GE2PE: Persian End-to-End Grapheme-to-Phoneme Conversion

Anonymous ACL submission

Abstract

Text-to-Speech (TTS) systems have made significant strides, enabling the generation of speech from grapheme sequences. However, for low-resource languages, these models still struggle to produce natural and intelligible speech. Grapheme-to-Phoneme conversion (G2P) addresses this challenge by enhancing the input sequence with phonetic information. Despite these advancements, existing G2P systems face limitations when dealing with Persian texts due to the complexity of Persian transcription. In this study, we focus on enriching resources for the Persian language. To achieve this, we introduce two novel G2P training datasets: one manually labeled and the other machine-generated. These datasets comprise over five million sentences alongside their corresponding phoneme sequences. Additionally, we propose two evaluation datasets tailored for Persian sub-tasks, including Kasre-Ezafe detection, homograph disambiguation, and handling out-of-vocabulary (OOV) words. To tackle the unique challenges of the Persian language, we develop a new sentence-level End-to-End (E2E) model leveraging a two-step training approach, as outlined in our paper, to maximize the impact of manually labeled data. The results show that our model surpasses the state-of-the-art performance by 1.86% in word error rate, 4.03% in Kasre-Ezafe detection recall, and 3.42% in homograph disambiguation accuracy.

1 Introduction

Grapheme is the smallest functional unit of a language's writing system, Phoneme is the smallest distinguishable sound unit of a language, and G2P is an important part of Text-to-Speech (TTS) and Automatic Speech Recognition (ASR) (Yolchuyeva et al., 2019a; Hasegawa-Johnson et al., 2020). E2E TTS systems using grapheme as input perform poorly on OOV words and homograph disambiguation (Huang et al., 2023); This phenomenon is more pronounced for low-resource languages. Using G2P to convert the written form of text to pronunciation form, and leveraging this form as input to TTS systems can considerably improve the intelligibility of the generated speech.

G2P is similar with the Machine Translation (MT) task except that G2P is usually done on an isolated word tokenized at a character level. As a result of this character-level tokenization, transformers have performed poorly on G2P unlike in MT. However, it is shown that the reason behind this anomaly is the lack of information while updating model parameters, and it can be resolved by increasing the batch size (Wu et al., 2021). This finding has led to high performance and efficiency in transformer-based G2P models (Yolchuyeva et al., 2019c). Following this success, knowledge transfer has been investigated through multilingual and multitask training (Zhu et al., 2022; Ploujnikov and Ravanelli, 2022), and grapheme pretraining (Dong et al., 2022). Some research has also focused on transfer learning specifically for low-resource languages (Deri and Knight, 2016) and data augmentation methods for training large models (Vesik et al., 2020).

Although in real world applications, G2P is mainly employed for achieving better performance in low-resource TTS, recent works on G2P systems mainly focus on high-resource languages like, English and pay less attention to the challenges of G2P for other languages. The Persian language (a.k.a Farsi) is a low-resource language known as one of the most challenging languages in this field (Mortensen et al., 2018; Sokolov et al., 2019; Rezaei et al., 2022) due to its unique features. Firstly, short vowels (/a/, /e/, and /o/) are not written in Persian text resulting in a lack of information while generating the phoneme sequence. Secondly, there are many homographs in Persian due to the absence of short vowels e.g., /kerm/, /kerem/, and /karam/ are identical in written form. Finally, Kasre-Ezafe, an /e/ sound connecting nouns to adjectives and descriptive nouns, is not written in Persian text.

As a result of the mentioned features, a Persian G2P system requires: morphology and phonology to predict the omitted short vowels of each noun; syntactical relations to detect Kasre-Ezafe; and semantical knowledge to disambiguate homographs. Therefore, unlike English G2P that uses words as input, Persian G2P needs a phrase-level or sentence-level input to achieve acceptable results. In this work, the main goal is to improve G2P accuracy and efficiency by employing sentence-level inputs. However, lack of data and evaluation standards appear to be the main obstacles to achieving this goal. We try to overcome these challenges by providing new training datasets and an evaluation benchmark tailored for specific features of the Persian language. In this work, we introduce:

- Four sentence-level Perian G2P datasets: machine-generated training data; manually labeled training data; manually labeled evaluation data focusing on Kasre-Ezafe; manually labeled evaluation data focusing on homographs.

- A new sentence-level Persian G2P model and a two-step training method for low-resource settings.

- A new benchmark to unify Persian G2P evaluation.

2 Related Work

The initial G2P systems used lexicons to map words to pronunciations (Kim et al., 2015). However, a comprehensive coverage of all words is not feasible as language varies over time, location, and usage domain. Therefore, rule-based methods are employed alongside lexicons to alleviate this problem (Kłosowski, 2022; Řezáčková et al., 2021; Yamasaki, 2022). Although rule-based systems address the OOV problem, they introduce new challenges: 1) designing rules requires language expertise; 2) the combination of all rules must be checked to ensure no out-of-language words are generated; 3) the rules might still not cover all words in the language (Bisani and Ney, 2008).

