Interspeech 2018
2-6 September 2018, Hyderabad

A Hybrid Approach to Grapheme to Phoneme Conversion in Assamese

Somnath Roy[1], Shakuntala Mahanta[2]

[1] Machine Learning Division, Melvault Software Solutions Pvt. Ltd., Hyderabad
[2] Department of Humanities and Social Science, IIT, Guwahati

somnath.roy@melvault.com, smahanta@iitg.ernet.in

Abstract

Assamese is one of the low resource Indian languages. This paper implements both rule-based and data-driven grapheme to phoneme (G2P) conversion systems for Assamese. The rule-based system is used as the baseline which yields a word error rate of 35.3%. The data-driven systems are implemented using state-of-the-art sequence learning techniques such as —i) Joint-Sequence Model (JSM), ii) Recurrent Neural Networks with LTSM cell (LSTM-RNN) and iii) bidirectional LSTM (BiLSTM). The BiLSTM yields the lowest WER i.e., 18.7%, which is an absolute 16.6% improvement on the baseline system. We additionally implement the rules of syllabification for Assamese. The surface output is generated in two forms namely i) phonemic sequence with syllable boundaries, and ii) only phonemic sequence. The output of BiLSTM is fed as an input to Hybrid system. The Hybrid system syllabifies the input phonemic sequences to apply the vowel harmony rules. It also applies the rules of schwa-deletion as well as some rules in which the consonants change their form in clusters. The accuracy of the Hybrid system is 17.3% which is an absolute 1.4% improvement over the BiLSTM based G2P.

Index Terms: Assemese G2P, Syllabification, LSTM-G2P, BiLSTM-G2P

1. Introduction

Grapheme to phoneme (G2P) conversion is one of the intriguing problems in natural language processing and it has attracted significant amount of attention owing to its complexities. A G2P system converts an input grapheme sequence into its corresponding phonemic sequence. In traditional approaches like joint sequence model, G2P conversion is a two step process—i) alignment of grapheme and phoneme sequence and ii) training a classifier on the alignment. The use of LSTM for G2P modeling is autonomous to the alignment as it forgoes the need of alignment. Moreover, the LSTM based G2P models produce comparably better result than conventional models [1]. The intriguing aspects of G2P are briefly mentioned below.

- The length of the grapheme sequence and phoneme sequence may not be the same during alignment, e.g., google (length=6) is aligned to ɡuːɡəl (length =5).
- The G2P mapping is not one-to-one in most of the languages, e.g., "but" becomes /bət/ and "put" becomes /put/. That is /u/ is mapped to both /ə/ and /u/
- Context plays a dominant role in determining the pronunciation, e.g., past" becomes /paːst/ and "paste" becomes /peist/.

A G2P can be developed either by using rule-based approach or data-driven approach. The rule-based approach requires the language specific rules prepared by the expert(s). The drawback of the rule-based systems are that the rules designed by experts cannot be exhaustive and sometimes the rules may clash. Therefore, rule-based systems are more prone to errors. On the other hand, data-driven processes can better capture the non-linear mappings for grapheme to phoneme conversion and are comparatively less prone to errors. Luckily, many data-driven state-of-the-art methodologies which are found for sequence-to-sequence modeling can be conveniently used for G2P conversion [1, 2, 3].

A G2P converter is one of the essential components in building a Text-to-Speech (TTS) and speech to text (STT) system. These G2P systems can be used for building a lexicon as well as for finding the pronunciation of out of vocabulary (OOV) words in TTS or STT systems. In this paper we develop various G2P converters for Assamese. The baseline G2P converter is rule-based system developed using the language specific rules designed by the second author. These rules are discussed in detail in the following section. Other G2P systems for Assamese are developed using state-of-the-art sequence modeling techniques. The bidirectional-LSTM (BiLSTM) based G2P system for Assamese is found to be most accurate. We propose a Hybrid G2P system for Assamese which correct some of the pronunciation errors generated by BiLSTM. Moreover, we also apply the rules of syllabification and the surface output is generated in two forms—i) phoneme sequence with syllable boundaries and ii) only phoneme sequences. The schematic diagram of the proposed system is shown below in Fig 1. The motivation behind the present work is the following.

