Augmented-syllabification of n-gram tagger for Indonesian words and named-entities

Suyanto Suyanto a,*, Andi Sunyoto b, Rezza Nafi Ismail a, Ade Romadhony a, Febryanti Sthevanie a

a School of Computing, Telkom University, Bandung, Indonesia
b Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia

HIGHLIGHTS

• A new augmented graphemic-syllabification of n-gram tagger model is proposed.
• Three augmentation methods are applied to the training sets to reduce the OOV rate.
• The n-gram tagger is developed using combined original and augmented words.
• Evaluation is based on 5-fold cross-validation of formal words and named-entities.
• The proposed model significantly outperforms three recent models.

ARTICLE INFO

Keywords:
Bahasa Indonesia
Data augmentation
Graphemic syllabification
n-Gram tagger
Transposing nuclei

ABSTRACT

As one of the statistical-based models, an n-gram syllabification commonly gives a high syllable error rate (SER) for Bahasa Indonesia, one of the low-resource languages, since it fails for a high out-of-vocabulary (OOV) rate. Two previous models: bigram-syllabification with flipping onsets (BFO) and a combination of bigram with backoff smoothing based on phonological similarity (CBSPS), which use augmentation methods, can reduce the OOV rate. However, there are two problems in both BFO and CBSPS. First, they use an n-gram that is applied syllable-level, instead of grapheme-level, so that they suffer on the sparsity of n-grams. Second, they rely on a procedure to detect the positions of both vowels and diphthongs. Both problems make them not capable of distinguishing diphthongs from derivative words as well as syllabifying named-entities, which have many ambiguities related to vowels and semi-vowels. In this paper, a syllabification based on an n-gram tagger, which is applied on grapheme-level and does not rely on both vowel and diphthong detections, is developed to solve both problems. Besides, three data augmentation methods are exploited to enrich the dataset. The 5-fold cross-validations (5-FCV) using both datasets of 50 k words and 15 k named-entities show that the proposed augmented-syllabification of n-gram tagger (ASnGT) model is significantly better than both BFO and CBSPS. It is also significantly better than the fuzzy k-nearest neighbor in every class (FkNNC)-based model for formal words and named-entities. However, it suffers from derivative words, where it cannot easily distinguish them from both absorption words and terms of foreign languages. Besides, it also undergoes some foreign named-entities.

1. Introduction

A syllabification model, which functions to decompose a word into syllables automatically, plays an essential role in the linguistics area. In general, it is widely used in many applications based on spoken language processing and linguistics, such as speech synthesizers [8], [11], [17], speech recognizers [7], [19], [26], emotion classification [2], speaker verification [16], speaking rate estimation [31], word count estimation [20], [24], and development of a speech corpus for text-to-speech or speech-to-text models [4].

* Corresponding author.
E-mail address: suyanto@telkomuniversity.ac.id (S. Suyanto).

https://doi.org/10.1016/j.heliyon.2022.e11922
Received 31 March 2022; Received in revised form 3 November 2022; Accepted 18 November 2022

2405-8440/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
A dataset of words and their syllabifications:
1. bari  ba.ri

Tagging syllables for original word

List of original tagged-syllables:
1. b'αɾi

Swapping consonant-graphemes

List of swapped-words and syllabifications:
1. bali  ba.li
2. pari  pa.ri
3. pali  pa.li

Flipping onsets for all the original and swapped-words

List of flipped-words and syllabifications:
4. rabi  ra.bi
5. labi  la.bi
6. rapi  ra.pi
7. lapl  la.pl

Transposing nuclei for all the original, swapped-, and flipped-words

List of transposed-words and syllabifications:
8. bira  bi.ra
9. bila  bi.la
10. pira  pi.ra
11. pila  pi.la
12. ria  ri.α
13. liba  li.ba
14. ril  ri.l
15. lipa  li.pα

Tagging syllables for all augmented-words

List of tagged-syllables for augmented-words:
1. b'αɾi
2. p'αɾi
3. p'αɾi
4. r'αɾi
5. l'αɾi
6. l'αɾi
7. l'αɾi
8. b'ıɾa
9. b'ıɾa
10. p'ıɾa
11. p'ıɾa

Fig. 1. Training of ASnGT.