Labeling words with phonetic labels is much easier compared to designing rules, leading to the use of probabilistic models to predict phoneme sequences of words (Novak et al., 2012; Rao et al., 2015). With the success of Recurrent Neural Networks (RNNs) in machine translation, the community started using RNNs for G2P (Rao et al., 2015; Milde et al., 2017; Behbahani et al., 2016; Wang et al., 2023), yielding superior results compared to probabilistic models. Convolutional Neural Networks (CNNs) have also been explored to reduce computational costs and create more efficient models, yet CNNs have lower accuracy compared to RNNs (Yolchuyeva et al., 2019b; Wang et al., 2023).

Yolchuyeva et al. (2019c) show that transformers have higher accuracy compared to RNNs and CNNs while using fewer parameters. Additionally, Sun et al. (2019) demonstrate that using an ensemble of all three previously mentioned architectures and then transferring the knowledge to a smaller transformer through knowledge distillation achieves superior results. Following the introduction of transformers, attention has turned from model architecture towards other methods to enhance performance.

Data Augmentation Complex models require more training data, but generating G2P data is an expensive endeavor that requires language expertise. Data augmentation methods have been proposed to address these issues by automatically generating data (Vesik et al., 2020; Huang et al., 2023). In Vesik et al. (2020), a new set of words is collected from Wikipedia articles and converted to phoneme sequences (silver labels) using a model trained on manually labeled data (golden labels). In the second step, the model is trained on a combination of silver and gold labels. Contrary to expectations that data augmentation should decrease the error rate, it actually has reverse results. Ryan and Hulden (2020) use recurrent subwords with unchanging pronunciations in the data and concatenate them to create new words for training. This method results in consistent error rate decrease for extremely low-resource settings with 500 or fewer words. However, this is not the case for languages with more training data.

Multilingual and Transfer Learning (Milde et al., 2017; Vesik et al., 2020; Zhu et al., 2022) use multilingual training to reduce G2P errors. Milde et al. (2017) use bilingual English-German training resulting in better performance for English but worse performance for German. Vesik et al. (2020) use multilingual training on 15 languages and show that language similarity can positively affect results, but similar alphabet (script) does not affect the knowledge transfer. Zhu et al. (2022) demonstrate that massively multilingual models trained on 99 languages can perform as well as unilingual ones. They further explore the effect of the level of tokenization in G2P and find that character-level tokenization performs better compared to subword-level models. Furthermore, they showe that using the multilingual model as a starting point to train on a new language performs better compared to a model pretrained on masked language modeling (MLM).

Similar to Zhu et al. (2022), Dong et al. (2022) and Řezáčková et al. (2021) explore MLM pretraining for G2P models. In Řezáčková et al. (2021), subword-level MLM pretraining on sentences is done before subword-level G2P training, resulting in lower error rates compared to RNN-based G2P. Dong et al. (2022) train BERT on character-level MLM for isolated words. The resulting BERT model is once used as the encoder of a transformer-based G2P model, and in another instance, BERT embeddings are combined with the encoder's self-attention and decoder's encoder-decoder attention. It is shown that for medium-resource languages, fusing BERT embeddings in attention has the best performance, and for low-resource languages, BERT as encoder performs best.

Another approach to transfer learning is multi-task training, explored by Ploujnikov and Ravanelli (2022) and Wang et al. (2021), where a combination of G2P with homograph disambiguation and grapheme-phoneme alignment is used respectively to train G2P models, leading to better performance on English G2P compared to RNN-based G2P. Deri and Knight (2016); Peters et al. (2017); Li et al. (2022) strive to adapt high-resource G2P models for low-resource languages. Deri and Knight (2016) have collected G2P data for 85 high-resource languages and 229 low-resource languages, where low-resource data is only used for evaluation. They define lang2lang and phone2phone metrics to measure linguistic and phonetic distance between languages, and for each low-resource language, the nearest high-resource language is used to create a model for the respective low-resource language. The adaptation is done in two ways: adapting the output and adapting the training data using the phone2phone metric to find the nearest high-resource phoneme to each low-resource phoneme.

Peters et al. (2017) use the data introduced by Deri and Knight (2016) to train a multilingual model on all high-resource and low-resource languages by adding a prefix to the input indicating the language. Their results show improvement on low-resource languages but not on high-resource languages compared to the previous work. They also investigated model embeddings and mention there is considerable alignment between phoneme embeddings and the phone2phone metric. However, there is no correlation between language prefix embeddings and the lang2lang metric, meaning generalizing the multilingual model to new languages using the prefix embedding won't be an option. Li et al. (2022) train a model for each of 260 languages that had enough training data. Then for each of the 600 low-resource languages, an ensemble of k nearest languages is used, where the nearest languages are found based on the language family tree. The results show an improvement in error rates compared to models trained on English-only, multilingual, and nearest language data.

Context-based Models For many languages like Chinese, one of the main challenges of sentence-level G2P is homograph disambiguation. Previous works have attempted to incorporate context in their models to overcome the homograph disambiguation challenge. For instance, Kim et al. (2023) use a window of the input for Chinese G2P. Řezáčková et al. (2021), Huang et al. (2023), and Ploujnikov and Ravanelli (2022) use sentence-level input for English G2P. In addition, Rezaei et al. (2022) and Behbahani et al. (2016) use context at the phrase and sentence levels, respectively, for Persian G2P. Furthermore, Zhao et al. (2022) employ context embedding in transformer-based G2P to reduce output errors caused by typos in the input.

