesian language. The proposed data-driven model uses a four-feature phoneme encoding and a phonotactic-based pre-syllabification. Evaluating on five datasets with a training set of 40K words and a test set of 10K words each shows that the proposed encoding significantly reduces the average syllable error rate (SER) by 13.90% relatively to the commonly used orthogonal binary encoding and the pre-syllabification also reduces the average SER up to 17.17% relatively to the PNNR without pre-syllabification. Five-fold cross-validating proves that the proposed PNNR-based syllabification is stable by producing an average SER of 0.64%. Most errors come from derivatives with the prefixes 'ber', 'per', and 'ter' as well as from compound words. This result is also significantly lower than a Look-Up-based syllabification that gives an average SER of 2.60%.

Keywords: Indonesian syllabification, four-feature phoneme encoding, phonotactic knowledge, pseudo nearest neighbour rule

---

*Corresponding author

Email addresses: `suyanto@telkomuniversity.ac.id` (Suyanto), `shartati@ugm.ac.id` (Sri Hartati), `aharjoko@ugm.ac.id` (Agus Harjoko), `Dirk.VanCompernolle@esat.kuleuven.be` (Dirk Van Compernolle)

1. Introduction

There are two approaches to automatic syllabification: rule-based and data-driven. The rule-based approach, using the sonority sequencing principle, legality principle, and maximal onset principle, produces ambiguous syllabification that are only valid for some cases [1]. This approach gives high accuracy for languages with few simple and consistent syllabification rules, such as Sinhala [2] and Spanish [3]. But, it performs poorly for a slightly more complex syllabic language, such as Malay [4]. Hence, many data-driven methods have been developed, such as the IGTree learning algorithm [5], weighted finite-state transducers [6], combination of treebank and bracketed corpora training [7], neural network ([8], [9], [10]), probabilistic context-free grammars [11], joint n-gram models [12], combination of support vector machine and hidden Markov model [13], syllabification by analogy (SbA) [14], counting the actual syllables to determine the best split of word-medial consonant sequences [15], segmental conditional random fields [1].

The data-driven approaches give higher accuracy than the rule-based ones for English, a complex syllabic language [16]. They also perform better for a language with low syllabic complexity, such as Italian, where the SbA (one of data-driven methods) reaches a word accuracy of 97.70% but the best rule set (SYL-LABE) achieves only 89.77% word accuracy for 44K words [17]. Adsett and Marchand proved that data driven approaches generally produce higher accuracy for nine European languages [14], where SbA gives the highest average word accuracy of 96.84% (standard deviation of 2.93) whereas Liang's algorithm produces a mean of 95.67% (standard deviation of 5.70).

However, the SbA, which globally classifies a pattern by finding the best analogy (minimizing shortest path using three metrics) in the training set [18], may not suitable for a language with many exceptions and ambiguities such as Indonesian. For example, a phoneme sequence /bəri/ (syllabified as /bə.ri/) can take an analogy from /məmbəri/ (/məm.bə.ri/) or vise versa, where /məm/ is a prefix for /bəri/ (give). But, /beragam/ (/bə.ra.gam/) can not take an analogy from /bəragama/ (/bər.a.ga.ma/) where /bər/ is a prefix for /agama/ (religion) and /bəragam/ (/bə.ra.gam/) is a derivative from a prefix /bər/ and a root /ragam/ (various).

Based on the method of classifying languages proposed by Dauer [19], Indonesian is categorized as a simple syllabic language. It has mostly CV syllables (where C is a single consonant and V is a single vowel) and more open syllables (ended with a vowel) than closed ones (ended with a consonant). A study of a vocabulary of 50K words, collected from the great dictionary of Indonesian language (*Kamus Besar Bahasa Indonesia*, KBBI) created by *Tim Penyusun Kamus Pusat Bahasa, Kementerian Pendidikan dan Kebudayaan* (The Lexicographer Team of Language Center, Indonesian Ministry of Education and Culture) and syllabified based on the rules of Indonesian syllabification developed by Alwi et al. [20], shows that Indonesian language has 50.63% CV syllables and 56.63% open syllables, as listed in Table 1. In contrast, English (a complex syllabic language) has about 35% of CV syllables and a wider variety of both open and closed syllables [19]. However, Indonesian is a syllabically rich language. Observing the 50K words of KBBI shows that Indonesian has 3.20 syllables per word on average (standard deviation 0.41) and 7.54 phonemes per word on average (standard deviation 0.42). It has 98.30% polysyllabic words, much more than monosyllabics that only account for 1.70%. But, English has 80% polysyllabic words and 20% monosyllabics based on Wordsmyth dictionary [18]. The distributions of the number of syllables per word in 50K words from KBBI and 50K words from Wordsmyth dictionary are illustrated by Figure 1. Meanwhile, the distribution of the number of phonemes per word in the 50K words from KBBI is shown in Figure 2.

A data-driven local classifier called PNNR, a variant of k-nearest neighbour classification rule (kNN), gives a low phoneme error rate (PER) for an Indonesian grapheme-to-phoneme (G2P) conversion [21]. It also has a capability of disambiguating homographs, such as a word '*apel*' where the grapheme ⟨e⟩ could be pronounced as either /ɛ/ like in '*apel pagi*' (morning ceremony) or /ə/ like in '*apel hijau*' (green apple), simply by using a longer graphemic contextual

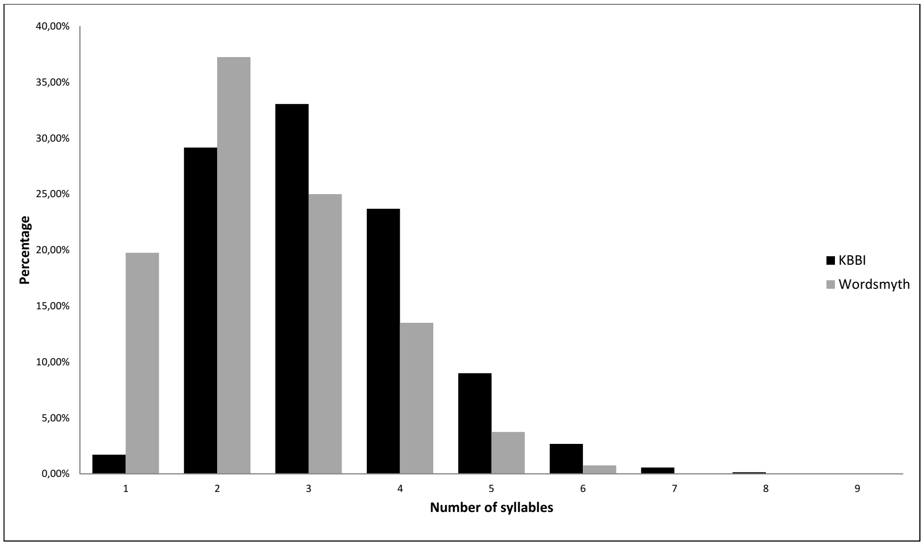

Figure 1: Distributions of number of syllables per word in the 50K words from KBBI and Wordsmyth dictionary