The graphemic syllabification commonly produces higher SER than the phonemic one [15]. However, it is faster and simpler, not only for the formal words but also for the named-entities without any specific linguistics knowledge. Therefore, in [27], syllabification is recommended to be based on a sequence of graphemes rather than phonemes because of the flexibility as well as simplicity.

In literature, a statistical-based syllabification is preferable to the rule-based one since it is much simpler in development [1]. Many researchers have proposed various methods, such as decision tree [13], context-free grammars [14], nearest neighbor [15], syllabification by analogy [1], finite-state transducers [9], dropped-and-matched model [18], hidden Markov model [10], conditional random fields [25], and n-gram [6]. Neural language models described in [12], [30] shows a high performance (with low SERs). However, some n-gram models proposed in [6], [29], [23] surprisingly produce comparable performances, and they are noticeably faster.

In [21], the researchers prove that a simple n-gram gives a low SER for the German language. It is also flexible for other languages with no specific knowledge. However, it is insufficient for a small dataset that generates massive out-of-vocabulary (OOV) bigrams. In [27], a simple bigram and flipping onsets model (BFO) is proposed to handle such a problem. This model is better than the standard bigram model with a relative SER decrement of 18.02%. Unfortunately, it cannot also tackle a small dataset. In [28], another simple model called combined standard bigram and swapping phonological similarities (CBSPS) is created. It performs better than BFO, which relatively reduces the SER by 31.39%. Unfortunately, both CBSPS and BFO use syllable-level bigrams (instead of grapheme-level) so long that they produce a high OOV rate. Consequently, they give high mean SERs of 13.48% and 14.15%, respectively, for a named-entity dataset (15 k entries) using a 5-FCV scheme.

Therefore, an augmented syllabification of n-gram tagger (ASnGT), which is applied on grapheme-level, is proposed in this research to tackle those issues. Three methods of augmentation: flipping onsets, swapping consonant-graphemes, and transposing nuclei are applied on both training sets (words and named-entities), which can be expected to decrease the OOV rate. Two n-gram taggers are then developed for both original and augmented words. Finally, both n-gram taggers are combined using an augmentation weight to create a decoder, which functions to define the syllabification points.
words and legal syllables for Bahasa Indonesia without shifting the points of syllabification. An investigation informs that 50 k words generate up to three million new words, which consist of 9,620,054 syllables. The interesting fact is that 87.20% of those syllables are the same as produced by the original words, and the rest 12.80% are never seen in the dataset.

Swapping consonant-graphemes As one of the simple languages, Bahasa Indonesia has less ambiguity for some phonemic symbols, where most graphemes are commonly spoken as unique phonemes. Hence, several onset graphemes can be switched using the categorization of phonemes described in [3]. There are fourteen graphemes clustering into seven groups (categories), which are simply swapped into their similar phonological one, as described in [28]. Such graphemes contained in words as well named-entities can be switched to generate other words with no changing one or more points of syllabification. Those seven categories are as follows [28]:

1. Plosive-bilabial: graphemes (b) and (p). For a given original word, swapping (b) into (p) may create a new word, such as “ba.ru” (new) is swapped into “pa.ru” (lung). Swapping (p) into (b) in a word “pe.r” (fairy) can also generate other word “be.r” (give).
2. Plosive-dental: graphemes (b) and (p). Swapping (d) to (t) in “de.bu” produces “te.bu” (cane) while switching (t) to (d) in “ts.yang” (show) creates “da.yang” (court lady).
3. Plosive-velar: graphemes (g) and (k). Swapping (g) in “ge.tar” (shakes) into (k) yields a new word “ke.tar” (daunted). Swapping (k) in “ka.mis” (Thursday) into (g) creates “ga.mis” (clothes).
4. Affricative-palatal: graphemes (c) and (j). An original word “cu.rang” (cheat) is swapped into “ji.rang” (canyon) while “ji.pang” (branch) is swapped to be “cu.pang” (bite mark).
5. Fricative-Labiodental: graphemes (f) and (v). Switching (f) to (v) in “fa.li” (thin metal) produces “vo.li” (volley). In contrast, swapping (v) in “vi.si” (vision) into (f) yields a new word “fi.si” (fission).
6. Fricative-Dental: graphemes (s) and (z). Switching (s) to (z) in “a.sam” (acid) generates “a.zam” (aim). Meanwhile, “ze.ni” (soldier) can be swapped to create “ze.ni” (art).
7. Trill/Lateral-Dental: graphemes (l) and (r). A word “lan.g.ka” (rare) is switched into “rang.ka” (frame) and “ram.bu” (sign) to be “lam.bu” (canoe).