3 Persian Language

Persian, an Indo-European language, uses the Arabic script, which originates from the Semitic language family with a vastly different phonetic system. This leads to inconsistencies between the written and spoken forms of Persian, resulting in a lack of orthographic transparency. Orthographic transparency is achieved when each grapheme corresponds to one and only one phoneme, and vice versa (Miangah and Vulanovic, 2021). In Persian, each consonant can be represented by up to four different graphemes, and given that short vowels are typically not written, each grapheme can correspond to up to four different pronunciations. Consequently, to manage this complexity and enable Persian G2P, the task is divided into three subtasks: OOV G2P, Kasre-Ezafe detection, and homograph disambiguation.

3.1 OOV G2P

In this task, the goal is to predict the phoneme sequence of new words not seen in the training data. Namnabat and Homayounpour (2006) employ a combination of neural networks and rule-based systems to perform this task using a modified version of the FarsDat data (further explained in Section 4.1). Behbahani et al. (2016) and Rezaei et al. (2022) use RNN and transformer models, respectively, on their own modified versions of FarsDat to perform OOV G2P.

3.2 Kasre-Ezafe Detection

From a grammatical perspective, Kasre-Ezafe is a feature that connects words in the noun group, adjective group, and prepositional group, thereby creating larger structures within the hierarchical structure of a sentence (Bijankhan, 2006). Although Kasre-Ezafe lacks intrinsic meaning, it significantly influences the syntactical relations and semantics of a sentence. With the introduction of Peykare (Bijankhan et al., 2011), a Part-of-Speech (POS) tagging dataset that includes an exclusive label for Kasre-Ezafe, many studies have focused on detecting Kasre-Ezafe as a binary classification task, which can be considered a subtask of POS tagging. Methods used for this binary classification include Classification and Regression Tree (CART) (Koochari et al., 2006), genetic algorithms (Shamsfard and Noferesti, 2014), Maximum Entropy (ME), Conditional Random Field (CRF), Statistical Machine Translation (SMT) (Asghari et al., 2014), RNNs based on gated recurrent units (Rezaei et al., 2022) and long short-term memory, CNNs, BERT, and XLMRoBERTa (Doostmohammadi et al., 2020).

3.3 Homograph Disambiguation

An important aspect of Persian natural language processing involves understanding the morphological, phonological, syntactical, and semantical relations among words (Bijankhan and Moradzade, 2004). Based on these relations, three categories of words are defined: 1) homonyms, which have the same written and spoken form but different meanings; 2) homophones, which have different written forms and meanings but similar pronunciation; and 3) homographs, which are written the same but have different meanings and pronunciations (these words may share the same POS tag or not). Additionally, there are Persian words that can be read with different pronunciations without changing their meaning, though the tone of speaking changes considerably. In TTS and G2P systems, accurately identifying the correct spoken form of these words and homographs based on context is essential for generating natural and intelligible output. Rezaei et al. (2022) employ an RNN-based model to perform homograph disambiguation on homograph words that take different POS tags; This is the only work on Persian homograph disambiguation.

3.4 Discussion

Although previous works on OOV G2P have modified and used the FarsDat data for training and evaluating their proposed methods, none of these works have published their datasets. This has led to a lack of resources for training Persian G2P models and the absence of a benchmark for comparing these methods. A similar issue exists in homograph disambiguation, as there has not been any publicly available data for this task in the Persian language. For Kasre-Ezafe detection, the introduction of Peykare provided a foundation for research. However, not all studies use the same proportion of Peykare for evaluating their models, making it difficult to compare their results. Furthermore, although the proposed models have achieved over 99% accuracy on Peykare, they still struggle to provide high-quality output in real-world applications.

Another unaddressed issue in Persian G2P is that the previously explored subtasks overlap significantly. To solve these subtasks, the model needs to reach an understanding of the language on different levels. According to Tenney et al. (2019), Language Models (LMs) exhibit signs of syntactical understanding in lower layers and semantical understanding in higher layers. Therefore, we argue that although each of these subtasks requires a specific level of language understanding, training an LM to address all tasks in a multitask manner might improve performance on all tasks. This is because they are highly correlated and unlikely to interfere with each other's training. Furthermore, a single E2E model is more parameter-efficient and easier to tune and train compared to a multi-module model that has a specific model for each subtask.

Dataset	Sentences	Unique Words	Avg. Word/Sent.	Avg. Char/Sent.
machine generated	5,375,235	1,054,620	25.26	126.46
farsdat aligned	909	4,954	28.12	144.28
kasre eval	257	1,624	12.79	65.20
homograph eval	269	1,667	13.40	63.24

Table 1: Statistics of the proposed datasets, including number of sentences, number of unique words, Average word per sentence and average character per sentence.

4 Datasets

To address the issues discussed in Section 3.4, we propose two datasets for training Persian G2P at the sentence level, aiming to overcome all mentioned challenges using a single LM. These datasets include a manually labeled dataset ("farsdat aligned") and an automatically labeled dataset ("machine generated"). Additionally, we propose two evaluation datasets, "homograph eval" and "kasre eval", to benchmark Persian G2P models. "homograph eval" consists of challenging sentences that include homographs, while "kasre eval" contains challenging sentences featuring Kasre-Ezafe. Statistics and data samples for all proposed datasets are available in Table 1 and Table 6 respectively.