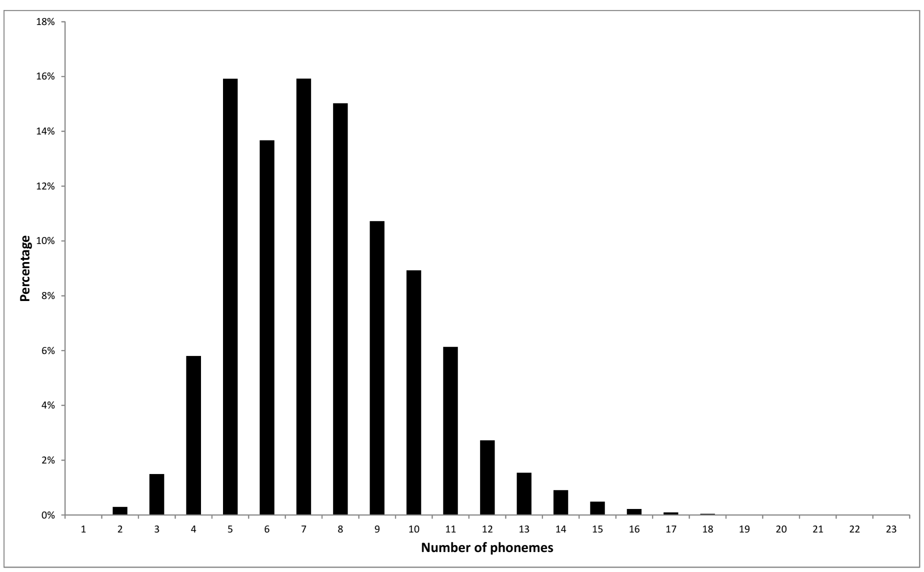

Figure 2: Distribution of number of phonemes per word in the 50K words from KBBI

Table 1: Syllable structures in Indonesian language

Number	Syllable structure	Frequency	Percentage (%)
1	V	6,606	4.08
2	CV	82,061	50.63
3	CCV	3,056	1.89
4	CCCV	44	0.03
5	VC	6,338	3.91
6	CVC	61,826	38.15
7	VCC	116	0.07
8	CVCC	252	0.16
9	CVCCC	6	0.00
10	CCVC	1,639	1.01
11	CCVCC	72	0.04
12	CCCVC	56	0.03

length. In this research, the PNNR is used to develop a new syllabification model for phoneme strings. It receives a phoneme sequence and produces its syllabification points. In this new model, a four-feature phoneme encoding and a phonotactic-based pre-syllabification procedure are proposed. This model will be compared to the Look-Up Procedure (LUP)-based syllabification model described by Marchand [18].

2. Research Method

An automatic syllabification is commonly applied to a word (called orthographic syllabification) rather than a phoneme sequence (called phonemic syllabification). In English, orthographic syllabification is useful to improve the accuracy of G2P. Bartlett proved that the information of orthographic syllabification improves the accuracy of English G2P conversion [22]. However, this fact can not be generalized to other languages.

Indonesian language has some different characteristics compared to English.

It has 29 affixes: 7 prefixes, 4 infixes, and 18 suffixes [20]. A prefix or an infix can be used individually or simultaneously with some suffixes to produce derivatives. Two prefixes, with a certain priority order, may be used simultaneously to build a derivative. Therefore, many derivatives can be derived from a root, as in Table 2. These facts make that Indonesian has some ambiguous orthographic syllabifications for certain similar words, e.g. a root '*teror*' (terror) is syllabified as ⟨***te.****ror*⟩ whereas a derivative '*terorak*', derived from '*orak*' (unravel), is syllabified as ⟨***ter.****o.rak*⟩. This ambiguity can be solved if both words are converted into phoneme sequences first, where they will be syllabified as /tɛ.rɔr/ and /tər.ɔ.rɑk/ respectively.

In this research, the rules of Indonesian syllabification developed by Alwi et al. [20] are used since they represent the major pronunciations in Indonesian. Syllabification and hyphenation in Indonesian language can be different for most derivatives ([20], [23]). For example, a root '*absah*' (valid) is syllabified as /ɑb.sɑh/ and hyphenated into ⟨*ab.sah*⟩, but its derivative '*keabsahan*' (validity) is syllabified as /kə.ɑb.sɑ.hɑn/ and hyphenated into ⟨*ke.ab.sah.an*⟩, where the grapheme ⟨h⟩ is in the syllable ⟨sah⟩ not in ⟨han⟩. Such case is called inside word resyllabification. A grapheme sequence in Indonesian can be converted into ambiguous phonemes. For example, a grapheme sequence ⟨ng⟩ can be converted into a single phoneme /ŋ/, such as '*bunga*' (flower) that is phonemized as /buŋɑ/, or two phonemes, /n/ and /g/, such as '*astringen*' (astringent) that is phonemized as /ɑstringən/. This example shows that syllabifying a phoneme sequence is easier than a grapheme sequence since the ambiguities of graphemes have been solved by converting them into Single Phonemic Symbols (SPS). Therefore, it is better to perform G2P before syllabification since the phonemic ambiguity is easier solved at the word level than the syllable level.

Based on the above characteristics, Indonesian syllabification is designed to be a phonemic syllabification, as illustrated by Figure 3, that consists of two subprocesses. In this paper the G2P is excluded from the syllabification system to focus the discussion on phonemic syllabification. In Figure 3, a phoneme sequence is first parsed to derive syllabification points based on phonotactic

Table 2: Examples of the usage of Indonesian affixes

Root	Affix	Derivative
beli (buy)	*meng-*	*membeli* (buy)
	meng-kan	*membelikan* (buy for)
	per-	*pembeli* (buyer)
	per-an	*pembelian* (purchasing)
	ber-an	*berbelian* (go shopping)
	ter-	*terbeli* (not deliberately bought)
	di-	*dibeli* (deliberately bought)
	di-kan	*dibelikan* (bought by someone)
	-kan	*belikan* (please buy)
	-an	*belian* (purchasing)
henti (stop)	*meng-kan*	*menghentikan* (to stop)
	meng-ber-kan	*memberhentikan* (to stop)
	per-an	*perhentian* (stopping point)
	per-ber-an	*pemberhentian* (stopping point)
	ber-	*berhenti* (to sop)
	ter-	*terhenti* (not deliberately stopped)
	di-kan	*dihentikan* (deliberately stopped)
	-kan	*hentikan* (please stop)

constraints as described in [20] and [23] that are applied to all Indonesian words without exception. For examples, the consecutive phonemes /mp/ in /empati/ and /kt/ in /strukturisasi/ should be split since there is no Indonesian syllable containing /mp/ nor /kt/. Secondly, a PNNR will find the remaining syllabification points. Since Indonesian syllables should contain a vowel (nucleus) that can be preceded by one or more consonants (onset) and followed by one or more consonants (coda) [20], the syllabification points should be between two vowels. Hence, the missing syllabification point in /em.pati/ can be either between /a/ and /t/ or between /t/ and /i/.

Data preprocessing. Defining a syllabification point in a phoneme sequence contextually depends on surrounding phonemes on the left and right. The number of surrounding phonemes, also known as contextual length L, varies depending on the language. For English, the optimum L is 7 (three left contextual phonemes, a focus phoneme, and three right contextual phonemes) as used by LUP-based syllabification [18]. In this research, a different scheme is used for Indonesian. It does not use the focus phoneme, but '|' instead, and assumes that the surrounding phonemes on the left and right have the same influence to decide a syllabification point. Since Indonesian has 7.54 phonemes per word on average the optimum L is assumed to be at least 8 (four contextual phonemes on the left and four on the right).

The difference between patterns used in LUP and PNNR is illustrated by Figure 4. In LUP, a phoneme sequence /buŋɑ/ (flower) is converted into three patterns as illustrated by Figure 4(a), where /*/ is 'no phoneme', L_i and R_i are the i-th contextual phonemes on the left and the right respectively, Class = 1 states that after the focus phoneme F is a syllabification point and Class = 0 is not a syllabification point. In PNNR, that phoneme sequence is also converted into three patterns as illustrated by Figure 4(b), where '|' is a candidate syllabification point or boundary (B).

In this research, all duplicate patterns used in PNNR are removed and then grouped into two classes: Class 0 and Class 1. A training set of 40K words produces 118K unique patterns in Class 0 (63%) and 69K in Class 1 (37%).