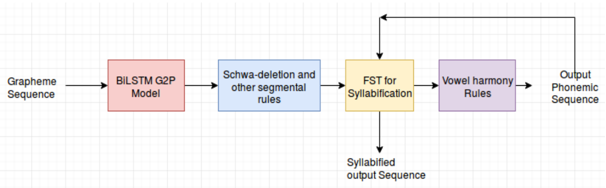

Figure 1: Schematic diagram of Hybrid G2P System for Assamese.

- Assamese is a low resource language which has not been explored enough for G2P modeling. Till date there does not exist an Assamese G2P system in

10.21437/Interspeech.2018-1694

the public domain which can be used for developing a TTS or STT for the language.

- The G2P system described in [4] is a rule-based system which uses HMM-GMM based acoustic model for obtaining the phoneme level alignment on a comparatively smaller dataset. Better phone level alignment can be produced by further feeding the HMM-GMM aligned output as an input to HMM-DNN based acoustic modeling techniques such as time-delay neural networks available in open source toolkits like Kaldi [5].
- Syllable is known to be a better unit for developing a TTS and STT in high resource languages like English and French [6, 7]. The usefulness of syllable as a unit for developing TTS and ASR for Indian languages is explored in [8, 9, 10]. Therefore, it would be beneficial to have a syllable level description for Assamese. The current work fulfills that need.

Rest of the paper is organized as follows. Section II describes the rule-based system for Assamese and the language related idiosyncrasies. Section III describes the details about the data collection and annotation and the details of G2P models trained on the hand annotated data set. Section IV describes the syllabification rules for Assamese. Section V describes the Hybrid G2P system. The result and comparison is presented in section VI. Section VII discusses conclusion and future work.

2. Rule-based G2P System for Assamese

Assamese belongs to the Indic group of 'alphasyllabaries' which is traced to the ancient Brahmi script. Alphasyllabaries represent a type of syllable based on the orthography, where the consonant is more basic and the vowel is some sort of a default or inherent vowel [11]. Assamese has eight oral vowels, at least two semi vowels and twenty three consonants [12]. Phonemic nasal vowels are present and the symbol in the register signifies the nasalised /a/ whereas the diacritic mark ঁ (also called chandrabindu) stands for nasalization across all vowel. The details of the Assamese phoneset is shown below in Table 1 and Table 2.

Schwa deletion in Assamese is quite tricky because incorrect schwa deletion may lead to different words, e.g., /parɔ/ meaning 'pigeon' and /par/ meaning the 'bank of any waterbody'. Assamese is a language which has shown to exhibit the property of vowel harmony. Vowel harmony is a process where adjacent vowels in a word change their vocalic features in order to attain similarity with the neighbouring vowels. The details of vowel harmony in Assamese can be found in [14, 15, 12]. The rules of G2P mapping is the following.

A. Akshara to Phoneme Correspondence

- Each consonant is attached with a default vowel called schwa. In Assamese it is phonetically realized as /ɔ/.
- Schwa is generally deleted in the end of a word. However, in honorific words schwa gets converted to /ɔ/, e.g., /robɔ/ (wait, honorific, 3rd person).
- Assamese also have consonant clusters, e.g., /k^{h}j/.
- The inherent schwa is deleted if the consonant has a diacritic mark hosonto.
- The consonant clusters xC becomes sC, e.g., /oboxt^{h}aːn/ becomes /obost^{h}aːn/ (station).

B. Internal (Medial) Schwa Deletion Rules

- schwa is deleted if it is followed by any other vowel.
- Schwa is not realized at the morphological boundaries.
- The syllable weight does not play any role in internal schwa deletion in Assamese unlike to Hindi as described in [16, 17].