4.1 FarsDat Aligned

FarsDat (Bijankhan et al., 1994) is an ASR dataset where the recorded speech of all participants is accompanied by phoneme labels generated by language experts. Although FarsDat can be a great source for Persian G2P, the transcripts are not cross-checked with the speech, and the phoneme sequence is generated based on participants' utterances, leading to misalignment between the grapheme and phoneme sequences. Additionally, participants come from different regions of Iran with varying accents, resulting in inconsistencies in word pronunciation. Furthermore, some of the texts read by participants require college-level reading, which not all participants can properly handle.

In response, utterances of five participants with Tehrani accents and college-level or higher education were chosen to create a G2P dataset. First, each sentence of the transcripts was aligned with its phoneme sequence. If a full sentence was skipped by the participant, it was removed from the transcript. We then examined the words and modified the phoneme sequences if a word was mispronounced or a completely different word was pronounced instead. Furthermore, all words ending with Kasre-Ezafe were labeled with the token "1" added to the end of their phoneme sequence. This token serves as an indicator of Kasre-Ezafe occurrence and distinguishes such words from those that naturally end with the /e/ phoneme.

4.2 Machine Generated

We used "farsdat aligned" to train a sentence-level G2P model, with the results available in Appendix A indicating that the data was insufficient to train a Persian G2P model. Therefore, following Vesik et al. (2020), we augmented the data using existing G2P models and used "farsdat aligned" for model tuning. Furthermore, G2P models are sensitive to data domain (for more information on G2P data size and domain, refer to the pilot experiments in Appendix A). Therefore, to provide a corpus that covers both formal and informal versions of contemporary Persian, we sampled text from Peykare (Bijankhan et al., 2011), Miras (Sabeti et al., 2018), and Naab (Sabouri et al., 2022) including five million sentences after removing duplicates. Before generating phoneme sequences for each sentence, the sampled text was cleaned using the pre-processing script introduced by Sabouri et al. (2022), and the results were normalized using Parsivar[1] to reduce the error rate during automatic phoneme sequence generation. Finally, the best current G2P model introduced by Rezaei et al. (2022) was used to generate phoneme sequences for the sampled sentences. This model also generates "1" for words ending with Kasre-Ezafe.

4.3 Evaluation Data

To benchmark Persian G2P models regarding all existing challenges, we provide two evaluation datasets, "homograph eval" and "kasre eval" containing challenging cases of homograph disambiguation and Kasre-Ezafe detection, respectively. The challenging test cases include sentences that

[1] https://github.com/ICTRC/Parsivar

previous G2P models failed to predict accurately in addition to sentences that are hard for humans to correctly read at first glance. All words that have homographs are labeled with the token "2," and all words ending with Kasre-Ezafe are labeled with the token "1" in the phoneme sequence. As a result, in addition to evaluating G2P models based on their error rate in OOV G2P, we can also assess their performance in Kasre-Ezafe detection and homograph disambiguation.

5 Experimental Setup

To address the challenges previously discussed and provide a Persian end-to-end G2P model (GE2PE), we propose a byte-level transformer with one sentence as input. To mitigate the lack of data resources during training, we implement a two-step training process that optimizes the use of manually labeled data ("farsdat aligned"). In the following sections, we offer detailed explanations of our model architecture, baselines, proposed training methods, and evaluation metrics.

5.1 Models

Following Zhu et al. (2022), we use ByT5 (Xue et al., 2022), a text-to-text transformer with input tokenized at the byte level. The byte level tokenization makes the model flexible enough to handle new words which frequently occurs in low resource G2P. To be able to train a single model on all Persian G2P subtasks, context is needed. Therefore, instead of using isolated words as input, similar to Řezáčková et al. (2021), we use a complete sentence as input. Considering the lack of data and computational resources, the number of blocks in the encoder and decoder of ByT5 is reduced to two in each. We tried other transformer architectures as well which results can be found in our pilot experiments in Appendix A.

The proposed model is compared to the state-of-the-art Persian G2P model (Rezaei et al., 2022) which uses a 4x4 transformer on words for OOV, and two GRU networks on a window of five words for Kasre-Ezafe detection and homograph disambiguation. Their model is trained on all FarsDat data (100 participants) modified by authors including 42,000 sentences and one million words. We also compare our model with the best version of Persian G2P (ByT5-small) among the multilingual and monolingual models provided by Zhu et al. (2022).

5.2 Training Method

Similar to Vesik et al. (2020), we first combined the two proposed datasets, "farsdat aligned" and "machine generated", using the best ratio (manually labeled:machine generated = 1:4) proposed by Fadaee and Monz (2018). However, the model's output was not intelligible until we reached a ratio of 1:20. At this ratio, the model repeated the frequent errors present in the "machine generated" data and no improvement based on "farstdat aligned" was observed (output samples in Appendix A). This outcome aligned with the findings of Vesik et al. (2020), where using silver labels mixed with gold labels resulted in worse performance.