Phonemic contextual weight. In [21], an exponentially decaying contextual weight function is used for Indonesian G2P which estimates the trend of the information gain (IG). That contextual weight function is adapted in this research, as

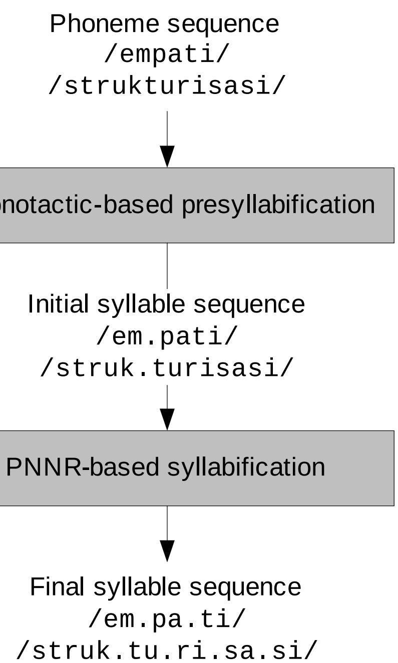

Figure 3: Design of syllabification using phonotactic knowledge and PNNR

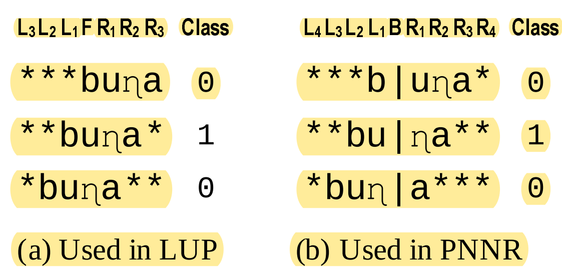

Figure 4: Conversion of a phoneme sequence /buŋa/ into three patterns using the LUP scheme with $L = 7$ (a) and the PNNR scheme with $L = 8$ (b)

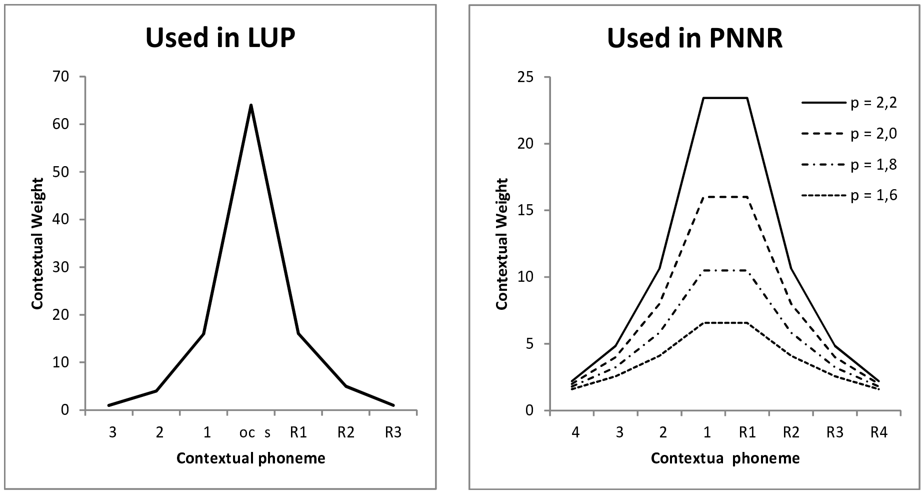

Figure 5: Difference between contextual weights used in PNNR and LUP version 10

formulated by Equation 1

$$w_i = p^{(L/2)-i+1} \tag{1}$$

where w_i is the weight for the i-th contextual phoneme, p is an exponential constant, and L is the phonemic contextual length distributed equally to the left and right of the boundary. Thus, the first contextual phoneme has the maximum weight since it is the most important phoneme for syllabification.

Based on the Equation (1), if $p = 1.6$ and $L = 8$, then the weight of the first contextual phoneme w_1 is $(1.6)^{(8/2)-1+1} = (1.6)^4 = 6.55$ and the total weight of the second to the fourth contextual phonemes w_{2-4} is $\sum_{i=2}^{4}(1.6)^{(8/2)-i+1} = 8.26$. It means the further contextual phonemes have enough influence. But, if $p = 4$, then $w_1 = 256$ and $w_{2-4} = 84$. It makes the first contextual phoneme fully dominant. The constant $p = 2$ produces a balance, where $w_1 = 16$ and $w_{2-4} = 14$. Hence, the optimum p is predicted to be around 2. This scheme is quite different to that used in LUP version 10 (with weight vector [1 4 16 64 16 5 1] described by Marchand [18]), as illustrated by Figure 5.

Four-feature phoneme encoding. The LUP simply uses two binary values to define a similarity between two patterns, where the same contextual phonemes give similarity of 1 and otherwise 0. This scheme may produce some confused similarities for intraclass and interclass patterns. For example, using LUP version 10 (with weight vector [1 4 16 64 16 5 1]), two interclass patterns /*paksa*/ (force) that syllabified as /pak.sa/ (syllable structure: CVC.CV) and /*pakaɪ**/ (use) that syllabified as /pa.kaɪ/ (CV.CV) give much higher similarity (instead of lower), 86, than two intraclass patterns /*paksa*/ and /*biŋkaɪ*/ (frame) that syllabified as /biŋ.kaɪ/ (CVC.CV), where produces similarity of 2. Hence, these patterns using weighted binary valued similarities will be hard to be used in developing a syllabification model.

The best encoding for neural network-based hyphenation and syllabification is orthogonal binary code (OBC) ([9], [10]). But, this encoding produces high SER since it sees graphemes or phonemes equally as independent inputs with same distances (has two different bits) without considering them contextually in a word. Therefore, in this research, a four-feature encoding {consonant/vowel, manner of articulation, place of articulation, voiced/unvoiced} is proposed by considering the categorization of Indonesian phonemes in [20]. The four-feature codes (FFC) for 38 Indonesian phonemes (symbolized using SPS with corresponding International Phonetic Alphabet (IPA)) and 3 additional non-phonemic symbols (*, -, and space) are listed in Table 3. There are six double phonemes, symbolized as /1/ to /6/, that contain a glottal, such as /ɑ + ʔ/ from a word '*saat*' (time) that is pronounced as /sɑʔɑt/.

In the four-feature encoding, the distance between two phonemes is defined as the number of different features. This encoding produces a small distance between phonemes with similar features, such as /b/ and /p/ or /d/ and /t/. It is motivated by some common cases in Indonesian. For examples, in two words '*sabda*' (word) and '*sapta*' (seven), the syllabification points are between those consonant sequences. Thus, /b/ and /p/ as well as /d/ and /t/ should have very small distance. By encoding /b/ into CPBU {Consonant, Plosive, Bilabial, Unvoiced} and /p/ into CPBV, then the distance between /b and /p/ is 1 since

Table 3: Four-feature codes for 38 Single Phonemic Symbols (SPS) with the corresponding International Phonetic Alphabet (IPA) and 3 non-phonemic symbols using the symbol set [{Vowel, Consonant}, {Low, Mid, High, Plosive, Affricative, Fricative, Nasal, Thrill, lateRal, Semivowel}, {Front, Central, backK, Bilabial, Labiodental, Dental, Palatal, Velar, Glottal}, {Voiced, Unvoiced}]