Flipping onsets As the name suggested, this augmentation method simply works by flipping two first onsets contained in a given original word. For example, flipping two onsets (b) and (r) in a word “ba.ru” (blue) generates a new word “ri.bu” (thousand). This method is just applied to two first onsets in the original word. In [27], flipping three or more onsets in a word produces many OOV words instead.

Transposing nuclei This augmentation method transposes two first nuclei in a given original word. First, the positions of two nuclei should be detected. Both nuclei are then transposed to create a new word. A nucleus can be one of the four possible components: single vowel, diphthong, semi-vowel, or random sequence of vowels. For instance, transposing two first nuclei of single vowels (a) and (i) in a word “sa.ki” (sick) generates another word “si.kat” (brush), transposing two first nuclei of a single vowel (u) and a semi-vowel (y) in a person name “bu.dy” creates “by.du”.

Combining three augmentation methods Sequential combining those three augmentation methods enlarges the dataset to 60 folds since they consider all possible combinations. For instance, let the given original word is “ba.ri” (while). Firstly, swapping both (b) and (p) into their phonological-similar graphemes creates three new words: “ba.li” (an island in Indonesia), “pa.ri” (stingray), and “pa.li” (taboo). Secondly, flipping onsets in the original word builds “ra.bi” (rabbitt). Next, turning onsets in those swapped-words generates other words: “la.bi” (soft
turtule), “ra.pi” (neat), and “la.pi” (OOV). Finally, transposing nuclei in the original word yields an OOV word “bl.rr” and exchanging nuclei in those augmented words builds “bi.la” (if), “pl.rr” (OOV), “pl.la” (OOV), “rk.ba” (usury), “ll.ba” (OOV), “rk.pa” (OOV), and “ll.pa” (OOV). Thus, the three methods are capable of augmenting “ba.ri” to be fifteen other words: eight formal and seven OOV.

2.2. Syllable tagger

A syllable tagger is formulated as follows. Let a word $W$ be represented as a sequence of letter $u_i = w_1, w_2, \ldots, w_n$. Searching the most likely sequence of tag $t_i = t_1, t_2, \ldots, t_n$ of $W$ can be seen as selecting a sequence of tag $t_i$ that reaches the maximum $P(t_i | u_i)$, a conditional probability calculated as in Eq. (1):

$$\arg\max_{t_i} P(t_i | u_i) = \arg\max_{t_i} P(t_i | P(u_i | t_i^*)).$$

A Markov assumption is then applied here. Thus, the probability of tag $t_i$ contextually depends on the previous tags with a contextual length or size of $k$, which is formulated as in Eq. (2):

$$P(t_i | u_i) = \prod_{j=i-k}^{i} P(t_j | t_{i-k}, \ldots, t_{i-1}).$$

In the tagging process, a padding tag $t_i = (#)$ is defined for $i < 1$. It makes sure the probability for all $t_i$, where $1 \leq i \leq n$, is taken into account. Besides, as described in [21], a tag of word-end marker ($\natural$) is defined for $t_{n+1}$ to find the minimum probability of improper syllabification.

Next, a restriction is applied, where only one letter $w_i$ is used to represent every tag $(u_i!)$ and $(u_i^*)$. Hence, for $t_i \in \{w_i, w_i^*\}$, the chance $P(w_i | t_i)$ must be 1 while $P(w_j | t_i)$ should be 0 for $w_j \neq w_i$, where $P(w_i | t_i)$ denotes the probability of $w_i$ that represents the tag $t_i$. Here, only two possible tags can be considered for each letter. Thus, $P(u_i | t_i^*)$ in Eq. (1) can be removed, and the probability can be reformulated as in Eq. (3):

$$\arg\max_{t_i} P(t_i | u_i) = \arg\max_{t_i} \prod_{j=i-k}^{i} P(t_j | t_{i-k}, \ldots, t_{i-1}), t_i \in \{w_i, w_i^*\}.$$

This tagging considers states for every letter, where each state $S_i$ defines tags $t_i$ (can be single or multiple) relying on $k$. The transition between two states $S_i$ to $S_j$, which is denoted as $A_{ij}$, is the conditional probability $P(t_j | t_i)$. In this research, an efficient Viterbi algorithm is exploited to maximize the probability of the tag sequences. Fig. 3 visualises the syllable tagger using the Viterbi algorithm for the word ‘bari’ (while).