To maximize the effect of "farsdat aligned" and reduce the errors caused by the noise in "machine generated", we take insight from Ratle et al. (2010), and first train the model on "machine generated" data, then finetune it on "farsdat aligned." To avoid overfitting on noisy data, since "machine generated" contains errors in phoneme sequences, we use the "farsdat aligned" validation set during the first training step. This way, training can be stopped as soon as the model starts learning the noise.

5.3 Evaluation Metrics

Phoneme Error Rate (PER) and Word Error Rate (WER) are the two metrics used in G2P evaluation. In PER, the Levenshtein distance is calculated at the character level, while in WER, the same distance is calculated at the word level. If the number of substitutions, insertions, and deletions are denoted as S, I, and D respectively, and the number of reference phonemes (or words for WER) is represented by N, then the error rate is calculated as:

$$ErrorRate = \frac{S + I + D}{N} \quad (1)$$

In addition to these metrics, we use the "1" token to identify words ending with Kasre-Ezafe. Considering the low frequency of these words, we calculate recall and precision to evaluate the model's ability to detect Kasre-Ezafe. For evaluating the model's performance on homograph disambiguation, we first minimize the Levenshtein distance to find a word-level alignment between the reference phoneme sequence and the predicted phoneme sequence. Then, based on the "2" tokens, homographs are identified, and accuracy in homograph disambiguation is reported as the ratio of homo-

Model	PER%	WER%
silver GE2PE	3.75	17.97
GE2PE	**2.92**	**14.83**
(Rezaei et al., 2022)	2.96	16.69

Table 2: average of PER and WER on both “kasre eval” and “homograph eval” datasets.

Model	Kasre-Ezafe		Homograph
	Rec.%	Prec.%	Acc.%
GE2PE	**73.93**	**74.97**	**61.86**
(Rezaei et al., 2022)	69.90	69.72	58.44

Table 3: Kasre-Ezafe detection and homograph disambiguation results based on “kasre eval” and “homograph eval” datasets.

Model	Original		Modified	
	PER	WER	PER	WER
silver GE2PE	**7.02**	**32.20**	**5.17**	**24.00**
GE2PE	9.04	36.00	7.19	28.40
(Zhu et al., 2022)	12.28	51.20	-	-

Table 4: PER and WER on original and modified versions of Zhu et al. (2022)’s test set.

graphs that were predicted correctly, where "correctly" means having zero PER.

6 Results

In the first experiment, we compare our proposed model to the multi-module model introduced by Rezaei et al. (2022) on the “kasre eval” and “homograph eval” datasets. In the second experiment, we compare our proposed model to the multi-lingual model presented by Zhu et al. (2022) using the test set provided in their paper[2]. This comparison is because the multi-lingual model is trained solely on isolated words and is not capable of processing sentence-level Persian inputs.

To assess the effectiveness of our training method in maximizing the impact of manually labeled data, we calculated PER and WER for both evaluation datasets in the first experiment. The results, summarized in Table 2, indicate that our two-step training approach not only surpasses the silver GE2PE model (the model solely trained on “machine generated”) but also outperforms the multi-module model. It is notable that our proposed model has only one-sixth of the parameters of the multi-module model and was trained on just 900 manually labeled sentences.

Table 3 presents the evaluation results for Kasre-Ezafe detection and homograph disambiguation. The results show improvements in both tasks compared to the multi-module model. Specifically, some sentences in the “kasre eval” dataset require the entire sentence context for accurate Kasre-Ezafe detection, whereas the multi-module model uses only a five-word window. This broader context utilization likely contributes to our model’s superior performance in this task.

Unlike Kasre-Ezafe detection, there is no explicit token in the phoneme sequence of the training data to indicate the occurrence of homographs. Thus, our model was not explicitly trained for homograph disambiguation. Nevertheless, the language understanding gained through the G2P training process appears to enhance its performance in this task.

PER and WER are reported on Zhu et al. (2022)’s original test set for the multi-lingual baseline, silver GE2PE, and GE2PE models in Table 4. Although both versions of our proposed model outperform the baseline, the error rates are much higher compared to previous test sets, and surprisingly, silver GE2PE performs better than GE2PE. To better understand this phenomenon, we examined frequent errors for these models. Interestingly, the most frequent error occurred with words starting with a vowel in their phoneme sequence. However, no syllable can start with a vowel in the Persian language. Therefore, we modified the data and addressed this issue by adding the / ’/ consonant to the start of the phoneme sequence for all words starting with a vowel. The error rates on the modified test set are reported in Table 4.

After addressing this issue, we compared the frequent errors of silver GE2PE and GE2PE, with samples of this comparison found in Table 5. Five categories of errors were identified in the outputs: 1) wrong short vowel prediction, 2) correct prediction but erroneous data, 3) late stop-token generation (only in GE2PE), 4) generating /’i/ instead of /yi/ (only in GE2PE), and 5) wrong Kasre-Ezafe generation (only in silver GE2PE).