Number	SPS	IPA	Code	Number	SPS	IPA	Code
1	a	ɑ	VLCV	22	g	g	CPVV
2	e	ɛ	VMFV	23	c	ʧ	CAPU
3	E	ə	VMFV	24	j	ʤ	CAPV
4	i	i	VHFV	25	f	f	CFLU
5	o	ɔ	VMKV	26	s	s	CFDU
6	u	u	VHKV	27	z	z	CFDV
7	A	aɪ	VMFV	28	m	m	CNBV
8	U	aʊ	VMKV	29	n	n	CNDV
9	Y	eɪ	VMFV	30	h	h	CFGU
10	O	ɔɪ	VHCV	31	r	r	CTDV
11	1	ɑ + ʔ	VLCV	32	l	ɫ	CRDV
12	2	ɛ + ʔ	VMFV	33	w	w	CSBV
13	3	ə + ʔ	VMFV	34	y	j	CSPV
14	4	i + ʔ	VHFV	35	K	x	CFVU
15	5	ɔ + ʔ	VMKV	36	G	ŋ	CNVV
16	6	u + ʔ	VHKV	37	N	ɲ	CNPV
17	b	b	CPBU	38	S	ʃ	CFPU
18	p	p	CPBV	39	*		****
19	t	t	CPDU	40	-		****
20	d	d	CPDV	41	space		****
21	k	k	CPVU				

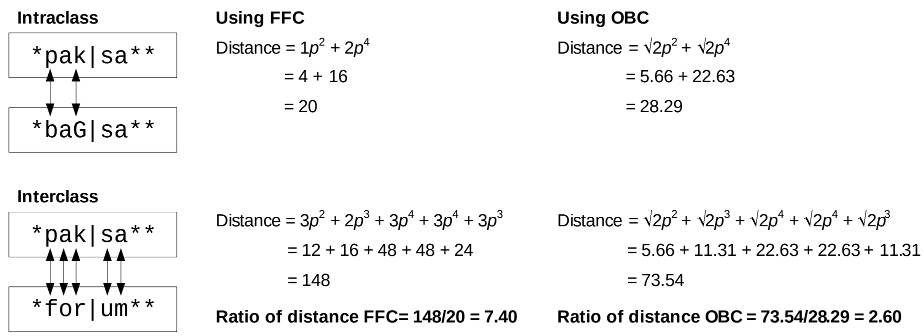

Figure 6: Distance ratios between interclass and intraclass patterns using FFC and OBC using $p = 2$, where '|' is a candidate for syllabification point and /*/ is 'no phoneme'

they have one different feature. In contrast, two phonemes with completely different features, such as /a/ (encoded into VLCV) and /b/ (CPBU), have a maximum distance of 4 since they have four different features.

The FFC produces more values (integer 0 to 4) than OBC (only $\sqrt{2}$). In the case of syllabification, it produces a higher ratio of the distance between interclass and intraclass patterns than OBC, as illustrated by Figure 6. Two intraclass patterns `*pak|sa**` and `*baG|sa**` (have 2 different phonemes on the 2nd and 4th left contexts) with $p = 2$ produce small distance $= 1 \times 2^2 + 2 \times 2^4 = 20$. On the other hand, two interclass patterns `*pak|sa**` and `*for|um**` have a much bigger distance, up to 148. Thus, FFC produces distance ratio of 7.40. But, OBC gives lower distance ratio of 2.60. Hence, the FFC is predicted to be capable of making the intraclass patterns closer and the interclass patterns further. However, many examples are needed to make this prediction much more valid. An observation on randomly selected 1K patterns in Class 0 and 1K patterns in Class 1 using FFC and $p = 2$ gives average distances of intraclass patterns 142.81 and interclass patterns 177.58. It produces a distance ratio of $177.58/142.81 = 1.24$, higher than OBC wich gives $79.82/75.59 = 1.06$.

PNNR-based syllabification. PNNR to classify two classes works by finding the minimum probabilistic nearest neighbour distance between the current pattern

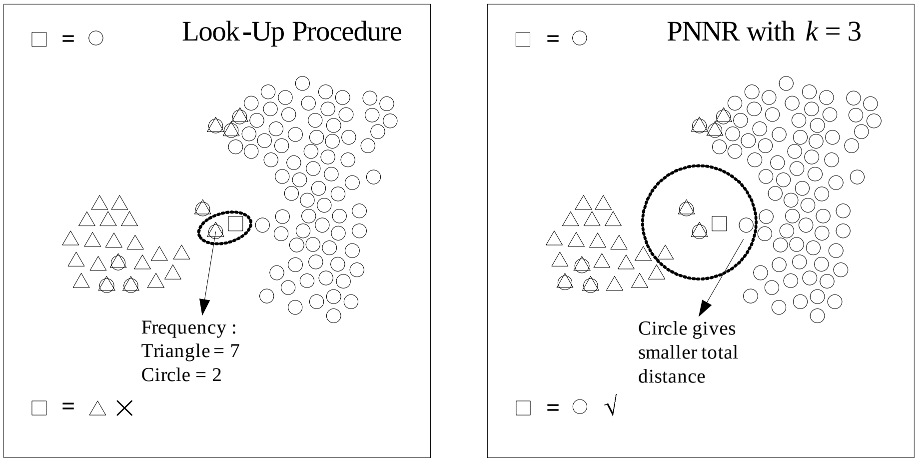

Figure 7: Difference between LUP-based global classifier (all patterns in the training set are taken into account) and PNNR-based local classifier (only unique patterns are taken into account)

and both classes. The neighbourhood weight for the j-th neighbour, u_j, proposed by Zeng [24] is formulated as

$$u_j = \frac{1}{j} \tag{2}$$

where j is the ranking in ascending order based on the distance.

PNNR works locally, where only k unique patterns in the training set are taken into account. This scheme is conceptually better for handling anomalies than LUP that works globally by taking into account all patterns in the training set, as illustrated by Figure 7. However, to adapt to a given case the neighbourhood weight in PNNR needs to be modified by adding an exponential constant c as proposed by Suyanto [21], which is formulated in Equation 3

$$u_j = \frac{1}{j^c} \tag{3}$$

where c is a real value around 1.

The motivation to modify the neighbourhood weight is conceptually illustrated by Figure 8. For an Indonesian G2P, the optimum c is 1.6 that produces

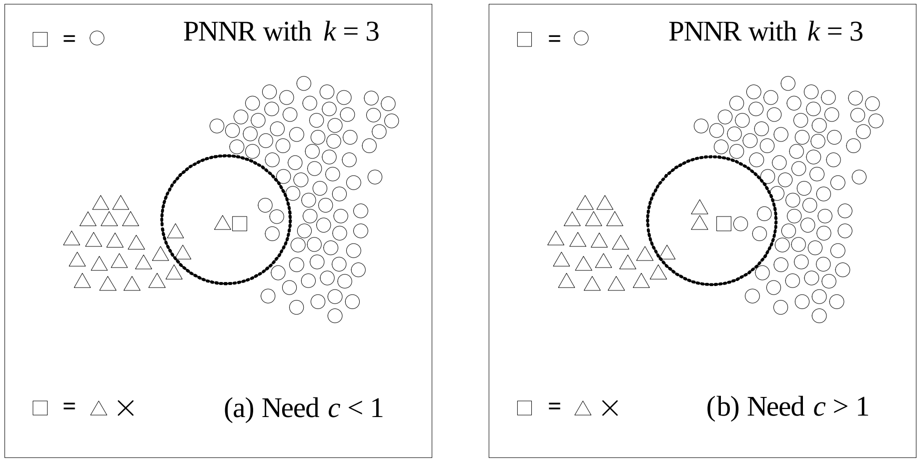

Figure 8: Two different cases: need $c < 1$ (left) and need $c > 1$ (right)

a relative PER reduction of 1.78% (compared to the original formula, where c = 1.0) [25].

PNNR for syllabification needs to decide between two classes: syllabification boundary or not, and works by finding the minimum probabilistic nearest neighbour distance between the current pattern and both classes. The neighbourhood weight for the j-th neighbour, u_j, is formulated in Equation 2.