2.3. Probability smoothing

A smoothing scheme called Generalized Modified Kneser-Ney (GKN) [22] is exploited here to tackle the OOV. The standard Kneser-Ney (KN) smoothing is formulated as in Eq. (4):

$$P_{KN}(t_i | u_{i-1}) = \frac{\max(c(t_{i-1}) - D, 0) + \gamma(t_{i-1})}{c(t_{i-1}) + \gamma(t_{i-1})},$$

where $c(*)$ is the tag frequency (*), occurs in the dataset of training, $\gamma(*)$ is any constant such that the sum of distribution is equal to 1, $D$ is the discount, and $k$ is the contextual length or size [5].

GKN adds a new discount bound parameter into an arbitrary number $B$, which is formulated as in Eq. (5):

$$P^B(n) = \begin{cases} 0 & \text{if } i = 0 \\ i - (i + 1) \frac{n_{i+1}(m)}{n_{i+1}(m)+n_{i+1}(m)+n_{i+1}(m)} & \text{if } i < B \\ B - \frac{n_{i+1}(m)+n_{i+1}(m)+n_{i+1}(m)}{n_{i+1}(m)+n_{i+1}(m)+n_{i+1}(m)} & \text{if } i \geq B, \end{cases}$$

where $n_{i+1}(m)$ is the total unique grams with a frequency $i$ in the $m$ sized n-gram [5]. As proven in [22], this discount parameter enhances the smoothing scheme, especially for a higher-order n-gram that produces a high OOV rate.

2.4. ASnGT model

The probability in ASnGT is developed by combining probabilities from both original and augmented words, which is formulated as in Eq. (6):

$$P_{KN} = P_{KNO} + wP_{KNA}$$

where $P_{KNO}$ is the KN smoothed probability of the tag-syllable from the original word in Eq. (4), $P_{KNA}$ is the KN smoothed probability of the tag-syllable from the augmented-words, $P_{KN}$ is the combined probability of both previous tag-syllables, and $w$ is the augmentation weight.

3. Result and discussion

Two datasets in [15] are used here to investigate the ASnGT performance compared to three similar models. The first dataset consists of 50 k words (with their syllabifications), while the second one contains 15 k named entities. Both are used for investigation based on 5-FCV.

3.1. Investigation on the first dataset

Three experiments are carried out here to optimize the three parameters sequentially. The gram size $n$ is firstly optimized using a fixed
discount bound $B = 3$ since it is the most sensitive parameter in the model. Secondly, $B$ is then optimized using the optimum $n$ that is found in the first step. ASnGT is finally compared to three other models using a metric of SER.

**Optimization of the gram size $n$** Here, ASnGT is first evaluated using a fixed $B = 3$ to find the gram size $n$. Fig. 4 shows that a small $n = 3$ gets a big mean SER. It reaches the optimum value of 5, which gives a mean SER of 0.95%. Enlarging $n$ to 6 increases the mean SER slightly.

**Optimization of discount bound $B$** Next, ASnGT is evaluated using the optimum $n = 5$ to find the optimum $B$. Fig. 5 implies that $B$ is less sensitive. It reaches low SERs for any value from 2 to 6. Its optimum value is 4 with a mean SER of 0.93%.

**Optimization of augmentation weights $\omega$** The augmentation weights $\omega$ in the proposed ASnGT are then optimized using both fixed $n$ and $B$. The simulation result is shown in Fig. 6. The optimal $\omega = 0.20$ gives the smallest SER of 0.90%. It means that the augmented words contribute 20% to the results.

**Comparing to three similar models** Here, ASnGT is investigated by comparing its performance to three similar models: BFO [27], CBSPS [28], and FkNNC [15]. An investigation is carried out using a 5-FCV scheme for a dataset containing 50 k words in [15], [27], [28]. To get fairness, all methods are in their best parameters. Fig. 7 clearly concludes that ASnGT is much better than two other $n$-gram models: BFO and CBSPS. It produces an average SER of 0.90%, which is much lower than both models, with respective mean SERs of 3.11% and 2.61%. It shows that
ASnGT relatively reduces the mean SER by 70.96% and 65.93%, respectively. It is also much better than another more complex technique called FkNNC, which gives a high mean SER of 2.27%. Hence, ASnGT provides a relative decrement of the mean SER by 60.20%.