The main reasons GE2PE performed worse than

[2] https://github.com/lingjzhu/CharsiuG2P/blob/main/data/test/fas.tsv

Source	Error Samples		
	shared	GE2PE	silver GE2PE
Grapheme Data	برون، دوحه، درگر	چه، ترسایی، غش‌غش	ترب، گورخر، کف‌گیری
Phoneme Data	dorg/r, duhe, berun	q/$q/$, t/rsayi, ce	k/fgiri, gurex/r, torob
silver GE2PE	d/rgar, dohe, borun	qeqe, t/rsayi, ce	k/fegiri, gurx/r, torb
GE2PE	d/rg/r, dohe, borun	qeqeqe$, tarsa@i, cece	k/fgiri, gurex/r, torob

Table 5: Error samples occurring in experiments using Zhu et al. (2022)'s test set, categorized based on their occurrence in silver GE2PE and GE2PE outputs.

silver GE2PE were errors 3 and 4, caused by "farsdat aligned" features. This dataset contains only long sentences, which biases the model towards longer outputs and delays the generation of the stop token. This can be mitigated by including isolated words and short sentences in the training data. Furthermore, two consecutive "y" in grapheme can be read as /yi/ or /'i/, but the latter is the old Persian standard used in FarsDat, while the former is the modern standard. This error can be corrected by editing "farsdat aligned" to follow modern Persian standards. Another significant issue is type 2 errors, which highlight the low quality of the only available public Persian G2P resource.

7 Conclusion

With the recent growth of high-resource TTS systems, the G2P module has been removed from the pipelines, and speech has been generated using graphemes in an E2E manner. However, phonemes are still needed to generate natural and intelligible speech for low-resource languages. Although G2P is mainly used for these languages in real world applications, little work has been done on low-resource G2P. In this work, we emphasized the need for new data resources and conversion approaches for Persian, a low-resource language, and provided new datasets for training and evaluating Persian G2P with regard to three important Persian G2P challenges: OOV, Kasre-Ezafe detection, and homograph disambiguation. Additionally, a new E2E model was introduced to address these Persian G2P challenges and serve as a baseline for the newly proposed datasets.

Although using the proposed data, model, and training method led to state-of-the-art results in OOV, Kasre-Ezafe detection, and homograph disambiguation, there is still room for improvement. The current work uses maximum likelihood loss to train the model for all tasks. However, adding a task-specific loss for Kasre-Ezafe detection can further improve the results. Future work can also focus on augmenting data for homograph disambiguation and using task-specific loss for homograph disambiguation as well. These enhancements can further improve the results of the two tasks without any changes to the model architecture or training procedure.

Limitations

FarsDat is a valuable resource for providing gold labels for the G2P task. However, in this study, we were only able to modify the data of five participants with the Tehrani accent. Modifying the data of all 100 participants would not only enhance the current model's output quality but also enable the development of G2P models for various Iranian accents of Persian.

Furthermore, we did not apply any specific loss function for each task during training, relying instead on the additional tokens added for Kasre-Ezafe. Although these tokens might implicitly train the model on different tasks, an explicit training method could yield better results. Additionally, due to limited computational resources, we were unable to test other architectures for the defined multi-task objective.

It is also important to note that low PER and WER and high accuracy in Kasre-Ezafe detection and homograph disambiguation do not guarantee the intelligibility of the output. For example, if one phoneme of a word is generated incorrectly, the audience might still infer the intended word based on the remaining phonemes or the context, or they might interpret it as an entirely different word or meaning. The quality and usability of these systems can only be accurately assessed when used in a TTS pipeline in practice.

References

Habibollah Asghari, Jalal Maleki, and Heshaam Faili. 2014. A probabilistic approach to Persian ezafe recognition. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 138–142, Gothenburg, Sweden. Association for Computational Linguistics.

Yasser Mohseni Behbahani, Bagher Babaali, and Mussa Turdalyuly. 2016. Persian sentences to phoneme sequences conversion based on recurrent neural networks. *Open Computer Science*, 6(1):219–225.

Mahmood Bijankhan. 2006. Feasibility for analyzing the kasre ezafe of persian language with pattern matching method. *Research Institute of Culture, Art and Communication - Research Institute of Communication - Department of Persian Language and Information Technology*.

Mahmood Bijankhan and Shahrooz Moradzade. 2004. Homographs in persian transcript. *Collection of lectures, reports and abstracts of projects in the first Persian language and computer research workshop, University of Tehran*, pages 53–63.

Mahmood Bijankhan, Javad Sheikhzadegan, and Mahmood R Roohani. 1994. Farsdat-the speech database of farsi spoken language. proccedings australian conference on speech science and technology.

Mahmood Bijankhan, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. 2011. Lessons from building a persian written corpus: Peykare. *Language Resources and Evaluation*, 45(2):143–164.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408, Berlin, Germany. Association for Computational Linguistics.

Lu Dong, Zhi-Qiang Guo, Chao-Hong Tan, Ya-Jun Hu, Yuan Jiang, and Zhen-Hua Ling. 2022. Neural grapheme-to-phoneme conversion with pre-trained grapheme models. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6202–6206.

Ehsan Doostmohammadi, Minoo Nassajian, and Adel Rahimi. 2020. Persian ezafe recognition using transformers and its role in part-of-speech tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 961–971, Online. Association for Computational Linguistics.

Marzieh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium. Association for Computational Linguistics.