The total distance between the current pattern and a class taking into account the k closest patterns is calculated using Equation 4

$$T = \sum_{j=1}^{k} u_j \sum_{i=1}^{L/2} (d_{li} w_i + d_{ri} w_i) \tag{4}$$

where u_j is the weight for the j-th neighbour, L is the contextual length, and d_{li} and d_{ri} are the distances of the i-th contextual phoneme on the left and right calculated using the four-feature phoneme encoding.

How the PNNR-based syllabification works is illustrated by Figure 9. Since /m/ and /p/ are split based on the phonotactic rule, a phoneme sequence /em.pati/ is converted into two patterns, (a) and (b), that correspond to the two possible syllabification boundaries in /pati/. Using k = 3, L = 8, p = 2.0, and c = 1.0, the first pattern is classified as syllabification point, but the second one is not. Thus, /em.pati/ is syllabified as /em.pa.ti/.

The example in Figure 9 is an illustration of the common case in which one of the syllabification points is unambiguously chosen. However, the application of Equation 4 only may be insufficient as it may suggest zero or multiple syllabification points. These problems can be solved simply by maximizing the ratio of the total distance of Class 1 and Class 0. For example, if the pattern empa|ti** produces total distance of Class 0 = 3 and Class 1 = 7 and the pattern mpat|i*** gives total distance of Class 0 = 21 and Class 1 = 29, then maximizing the ratio of total distance of Class 1 and Class 0 shows that the pattern empa|ti** is the winner and thus the phoneme sequence /em.pati/ is syllabified as /em.pa.ti/.

3. Results and Discussion

The dataset used in this research is a set of 50k words with corresponding phoneme sequences and their syllabification points. First, the dataset is randomly split into five subsets of 10K different words each. In a fold cross-validation, 40k words are used for parameter tuning and 10k for evaluation. For computational reasons each parameter is sequentially tuned from the hardest (no knowledge to predict) to the easiest using the described five fold cross validation procedure.

Phoneme encoding. First the PNNR-based syllabification model without phonotactic knowledge is evaluated to see the performance of the proposed four-feature encoding. Here, the model is tuned using some intuitive values of parameters as described in Section 2, i.e $c = 1.0$, $p = 2.0$, $L = 8$, and the hardest predicted parameter k is assumed to be 5 (slightly lower than the optimum $k = 6$ that used by a more complex model, Indonesian PNNR-based G2P [25]. Five-fold cross-validating shows that the four-feature encoding produces lower average syllable error rate (SER) as well as lower word error rate (WER) when compared to the orthogonal binary encoding as listed in Table 4. It gives SER of 0.80%, significantly lower than the orthogonal binary encoding that produces 0.93%. It relatively reduces the SER by 13.90%. This result proves that the

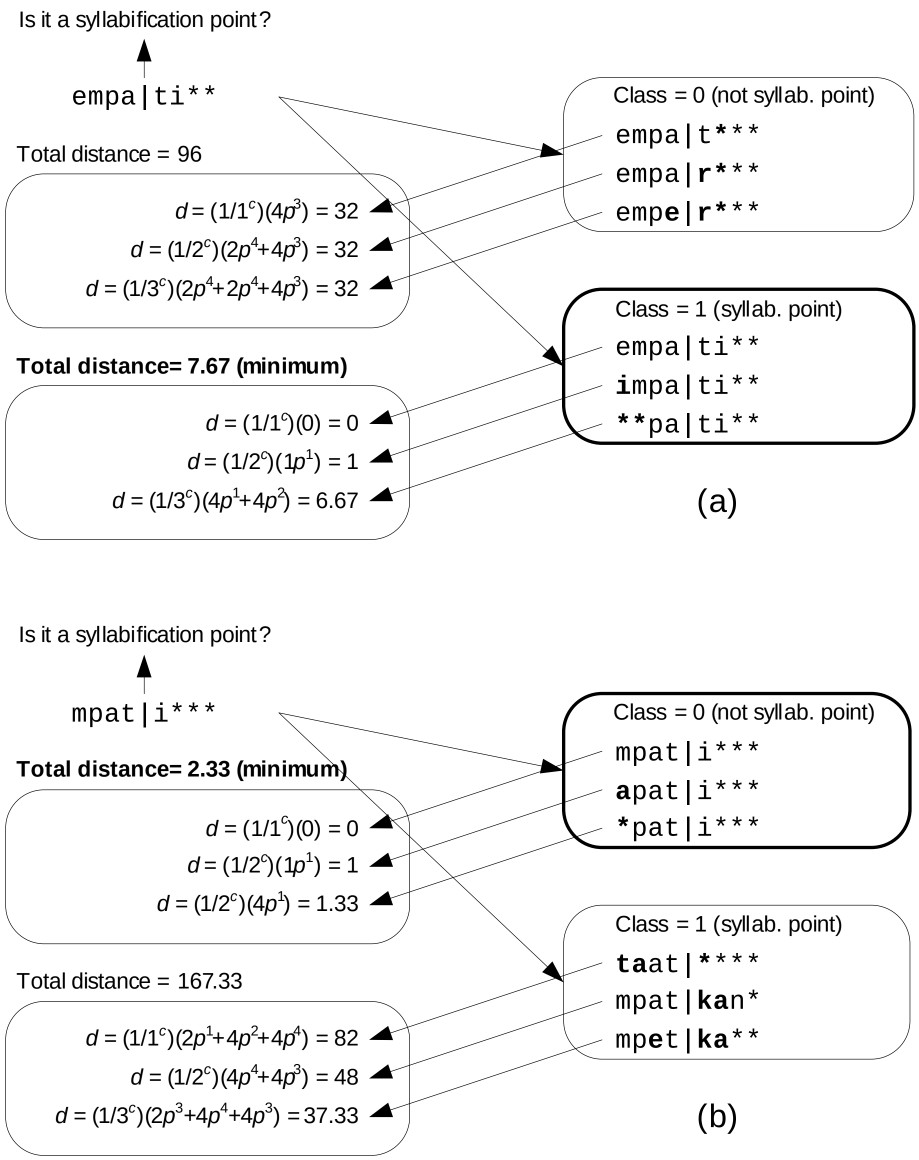

Figure 9: PNNR-based syllabification, using $k = 3$, $L = 8$, $p = 2.0$, and $c = 1.0$, syllabifies a phoneme sequence /em.pati/ into /em.pa.ti/

Table 4: Comparison of phoneme encoding

Phoneme encoding	Average SER	Average WER
Orthogonal binary code	0.93%	1.54%
Four-feature code	0.80%	1.32%

Table 5: Comparison of phonotactic knowledge

PNNR-based syllabification	Average SER	Average WER
PNNR without phonotactic knowledge	0.80%	1.32%
PNNR with phonotactic knowledge	0.66%	1.11%

proposed encoding, which produces smaller distances for patterns containing phonemes with some similar features, makes the PNNR capable of clustering the patterns more accurately.

Phonotactic knowledge. Secondly, PNNR with phonotactic knowledge is evaluated. Both PNNR without and with phonotactic knowledge use the same values of parameters: $k = 5$, $c = 1.0$, $p = 2.0$, and $L = 8$. The result in Table 5 shows that PNNR with phonotactic knowledge produces average SER of 0.66%, lower than the PNNR without phonotactic knowledge that gives 0.80%. It relatively reduces the SER by 17.17%.

Based on those results, the PNNR is designed to use both four-feature encoding and phonotactic knowledge. Next, the paramaters of PNNR are sequentially tuned.