C. Vowel Harmony Rules

- A syllable having nucleus ɔ gets transformed to /o/ if the following syllable's nucleus is /i/ or /u/ e.g., /bɔsɔr/ becomes /bosori/ (annually).
- A syllable having nucleus ɛ gets transformed to /e/ if the following syllable's nucleus is /i/ or /u/, e.g., /b^{h}ɛkʊlɑ/ becomes /b^{h}ekuli/ (frog).

3. Experiments

3.1. Data Collection and Annotation

The data was collected from various free resources available from the web, for instance the xobdo corpus of words for Assamese and words culled out of the RCILTS corpus of sentences at IIT Guwahati which were initially hand transcribed by the second author for creating the training set and test set of 34000 and 3000 words respectively.

3.2. Data-driven G2P Models

In data-driven G2P modeling the main objective is to compute the equation (1) as described below.

$$f(g) = \underset{s\epsilon S}{\operatorname{argmax}}\, p(g, s) \tag{1}$$

This implies that for a given graphemic sequence gϵ G* the output phonemic sequence sϵ S* is the one having highest joint probability. We have trained three sequence learning techniques for Assamese G2P modeling. These techniques are—i) Joint Sequence Model, ii) LSTM-RNN, and iii) BiLSTM. The Joint-Sequence Model [2] uses the unit called graphone. The graphones are nothing but the vocabulary of aligned graphemes and its corresponding phonemes. The equation (2) correctly describes the graphone (q).

$$q = (g, s)\epsilon G^{*} \times S^{*} \tag{2}$$

These graphones in joint sequence modeling are modeled for G2P conversion using n-gram models used in language modeling. The back-off distribution is determined using the Kneser-ney smoothing [18]. In our case we used sequitur toolkit which is the implementation of [2] for generating the joint sequence based G2P model. The only tunable hyperparameter in joint sequence model (JSM) is the history length. We achieved the highest accuracy for JSM using 5-gram and could not get any further improvement for n>5.

The LSTM-RNN implementation has an input layer size of 41 and output layer size of 33. The size of input

	Bilabial	Alveolar	Palatal	Velar	Glottal
Plosive	p p^h b b^h	t t^h d d^h		k k^h g g^h	
Nasal	m	n		ŋ	
Fricative		s z		x	h
Approximant		ɹ	j	w	
Lateral approximant		l			

Table 1: Assamese consonants adapted from [13]

	Front	Back	
High	i	U	+ATR
		ʊ	-ATR
Mid	e	o	+ATR
	ɛ	ɔ	-ATR
Low		ɑ	-ATR

Table 2: Assamese Vowels with Advanced Tongue Root (ATR) features adapted from [13]

Hyper-parameters	Specifications Used
No. of layers	1,2,3
Size of Model layers	64, 128, 256, 512, 1024
Learning rate	0.1 to 0.5
Learning decay factor	0.8, 0.85, 0.9
Optimizer	sgd, adam, rms-prop

Table 3: Specification of hyperparameters used in LSTM-RNN

and output layers denotes the number of graphemes and phonemes in Assamese respectively. The input layer is connected to the hidden LSTM layer and the hidden layer is connected to the output layer. The current implementation also allows multiple hidden unidirectional LSTM layers. Many possible combinations of hyper parameters such as the —i) size of model layers, ii) number of layers, iii) learning rate, iv) learning decay factor, and v) optimizer. The details are shown in Table 3. The training process becomes significantly slower by increasing the size of model layers. Therefore, we could only try the sizes 64, 128, 256, 512, and 1024 for size of model layers. Highest accuracy in LSTM-RNN is obtained using 3 layered network with the learning rate of 0.3, size of model layer as 1024, and optimizer as stochastic gradient descent (sgd)[1].
We used the BiLSTM with two forward and two backward layers similar to that reported in [1, 19] for G2P modeling. The number of nodes in hidden layer is kept constant at 256. The output layer is a connectionist temporal classifier (CTC) with a softmax error function.