Compared to others, the proposed ASnGT is more stable for all folds, where it produces low SERs in the range of 0.90% to 0.92%. This result is caused by its capability of distinguishing a diphthong from both the suffix and regular sequence of grapheme since it works on grapheme-level (not syllable-level). For instance, a diphthong (ai) is easily differentiated from a suffix (i) and a grapheme sequence of (e) by maximizing the probabilities of candidates tag-sequences. A careful investigation shows that solving this issue of diphthongs significantly reduces the SER produced by the previous models, as explained in [27, 28, 15], since Bahasa Indonesia has up to eighteen suffixes [3].

However, a detailed investigation informs that ASnGT suffers from derivative words containing a single or a combination of prefixes: “ber”, “per”, and “ter”. Most errors come from such derivatives as ASnGT cannot easily differentiate them from absorption words as well as foreign terms. ASnGT does not take into account any phonotactic constraint in Bahasa Indonesia. For instance, it syllabifies a grapheme sequence (berijithad) (an absorption word from Arabian language) into a wrong syllable sequence (ber.i.ji.ti.had), instead of the right one (ber.i.ji.ti.had). Another disadvantage, it has a bit higher complexity than both previous BFO [27] and CBSPS [28] models. It calculates all possible candidate tag-syllables without taking into account the common rule: “a syllable should have only one nucleus (vowel, semi-vowel, or diphthongs).” Nevertheless, this is not an urgent issue for a quite-fast personal computer today.

In the future, some Indonesian phonotactic constraints in [3] can be incorporated to improve the ASnGT performance. For instance, the wrong syllabification for a grapheme sequence (berijithad) into (ber.i.ji.ti.had) can be corrected by shifting the grapheme (j) to the left syllable since, based on Indonesian phonotactic constraints, there is no sequence of two graphemes (jt) in one syllable.
3.2. Investigation on the second dataset

ASnGT only exploits \( n \)-grams generated from both normal and augmented words (sequences of graphemes). Therefore, it is flexible for named-entities. Three experiments are carried out here to sequentially optimize the three parameters of ASnGT, which are similar to the experiments for the formal words dataset.

Optimization of the gram size \( n \) The ASnGT is firstly evaluated using a fixed \( B = 3 \) to find the gram size \( n \). Fig. 8 implies that a small \( n = 3 \) yields a high mean SER of 4.79%. Its optimum value is 4, which achieves a mean SER of 4.02%. Increasing \( n \) to 5 or 6 produces slightly higher SERs.

Optimization of the discount bound \( B \) Next, ASnGT is evaluated using the optimum \( n = 4 \) to find the optimum discount bound \( B \). Fig. 9 informs that \( B \) is stable to get low SERs for any value from 2 to 6. Its optimum value is 5, which produces a mean SER of 4.01%.

Optimization of augmentation weight \( w \) The augmentation weights \( w \) in the proposed ASnGT are then optimized using fixed \( n \) and \( B \). The empirical result is depicted in Fig. 10. The best \( w = 0.20 \) achieves the smallest SER of 0.90%. It means that the augmented words contribute 20% to the results.

Comparing to three similar models Here, ASnGT is also investigated by comparing its performance to BFO, CBSPS, and FkNNC. The investigation is carried out using a 5-FCV scheme for the named-entity dataset (containing 15 k entries) as used in [15], [27], [28]. To get fairness, all methods are in their best parameters. Fig. 11 concludes that ASnGT is much better than two other \( n \)-gram models: BFO and CBSPS. It produces an average SER of 3.92%, which is much lower than both
models, with mean SERs of 14.15% and 13.48%, respectively. It means that ASnGT relatively reduces the mean SER by 72.26% and 70.90%, respectively. It is also much better than FkNNC, which gives a quite high mean SER of 6.78%. Hence, ASnGT relatively reduces the average SER by 42.16%.

ASnGT does not use any vowel detection procedure, making it capable of solving many vowel ambiguities in the named-entities as described in [27] and [28]. For instance, ASnGT can easily syllabify the person names “An.dri”, “An.dry”, “An.drie”, “An.dhry”, “An.dhrie”, and “An.dhrye” by maximizing the probabilities of candidates tag-sequences without relying on the positions of the vowel-graphemes.