Neighbourhood size k. The number of neighbour, also called neighbourhood size, k in the PNNR is so varying based on the problem that it is difficult to be predicted. Hence, the PNNR is evaluated for varying k with $c = 1.0$, $p = 2.0$, and $L = 8$. The results in Figure 10 show that when $k = 1$ PNNR commonly produces high SER since considering only one neighbour can lead it to be a too general clustering. It also gives high SER when considers so many neighbours,

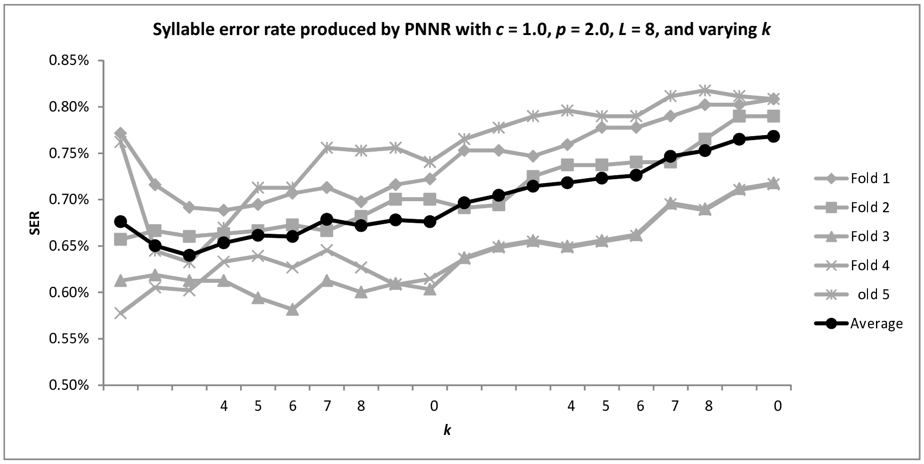

Figure 10: Performance of PNNR-based syllabification for varying k

10 or more, that make it be a too specific clustering. On average it produces the lowest SER when $k = 3$.

Power constant for neighbourhood weight c. Next, the PNNR with $k = 3$, $p = 2.0$, and $L = 8$ is evaluated using varying c. The result, as illustrated by Figure 11, shows that when c is less than 1.0 the PNNR yields high SER since the closer neighbour has quite similar distance to the further one. It also produces high SER when c is 1.4 or more since the closer neighbour has too high distance and the further is too low. It produces stable SER when $c = 1.0$ (with SER = 0.642%) to $c = 1.3$ (with the lowest SER = 0.640%). This result shows that the proposed constant c does not significantly reduce the SER compared to the original version of neighbourhood weight (with $c = 1.0$) proposed by Zeng [24], where the relative SER reduction is only $(0.642 - 0.640)/0.642 = 0.31\%$, since the optimum k is very small (3). It is different to the PNNR-based Indonesian G2P model with optimum $k = 6$, where $c = 1.6$ produces relative PER reduction of 1.78% compared to $c = 1.0$ [25].

Exponential constant for contextual weight p. The PNNR with $k = 3$, $c = 1.3$, and $L = 8$ is then evaluated using varying p. The result, as illustrated by Figure 12, shows that when p is so small, less than 2.0, the PNNR yields high SER

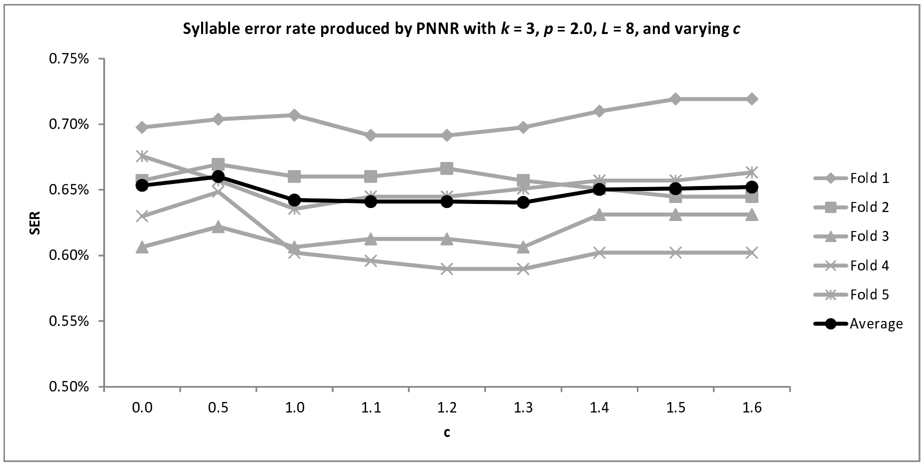

Figure 11: Performance of PNNR-based syllabification for varying c

because the importance of closer contextual phonemes is quite similar to the further ones. It gives the lowest SER of 0.64% when $p = 2.0$.

Contextual length L. The PNNR with $k = 3$, $c = 1.3$, and $p = 2.0$ is then evaluated using varying L. The result, as illustrated by Figure 13, shows that when L is 6 or less, the PNNR yields high SER since considering few contextual phonemes will lead to many ambiguous syllabification patterns. It gives stable low SER when L is 8 or more.

Syllabification errors. The most syllabification errors, about 60%, come from some derivatives with three prefixes 'ber', 'per', and "ter', where the PNNR can not distinguish them to the roots beginning with those such phoneme sequences. For example, /beragam/ (diverse) is syllabified as /be.ra.gam/ but /beragama/ (religious) should be split into /ber.a.ga.ma/. The second errors, around 20%, are from some compound words, such as /anorganik/ (inorganic) that should be syllabified as /an.or.ga.nik/ but the PNNR produces /a.nor.ga.nik/. The others come from some roots and derivatives with prefix 'meng' as well as suffixes 'an' and 'i'.