4. Syllabification in Assamese

Syllabification in Assamese allows only simple syllables. The inherent schwa is very frequent between two consonants which prevents the formation of complex codas in syllables. Simple syllables implies syllables which may allow complex onset but does not allow complex coda. More clearly, the simple syllables are open syllables like V, CV, and C^*V (where asterisk denote the Kleene Closure) and closed syllables like VC, and CVC (where C and V represents the set of all consonants and all vowels in Assamese phonemic inventory respectively). A detailed description of regular expression and finite state transducer based syllabification can be found in [17]. The following regular expression describes the rules of syllabification in Assamese.

- VV → V.V
- $C^*VCV \rightarrow C^*V.\ CV$
- $C^*VCVC \rightarrow C^*V.CVC$

We developed an FST for syllabification in Assamese which is based on Thrax [20]. Two symbol tables one for consonants and another for vowels are used for building Thrax based FST. Context-dependent rewrite rules (CDRewrite) are designed for syllabification. We have also implemented the rules of syllabification in our baseline python program (which is achieved by slicing the strings and by inserting syllable mark (" . ") at syllable boundaries).

5. Hybrid System

We generated the phonemic sequence of the test set using our best performing G2P model (i.e., BiLSTM based G2P). The phonemic sequences are separated and listed out as a separate word list. The schwa deletion and other rules described in section 2 are applied on the input words. The generated output list is then passed as an input to the FST for syllabification. The vowel harmony rules are applied on the syllabified words. There are many words where we have found this process is very effective. One such example is "sbad^{h}inb^{h}awe" (with free mind). The ouput generated by BiLSTM for this word is "xObad^{h}inb^{h}awe" which is incorrect but the Hybrid system correctly captures it. This signifies the importance of the Hybrid system. The Hybrid system implemented this way achieved a WER of 17.3% which is an absolute improvement of 1.4% on the G2P system trained on BiLSTM.

[1] We would like to emphasize here that we did not get best accuracy using either adam or rms-prop optimizer. In machine learning literature it can be found that adam and rms-prop work better than sgd. Moreover, the investigation on optimizer and its performance is beyond the scope of the present work.

Systems	%WER
Baseline (Rule-based System)	35.3
Joint Sequence Model	21
LSTM-RNN	21.6
BiLSTM	18.7
Hybrid System	17.3

Table 4: Comparison of the performance of developed G2P models with the baseline (Rule-based System)

6. Results and Comparison

The performance of the systems is compared on a test set of 3000 words which is different from the 34000 words used in the training set. The word error rate (WER) for baseline system is 35.3%. The BiLSTM based G2P model is found to be superior among all the data-driven models with a WER of 18.7%. The proposed Hybrid system has a gain of 1.4% on the test dataset. The WER in our case is the total count of words with wrong pronunciation (i.e., either due to the deletion or insertion of one or more phonemes) divided by the total count of number of words in the test set. Rest of the details related to performance can be found in Table 4 shown below. To the best of our knowledge we could not find any G2P system available in public domain for Assamese to which we can compare our system. The work of [4] is completely different from the current work. Moreover, she has not described any number related to the performance of Assamese G2P system in her work.

7. Conclusion and Limitation

The current work proposes a hybrid G2P system for Assamese. The hybrid system leverages both data-driven and rule-based system and yields an absolute 1.4% improvement over the BiLSTM based data-driven G2P system.

The BiLSTM is a powerful sequence to sequence modeling technique, and the WER of 18.3% is quite high. The reason is that the BiLSTM-based G2P converter is trained on the limited amount of training data. It can be further improved by training on a significant amount of annotated data. The baseline system is not that accurate because we did not implement the rules of schwa deletion at morphological boundaries. The word hamKuri (to trip and fall) is hamOKuri at the underlying level, and the schwa /O/ at morphological boundaries is not realized in surface form. The future work may implement the rule-based G2P system autonomous to the morphological boundaries as proposed in [17].