Mark Hasegawa-Johnson, Leanne Rolston, Camille Goudeseune, Gina-Anne Levow, and Katrin Kirchhoff. 2020. Grapheme-to-phoneme transduction for cross-language asr. In *Statistical Language and Speech Processing*, pages 3–19, Cham. Springer International Publishing.

Jocelyn Huang, Evelina Bakhturina, and Oktai Tatanov. 2023. Automatic heteronym resolution pipeline using rad-tts aligners. *arXiv preprint arXiv:2302.14523*.

Jungjun Kim, Changjin Han, Gyuhyeon Nam, and Gyeongsu Chae. 2023. Good neighbors are all you need for chinese grapheme-to-phoneme conversion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Nam Kyun Kim, Woo Kyeong Seong, and H. K. Kim. 2015. *Lexicon Optimization for WFST-Based Speech Recognition Using Acoustic Distance Based Confusability Measure and G2P Conversion*, pages 119–127. Springer International Publishing, Cham.

Piotr Kłosowski. 2022. A rule-based grapheme-to-phoneme conversion system. *Applied Sciences*, 12(5).

Abbas Koochari, Behrang QasemiZadeh, and Mojtaba Kasaeiyan. 2006. Ezafe prediction in phrases of farsi using cart. In *Proceedings of the I International Conference on Multidisciplinary Information Sciences and Technologies*, pages 329–332.

Xinjian Li, Florian Metze, David Mortensen, Shinji Watanabe, and Alan Black. 2022. Zero-shot learning for grapheme to phoneme conversion with language ensemble. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2106–2115, Dublin, Ireland. Association for Computational Linguistics.

Tayebeh Mosavi Miangah and Relja Vulanovic. 2021. The ambiguity of the relations between graphemes and phonemes in the persian orthographic system. *Glottometrics*, 50:9–26.

Benjamin Milde, Christoph Schmidt, and Joachim Köhler. 2017. Multitask Sequence-to-Sequence Models for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2017*, pages 2536–2540.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

M. Namnabat and M. M. Homayounpour. 2006. A letter to sound system for farsi language using neural networks. In *2006 8th international Conference on Signal Processing*, volume 1.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49, Donostia–San Sebastián. Association for Computational Linguistics.

Ben Peters, Jon Dehdari, and Josef van Genabith. 2017. Massively multilingual neural grapheme-to-phoneme conversion. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 19–26, Copenhagen, Denmark. Association for Computational Linguistics.

Artem Ploujnikov and Mirco Ravanelli. 2022. SoundChoice: Grapheme-to-Phoneme Models with Semantic Disambiguation. In *Proc. Interspeech 2022*, pages 486–490.

Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229.

Frédéric Ratle, Gustavo Camps-Valls, and Jason Weston. 2010. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2271–2282.

Mahdi Rezaei, Negar Nayeri, Saeed Farzi, and Hossein Sameti. 2022. Multi-module g2p converter for persian focusing on relations between words. *arXiv preprint arXiv:2208.01371*.

Zach Ryan and Mans Hulden. 2020. Data augmentation for transformer-based G2P. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 184–188, Online. Association for Computational Linguistics.

Behnam Sabeti, Hossein Abedi Firouzjaee, Ali Janalizadeh Choobbasti, S.H.E. Mortazavi Najafabadi, and Amir Vaheb. 2018. MirasText: An automatically generated text corpus for Persian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sadra Sabouri, Elnaz Rahmati, Soroush Gooran, and Hossein Sameti. 2022. naab: A ready-to-use plug-and-play corpus for farsi. *arXiv preprint arXiv:2208.13486*.

Mehrnoush Shamsfard and Samira Noferesti. 2014. A hybrid algorithm for recognizing the position of ezafe constructions in persian texts. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2(6):17–25.

Alex Sokolov, Tracy Rohlin, and Ariya Rastrow. 2019. Neural Machine Translation for Multilingual Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2019*, pages 2065–2069.

Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2019*, pages 2115–2119.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Kaili Vesik, Muhammad Abdul-Mageed, and Miikka Silfverberg. 2020. One model to pronounce them all: Multilingual grapheme-to-phoneme conversion with a transformer ensemble. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 146–152, Online. Association for Computational Linguistics.

Chunfeng Wang, Peisong Huang, Yuxiang Zou, Haoyu Zhang, Shichao Liu, Xiang Yin, and Zejun Ma. 2023. Liteg2p: A fast, light and high accuracy model for grapheme-to-phoneme conversion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yonghe Wang, Feilong Bao, Hui Zhang, and Guanglai Gao. 2021. Joint alignment learning-attention based model for grapheme-to-phoneme conversion. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7788–7792.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Tomohiro Yamasaki. 2022. Grapheme-to-phoneme conversion for Thai using neural regression models. In *Proceedings of the 2022 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4251–4255, Seattle, United States. Association for Computational Linguistics.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019a. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9(6).

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019b. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9(6).

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019c. Transformer Based Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2019*, pages 2095–2099.

Chendong Zhao, Jianzong Wang, Xiaoyang Qu, Haoqian Wang, and Jing Xiao. 2022. r-g2p: Evaluating and enhancing robustness of grapheme to phoneme conversion by controlled noise introducing and contextual information incorporation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6197–6201. IEEE.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. ByT5 model for massively multilingual grapheme-to-phoneme conversion. In *Proc. Interspeech 2022*, pages 446–450.