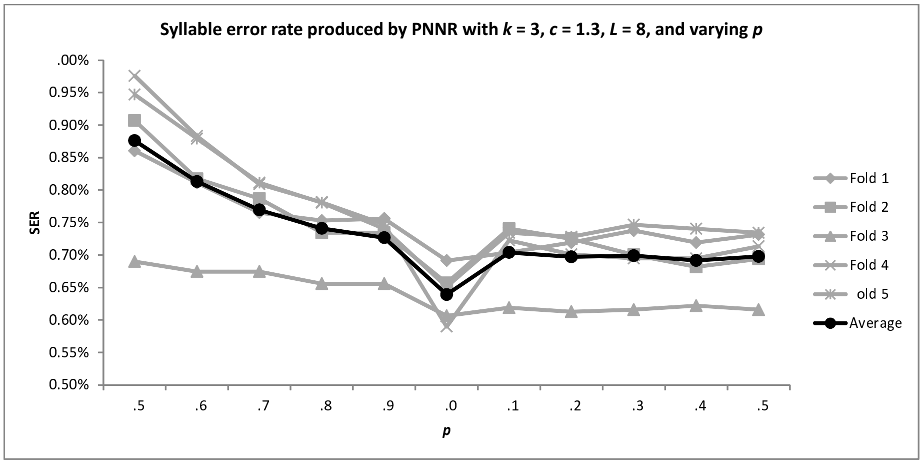

Figure 12: Performance of PNNR-based syllabification for varying p

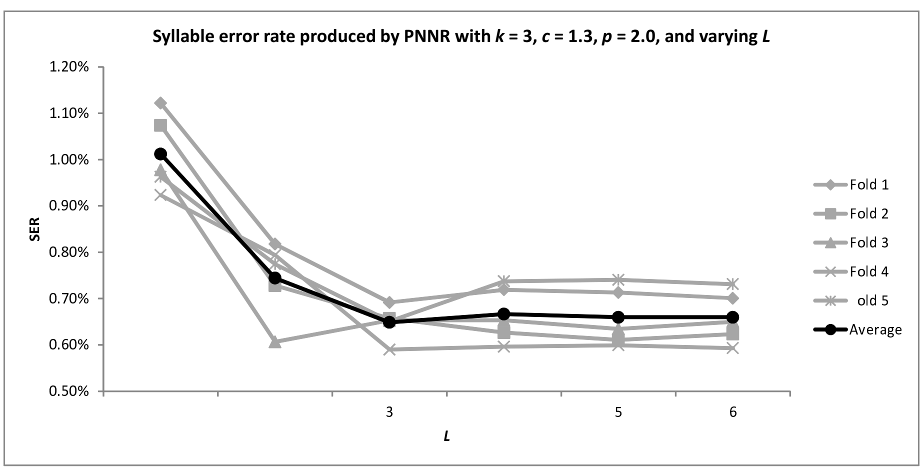

Figure 13: Performance of PNNR-based syllabification for varying L

LUP as Benchmark. Here LUP is reimplemented using the best weighting vector for English, [1 4 16 64 16 5 1], as used by Marchand [18]. It is implemented in a straightforward manner, with as only embedded knowledge that a syllable needs to contain exactly one vowel. The benchmark using 5-fold cross-validation is listed in Table 6. PNNR gives significantly lower both SER and WER than LUP because of two reasons: 1) PNNR classifies patterns locally (but LUP does globally) so that it is more capable of handling many syllabification anomalies in Indonesian; 2) PNNR exploits FFC (while LUP uses binary values) so that it defines a more precise distance between two phonemes.

Table 6: Comparison of PNNR and LUP

Fold	SER (LUP)	SER (PNNR)	WER (LUP)	WER (PNNR)
1	2.64%	0.69%	4.29%	1.15%
2	2.51%	0.66%	4.08%	1.22%
3	2.58%	0.61%	4.17%	1.04%
4	2.41%	0.59%	3.93%	0.99%
5	2.85%	0.65%	4.62%	1.07%
Average	2.60%	0.64%	4.22%	1.09%
Std. dev.	0.16%	0.04%	0.26%	0.09%

4. Conclusion

The proposed four-feature phoneme encoding significantly reduces the SER by 13.90% relatively to the previously used orthogonal binary encoding. The phonotactic-based pre-syllabification reduces the SER up to 17.17% relatively to the PNNR without pre-syllabification. Five-fold cross-validating proves that the PNNR-based syllabification is stable by producing average SER of 0.64%. The most errors come from derivatives with thre prefixes 'ber', 'per', and 'ter' as well as from compound words. This result is also significantly lower than

LUP-based syllabification that gives average SER of 2.60%. As a data-driven method, the PNNR-based syllabification can be applied to other languages, but the four-feature phoneme encoding and the phonotactic-based pre-syllabification should be modified based on the phoneme categorization and the phonotactic constraints in those languages.

Acknowledgment

The authors would like to thank to Telkom RDC and *Tim Penyusun Kamus Pusat Bahasa* for the text corpus as well as the Directorate General of Higher Eduction (DIKTI) for funding the sandwich-like program in Departement Elektrotechniek-ESAT, KU Leuven, Belgium.

References

[1] K. Rogova, K. Demuynck, D. Van Compernolle, Automatic syllabification using segmental conditional random fields, Computational Linguistics in the Netherlands Journal 3 (2013) 34–48.

[2] R. Weerasinghe, A. Wasala, K. Gamage, A rule based syllabification algorithm for Sinhala, in: Proceedings of the second International Joint Conference on Natural Language Processing, 2005, pp. 438–449. doi:10.1007/11562214_39.

[3] Z. Hernández-Figueroa, F. J. Carreras-Riudavets, G. Rodríguez-Rodríguez, Automatic syllabification for Spanish using lemmatization and derivation to solve the prefix's prominence issue, Expert Systems with Applications 40 (2013) 7122–7131. doi:10.1016/j.eswa.2013.06.056.

[4] M. Hafiz, K. Rabiah, A, A. Azreen, A. M Taufik, Syllabification algorithm based on syllable rules matching for Malay language, in: Proceedings of The 10th WSEAS International Conference on Applied Computer and Applied Computational Science, 2011, pp. 279–286.

[5] W. Daelemans, A. van den Bosch, T. Weijters, IGTree: Using trees for compression and classification in lazy learning algorithms, Artificial Intelligence Review 11 (1-5) (1997) 407–423. doi:10.1.1.29.4517.

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

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

[8] W. Daelemans, A. van den Bosch, A neural network for hyphenation, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN92), Brighton, United Kingdom, 1992, pp. 1647–1650 (vol. 2). doi:10.1016/B978-0-444-89488-5.50176-7.

[9] T. Kristensen, A neural network approach to hyphenating Norwegian, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IEEE, 2000, pp. 148–153 vol.2. doi:10.1109/IJCNN.2000.857889.

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

[11] K. Müller, Improving syllabification models with phonotactic knowledge, in: Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology - SIGPHON '06, 2006, pp. 11–20. doi:10.3115/1622165.1622167.

[12] H. Schmid, B. Möbius, J. Weidenkaff, Tagging syllable boundaries with joint n-gram models, Proceedings of the Annual Conference of the Interna-

tional Speech Communication Association, INTERSPEECH 1 (1) (2007) 49–52.

[13] S. Bartlett, G. Kondrak, C. Cherry, On the syllabification of phonemes, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 308–316. doi:10.3115/1620754.1620799.

[14] C. R. Adsett, Y. Marchand, A comparison of data-driven automatic syllabification methods, in: Proceedings of the 16th International Symposium on String Processing and Information Retrieval (SPIRE), Springer Berlin Heidelberg, 2009, pp. 174–181. doi:10.1007/978-3-642-03784-9_17.

[15] T. Mayer, Toward a totally unsupervised, language-independent method for the syllabification of written texts, in: Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, no. July, 2010, pp. 63–71.

[16] Y. Marchand, C. R. Adsett, R. I. Damper, Evaluating automatic syllabification algorithms for English, in: Proceedings of the 6th International Speech Communication Association ISCA Workshop on Speech Synthesis, 2007, pp. 316–321.

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

[18] Y. Marchand, C. R. Adsett, R. I. Damper, Automatic syllabification in English: a comparison of different algorithms, Language and speech 52 (Pt 1) (2009) 1–27. doi:10.1177/0023830908099881.

[19] R. M. Dauer, Stress-timing and syllable-timing reanalyzed, Journal of Phonetics 11 (1) (1983) 51–62.

[20] H. Alwi, S. Dardjowidjojo, H. Lapoliwa, A. M. Moeliono, Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar), 3rd Edition, Balai Pustaka, Jakarta, 1998.

[21] Suyanto, A. Harjoko, Nearest neighbour-based Indonesian G2P conversion, Telkomnika (Telecommunication, Computing, Electronics, and Control) 12 (2) (2014) 389–396.

[22] S. Bartlett, G. Kondrak, C. Cherry, Automatic syllabification with structured SVMs for letter-to-phoneme conversion., in: Proceedings of Human Language Technologies: The 2008 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 568–576.

[23] A. Chaer, Fonologi Bahasa Indonesia (Indonesian Phonology), Rineka Cipta, Jakarta, 2009.

[24] Y. Zeng, Y. Yang, L. Zhao, Pseudo nearest neighbor rule for pattern classification, Expert Systems with Applications 36 (2) (2009) 3587–3595. `doi:10.1016/j.eswa.2008.02.003`.

[25] Suyanto, S. Hartati, A. Harjoko, Modified Grapheme Encoding and Phonemic Rule to Improve PNNR-Based Indonesian G2P, International Journal of Advanced Computer Science and Applications (IJACSA) 7 (3) (2016) 430–435.