However, ASnGT fails to syllabify some named-entities that come from foreign languages, such as “Annabelle”, “George”, “Baha’ al-Din”. It is quite challenging to tackle this issue. One of the solutions is enlarging the dataset up to hundreds of thousands or millions of entries. Hence, in the future, the dataset will be enriched with such named-entities to increase its performance.

4. Conclusion

The 5-FCV examinations on both 50 k and 15 k datasets confirm the proposed ASnGT is significantly better than both previous BFO and CBPS models to tackle formal words and named-entities. It relatively lowers their mean SERs by up to 70.96% and 65.93%, respectively, for formal words as well as 72.26% and 70.90% for named-entities, respectively. These results are achieved since it solves both problems of distinguishing diphthongs from derivative words as well as syllabifying named-entities. Furthermore, it also significantly outperforms the previous FkNNC-based model, where it relatively lowers the mean SER by 60.20% and 42.16% for both formal words and named-entities, respectively. However, it suffers from derivative words that are confusing with absorption words as well as terms of foreign languages. It also fails to syllabify some named-entities that come from foreign languages. In the future, two improvements can be addressed by incorporating some Indonesian phonotactic constraints to solve the problems related to such
prefixes as well as enriching the dataset of named-entities with foreign names.

Declarations

Author contribution statement

Suyanto Suyanto: Conceived and designed the experiments; Contributed reagents, materials, analysis tools or data; Wrote the paper.
Andi Sunyoto; Febryanti Stievanie: Performed the experiments; Analyzed and interpreted the data.
Rezza Nafi Ismail: Performed the experiments; Wrote the paper.
Ade Romadhony: Conceived and designed the experiments; Contributed reagents, materials, analysis tools or data.

Funding statement

Prof. Dr. Suyanto Suyanto was supported by The Ministry of Education, Culture, Research, and Technology, Indonesia [KWR4.056/PNLT3/ PPM-LIT/2022, 019/SP2H/RT-JAMAK/LLA/2022, and 376/PNLT3/ PPM/2022].