Markéta Řezáčková, Jan Švec, and Daniel Tihelka. 2021. T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2021*, pages 6–10.

A Pilot Experiments

A.1 Data Experiments

We first trained the 2x2 ByT5 transformer on “farsdat aligned”. However, the output was completely irrelevant to the input (e.g., "@/z @/ran @/san @/n @/n @/san @/n @/san @/n @/..."). In the second attempt, the same model was trained on 4,000 sentences from the Miras corpus combined with “farsdat aligned”, but the same results were observed (e.g., " h/mcen m/re1 m/re1 m/re1 m/re1 m/re1 mare1 ..."). Finally, with 20,000 sentences from Miras and 10 epochs of training, we were able to generate reasonable outputs. The PER and WER on the validation set of the machine-generated data were 2.7% and 7.8%, respectively.

To evaluate the quality of the machine-generated data, we tested the model on in-domain (News) and out-of-domain data. Interestingly, the model could not generate the stop token in time for Persian poems and literary text where standard grammatical rules are not followed (e.g., the verb can appear anywhere in the sentence instead of at the end). As a result, we decided to sample data from multiple sources (Miras, Peykare, Naab) with different styles (News, history, literary, etc.). Another observation was that among multiple characters used for each Persian grapheme, the multi-module model (Rezaei et al., 2022) used for generating the machine-generated data recognized only one of the characters and ignored any other character appearing in the input. Furthermore, the multi-module model falsely generated Kasre-Ezafe when space was used instead of half-space. Therefore, we added text normalization to our preprocessing pipeline to ensure the highest quality output using the multi-module model.

A.2 Model Architecture

Due to a lack of computational resources, we ran the experiment for only two architectures, 3x1 and 2x2 ByT5 transformers. The 2x2 architecture performed well, as reported in section 6. However, in the 3x1 configuration, although all the words in the phoneme sequence were valid Persian words, they were completely irrelevant to the input (e.g., "@/mma @in ra b/raye mixah/m bud v/ nohs/d v/ @..." instead of "midanim hoquqe1 to boxor n/mir @/st @/mma d/r @iran beman"). This could be due to using only one block in the decoder. As a result, we chose to use the 2x2 architecture.

A.3 Implementation Details

We used a ByT5 model with 2 encoder blocks and 2 decoder blocks. The input and output sizes are 512 tokens, and the number of neurons in the feed-forward network is 512. There are 6 attention heads, and the size of vectors in the attention mechanism is 64. The training batch size is set to 25 with gradient accumulation equal to 2. The initial learning rate is set to 5e-4 with a cosine learning rate scheduler. The number of beams during inference is set to 5 for beam search. All experiments were run using Kaggle cloud resources (P100 GPU and 12 gigabytes of RAM) with the random seed equal to 1625. We were only able to run the experiment once due to lack of resources (each experiment takes 30 to 40 hours). All datasets used in this work are public datasets and the multi-module model (Rezaei et al., 2022) was used with the consent of authors.

Dataset	Sample
machine generated	و من هرگاه به سال‌هایی که هنوز در پیش روی ما است می‌اندیشم به سال‌های رشد و کشف دو جانبه نقاط ناشناخته و آن روزهای بزرگ به ناگاه قصر قدیمی دانلری در نظرم بسیار درخشان جلوه می‌کند و احساس می‌کنم زن خوشبختی هستم. v/ m/n h/rgah be salhayi ke h/nuz d/r pi$e1 ruye1 ma @/st mi@/ndi$/m be salhaye1 ro$d v/ k/$fe1 do janebeye1 noqate1 na$enaxte v/ @an ruzhaye1 bozorg be nagah q/sre1 q/dimiye1 danl/ri d/r n/z/r/m besiyar der/x$an jelve mikon/d v/ @ehsas mikon/m z/ne1 xo$b/xti h/st/m
farsdat align	اشاره ، پنجاهمین سالگرد تاسیس سازمان پیمان آتلانتیک شمالی ، ناتو ، در ماه آوریل هزار وَ نُهصد وَ نود وَ نُه با شرکت سران کشورهای عضو برگزار شد. @e$are p/njahomin salg/rde1 t/@sise1 sazemane1 peymane1 @atlantike1 $omali nato d/r mahe1 @avrile1 hezar v/ nohs/d v/ n/v/d v/ noh ba $erk/te1 s/rane1 ke$v/rhaye1 @ozv b/rgozar $od
kasre eval	آن مرد روزهای سخت پاییز عازم جنگ بین ایران و عراق شد. @an m/rde1 ruzhaye1 s/xt payiz @azeme1 j/nge1 beyne1 @iran v/ @/raq $od
homograph eval	قبل از خرید دستگاه بخور ، باید بدانید که آن را به چه منظور می‌خواهید تهیه کنید. q/bl @/z x/ride1 d/stgahe1 boxur2 bay/d bedanid ke @an ra be ce m/nzur mixahid t/hiyye konid

Table 6: Samples of the proposed datasets, grapheme sequences and their corresponding phoneme sequence.