The present work has one limitation though. The present work does not provide the performance measurement regarding error rates of our syllabification module. The testing of syllabification requires hand annotated syllabified pronunciations for the words which are under progress. In our future work, we would include the numbers on the performance of the syllabification module.

Acknowledgement

We would like to extend our appreciation to interns Prarthana Acharyya and Nilankana Das for transcribing a part of the corpus. We would also like to acknowledge the project "A parallel tagged corpus of Assamese and Bodo" awarded to Prof. Gautam Barua by DeiTy which resulted in some of the data used here. We also thank the xobdo (www.xobdo.org) team, especially Bikram Mazindar Baruah for providing us with the xobdo list of words. We would also like to thank to all anonymous reviewers for the valuable comments and suggestion.

8. References

[1] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 4225–4229.

[2] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," Speech communication, vol. 50, no. 5, pp. 434–451, 2008.

[3] S. Jiampojamarn, C. Cherry, and G. Kondrak, "Joint processing and discriminative training for letter-to-phoneme conversion," Proceedings of ACL-08: HLT, pp. 905–913, 2008.

[4] S. Sitaram, "Pronunciation modelling for low resource languages," Ph.D. dissertation, CMU, 2017.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," in IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[6] M. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," in Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80., vol. 5. IEEE, 1980, pp. 880–883.

[7] A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. R. Doddington, "Syllable-based large vocabulary continuous speech recognition," IEEE Transactions on speech and audio processing, vol. 9, no. 4, pp. 358–366, 2001.

[8] A. Lakshmi and H. A. Murthy, "A syllable based continuous speech recognizer for tamil," in Ninth International Conference on Spoken Language Processing, 2006.

[9] A. Bellur, K. B. Narayan, K. R. Krishnan, and H. A. Murthy, "Prosody modeling for syllable-based concatenative speech synthesis of hindi and tamil," in Communications (NCC), 2011 National Conference on. IEEE, 2011, pp. 1–5.

[10] H. A. Patil, T. B. Patel, N. J. Shah, H. B. Sailor, R. Krishnan, G. Kasthuri, T. Nagarajan, L. Christina, N. Kumar, V. Raghavendra et al., "A syllable-based framework for unit selection synthesis in 13 indian languages," in Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2013 International Conference. IEEE, 2013, pp. 1–8.

[11] S. Nag, "The akshara languages: what do they tell us about children's literacy learning?" Language-cognition: State of the art, pp. 291–310, 2011.

[12] S. Mahanta, "Locality in exceptions and derived environments in vowel harmony," Natural Language & Linguistic Theory, vol. 30, no. 4, pp. 1109–1146, 2012.

[13] ——, "Assamese," Journal of the International Phonetic Association, vol. 42, no. 2, pp. 217–224, 2012.

[14] ——, "On the convergence of positional markedness and positional faithfulness in vowel harmony," University of Pennsylvania Working Papers in Linguistics, vol. 13, no. 1, p. 16, 2007.

[15] ——, “Directionality and locality in vowel harmony,” Ph.D. dissertation, Utrecht University, 2007.

[16] P. Pandey and S. Roy, “A generative model of a pronunciation lexicon for hindi,” arXiv preprint arXiv:1705.02452, 2017.

[17] S. Roy, “Deriving word prosody from orthography in hindi,” in Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017). Kolkata, India: NLP Association of India, December 2017, pp. 2–12. [Online]. Available: http://www.aclweb.org/anthology/W/W17/W17-7502

[18] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” in Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, vol. 1. IEEE, 1995, pp. 181–184.

[19] R. Sproat and N. Jaitly, “Rnn approaches to text normalization: A challenge,” arXiv preprint arXiv:1611.00068, 2016.

[20] T. Tai, W. Skut, and R. Sproat, “Thrax: An open source grammar compiler built on openfst,” in IEEE Automatic Speech Recognition and Understanding Workshop, vol. 12, 2011.