Data availability statement

The data that has been used is confidential.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] C.R. Adsett, Y. Marchand, V. Kelelj, Syllabification rules versus data-driven methods in a language with low syllabic complexity: the case of Italian, Comput. Speech Lang. 23 (2009) 444–463.
[2] S. Alex, L. Mary, B. Babu, Attention and feature selection for automatic speech emotion recognition using utterance and syllable-level prosodic features, Circuits Syst. Signal Process. (2020).
[3] H. Alwi, S. Darmowidjjo, H. Lapoliwa, A.M. Moeliono, Tata Bahasa Bahaan Indonesia (The Standard Indonesian Grammar), 3 ed., Balai Pustaka, Jakarta, 2014.
[4] B.N. Budisi, Nurtono, S. Suyanto, Greedy algorithms to optimize a sentence set near-uniformly distributed on syllable units and punctuation marks, Int. J. Adv. Comput. Sci. Appl. 9 (2018) 291–296, https://theses.io/Downloads/Volume9No10/Paper.35-Greedy-Algorithms-to-Optimize-a-Sentence.pdf.
[5] S.F. Chen, J. Goodman, Empirical study of smoothing techniques for language modeling, Comput. Speech Lang. 13 (1999) 359–394.
[6] V. Doval, C. Gómez-Rodríguez, Comparing neural- and N-gram-based language models for word segmentation, J. Assoc. Inf. Sci. Technol. 70 (2019) 187–197.
[7] T. Fantaye, J. Yu, T. Hailu, Syllable-based speech recognition for a very low-resource language, Chaha, Association for Computing Machinery, 2019, pp. 415–420.
[8] P.M. Gade, S.D. Shirhadhakar, Syllable-Based Concatenative Speech Synthesis for Marathi Language, Lecture Notes in Networks and Systems, vol. 40, 2019, pp. 615–624.
[9] T.H. Hwang, Y. Mikami, Automatic syllable segmentation of Myanmar texts using finite state transducer, Int. J. Adv. ICT Emerg. Reg. 6 (2014) 2–9.
[10] J. Krantz, M. Dulin, P. De Palma, M. VanDam, Syllabification by phone categorization, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, New York, NY, USA, 2018, pp. 47–48, http://doi.acm.org/10. 1145/3205651.3208781.
[11] G. Li, L. Luo, C. Gong, S. Lv, End-to-end Tibetan speech synthesis based on phones and semi-syllables, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 1294–1297.
[12] G. Melis, C. Dyer, P. Blunsom, On the state of the art of evaluation in neural language models, in: International Conference on Learning Representations, 2018, https://openreview.net/forum?id=ByJuITgA.
[13] K. Müller, Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpus training, in: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL, 2001, pp. 410–417.
[14] K. Müller, Improving syllabification models with phonotactic knowledge, in: Proceedings of the Eighth Meeting of the Special Interest Group on Computational Phonology and Morphology - SIGPHON '06, 2006, pp. 11–20.
[15] E.A. Parande, S. Suyanto, Indonesian graphemic syllabification using a nearest neighbor classifier and recovery procedure, Int. J. Speech Technol. 22 (2019) 13–20, https://link.springer.com/article/10.1007/s11722-018-09693-9.
[16] J. Peng, Y. Zou, N. Li, D. Tao, D. Su, M. Yu, C. Zhang, D. Yu, Syllable-dependent discriminative learning for small footprint text-dependent speaker verification, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 350–357.
[17] V. Rajendran, G. Kumar, A robust syllable centric pronunciation model for tamil text to speech synthesizer, IETE J. Res. 65 (2019) 601–612.
[18] I. Ramli, N. Jamil, N. Seman, N. Ardi, An improved syllabification for a better Malay language text-to-speech synthesis (TTS), Proc. Comput. Sci. 76 (2015) 417–424.
[19] H.C. Ri, A usage of the syllable unit based on morphological statistics in Korean large vocabulary continuous speech recognition system, Int. J. Speech Technol. 22 (2019) 971–977.
[20] O. Räsänen, S. Seshadri, M. Casillas, Comparison of syllabification algorithms and training strategies for robust word count estimation across different languages and recording conditions, International Speech Communication Association, 2018, pp. 1200–1204.
[21] H. Schmid, B. Möbius, J. Weidenkaff, Tagging syllable boundaries with joint n-gram models, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 1, 2007, pp. 49–52.
[22] E. Shareghi, T. Cohn, G. Haffari, Richer interpolative smoothing based on modified Kneser-Ney language modeling, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 944–949.
[23] E. Shareghi, D. Gerz, I. Vulíč, A. Korhonen, Show some love to your n-gran: a bit of progress and stronger n-gram language modeling baselines, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4113–4118.
[24] N. Shokouhi, A. Ziaei, A. Sangwjan, J. Hansen, Robust overlapped speech detection and its application in word-count estimation for prof-life-log data, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 4724–4728.
[25] L.G. Singh, L. Laitonjam, S.R. Singh, Automatic syllabification for Manipuri language, in: The 26th International Conference on Computational Linguistics, 2016, pp. 349–357, https://www.actweb.ch/anthology/C16/C16-1034/.
[26] S. Sundara Pandiyan, P. Julian Benadit, T. Michael Mosen, An improved sub-word model based continuous robust automatic speech recognition system for tamil language, Test Eng. Manag. 82 (2020) 116–124.
[27] S. Suyanto, Flipping onsets to enhance syllabification, Int. J. Speech Technol. 22 (2019) 1031–1038, https://link.springer.com/article/10.1007/s11722-019-09649-y.
[28] S. Suyanto, Phonological similarity-based backoff smoothing to boost a bigram syllable boundary detection, Int. J. Speech Technol. 23 (2020) 191–204, https://link.springer.com/article/10.1007/s11722-020-09677-7.
[29] R. Tang, J. Lin, Progress and tradeoffs in neural language models, arXiv:1811. 00942v1, 2018.
[30] Z. Yang, Z. Dai, R. Salakhutdinov, W.W. Cohen, Breaking the softmax bottleneck: a high-rank RNN language model, in: International Conference on Learning Representations, 2018, https://openreview.net/forum?id=H1vF2ZG-CZ.
[31] C. Varra, S. Nagesh, O. Deshmukh, P. Kumar Ghosh, Noise robust speech rate estimation using signal-to-noise ratio dependent sub-band selection and peak detection strategy, J. Acoust. Soc. Am. 146 (2019) 1615–1628, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85052230229&doi=10.1122% 2f1.5124478&partnerID=40&md5=06266427343ae053c3bc6f65d3a3f53693.