Evidences of correspondences

Indonesian Syllabification Using
Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. LoA with Major Revision (22 September 2016)
3. Respond to Reviewers (11 October 2019)
4. Second Submission (11 October 2019)
5. **LoA with Minor Revision (21 October 2019)**
6. Final Submission (24 October 2019)
7. LoA with Fully Accepted (28 October 2019)

Ref: SPECOM_2016_8_R1
Title: Indonesian Syllabification Using a Pseudo Nearest Neighbour Rule and Phonotactic Knowledge
Journal: Speech Communication

Dear Mr. Suyanto,

Thank you for submitting your manuscript to Speech Communication. I have received comments from reviewers on your manuscript. Your paper should become acceptable for publication pending suitable minor revision and modification of the article in light of the appended reviewer comments.

When resubmitting your manuscript, please carefully consider all issues mentioned in the reviewers' comments, outline every change made point by point, and provide suitable rebuttals for any comments not addressed.

To submit your revised manuscript:

- Log into EVISE® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=SPECOM
- Locate your manuscript under the header 'My Submissions that need Revisions' on your 'My Author Tasks' view
- Click on 'Agree to Revise'
- Make the required edits
- Click on 'Complete Submission' to approve

What happens next?

After you approve your submission preview you will receive a notification that the submission is complete. To track the status of your paper throughout the editorial process, log in to EVISE® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=SPECOM.

Enrich your article to present your research with maximum impact. This journal supports the following Content Innovations:

- Explain your research in your own words and attract interest in your work using AudioSlides : 5-minute webcast-style presentations that are displayed next to your published article and can be posted on other websites. You will receive an invitation email to create an AudioSlides presentation within three weeks after your paper has been accepted.

- Interactive Plots: Interactive plot viewer providing easy access to the data behind plots. Please prepare a .CSV file with your plot data and test it online here before submitting as supplementary material.

I look forward to receiving your revised manuscript as soon as possible.

Kind regards,

Paul Foulkes
Subject Editor
Speech Communication

Comments from the editors and reviewers:
-**Reviewer 1**

-

The authors did a good job to take account my previous concerns/remarks.
I have now only a few minor revisions to suggest.

Page 2: "However, the SbA, which globally classifies a pattern by finding the best analogy (minimizing shortest path using three metrics) in the training set [18], may not suitable for a language with many exceptions and ambiguities such as Indonesian. [...] ...a root /ragam/ (various)": this is a strange statement/assumption. All the data-driven methods have to deal with this issue. In addition, SbA seems to have been tested (with relatively a good succes) on the English language that is even more inconsistent than the Indonesian language.

Page 2: "created by Tim Penyusun 40 Kamus Pusat Bahasa, Kementerian Pendidikan dan Kebudayaan
(The Lexicographer Team of Language Center, Indonesian Ministry of Education and Culture)":
to make the sente simpler to read, the above segment shoul be reference within a footnote.

Page 2: "that Indonesian" -> "that Indonesian language"

Page 2: "monosyllabics that only account" -> "monosyllabic words that only account"

page 2: "20% monosyllabics" -> "20% monosyllabic words"

Page 4: The tails of both the figures 1 and 2 are not particularly informative and could be
removed.

Page 5: "by Marchand [18]" -> "by Marchand and colleagues [18]"

Page 10: "described by Marchand [18])" -> "described by Marchand amnd colleagues [18])"

Page 22: "as used by Marchand [18]" -> "as used by Marchand and colleagues [18]"

-**Reviewer 2**

- I believe the authors have taken into account my comments in the previous review and that the article is now suitable for publication.

Evidences of correspondences

Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. LoA with Major Revision (22 September 2016)
3. Respond to Reviewers (11 October 2019)
4. Second Submission (11 October 2019)
5. LoA with Minor Revision (21 October 2019)
6. **Final Submission (24 October 2019)**
7. LoA with Fully Accepted (28 October 2019)

Comments from the editors and reviewers:

Reviewer 1

The authors did a good job to take account my previous concerns/remarks.
I have now only a few minor revisions to suggest.

Page 2: "However, the SbA, which globally classifies a pattern by finding the best analogy (minimizing shortest path using three metrics) in the training set [18], may not suitable for a language with many exceptions and ambiguities such as Indonesian. [...] ...a root /ragam/ (various)": this is a strange statement/assumption. All the data-driven methods have to deal with this issue. In addition, SbA seems to have been tested (with relatively a good succes) on the English language that is even more inconsistent than the Indonesian language.

The paragraph has been deleted.

We added the following sentence in the next paragraph (in between "… listed in Table 1." and "In contrast, English …. ")

"On the basis of the average syllabic complexity one might conclude that Indonesian is a language with low syllabic complexity."

Page 2: "created by Tim Penyusun 40 Kamus Pusat Bahasa, Kementerian Pendidikan dan Kebudayaan
(The Lexicographer Team of Language Center, Indonesian Ministry of Education and Culture)":
to make the sente simpler to read, the above segment shoul be reference within a footnote.

We corrected the sentence and put some of the information is a footnote for better readability: "The Lexicographer Team of the Language Center of the Indonesian Ministry of Education and Culture"

Page 2: "that Indonesian" -> "that Indonesian language"

"that Indonesian" → "that Indonesian language"

Page 2: "monosyllabics that only account" -> "monosyllabic words that only account" “monosyllabics that only account" → " monosyllabic words that only account”

page 2: "20% monosyllabics" -> "20% monosyllabic words"

"20% monosyllabics" → "20% monosyllabic words"

Page 4: The tails of both the figures 1 and 2 are not particularly informative and could be removed.

The tails of both figures 1 and 2 has been removed. In Figure 1, the number of syllables are plotted from 1 to 7 (instead of 1 to 9). In Figure 2, the number of phonemes are plotted from 1 to 16 (instead of 1 to 23).

Page 5: "by Marchand [18]" -> "by Marchand and colleagues [18]"

"by Marchand [18]" → "by Marchand and colleagues [18]"

Page 10: "described by Marchand [18])" -> "described by Marchand amnd colleagues [18])"

"described by Marchand [18])" → "described by Marchand and colleagues [18])"

Page 22: "as used by Marchand [18]" -> "as used by Marchand and colleagues [18]"

"as used by Marchand [18]" → "as used by Marchand and colleagues [18]"

Reviewer 2

I believe the authors have taken into account my comments in the previous review and that the article is now suitable for publication.

Evidences of correspondences

Indonesian Syllabification Using Pseudo Nearest Neighbour Rule and Phonotactic Knowledge

1. First Submission (07 January 2016)
2. LoA with Major Revision (22 September 2016)
3. Respond to Reviewers (11 October 2019)
4. Second Submission (11 October 2019)
5. LoA with Minor Revision (21 October 2019)
6. Final Submission (24 October 2019)
7. **LoA with Fully Accepted (28 October 2019)**

Ref: SPECOM_2016_8_R2
Title: Indonesian Syllabification Using a Pseudo Nearest Neighbour Rule and Phonotactic Knowledge
Journal: Speech Communication

Dear Mr. Suyanto,

I am pleased to inform you that your paper has been accepted for publication. My own comments as well as any reviewer comments are appended to the end of this letter. Now that your manuscript has been accepted for publication it will proceed to copy-editing and production.

Thank you for submitting your work to Speech Communication . We hope you consider us again for future submissions.

Kind regards,

Paul Foulkes
Subject Editor
Speech Communication

Comments from the editors and reviewers:

Have questions or need assistance?
For further assistance, please visit our Customer Support site. Here you can search for solutions on a range of topics, find answers to frequently asked questions, and learn more about EVISE® via interactive tutorials. You can also talk 24/5 to our customer support team by phone and 24/7 by live chat and email.
