Syllabic-Structure Decoder for Automatic Speech Recognition in Vietnamese

Anonymous ACL submission

Abstract

Most Automatic Speech Recognition (ASR) systems formulate transcription as a prediction problem over orthographic units such as characters, subwords, or words. Although effective, such representations do not explicitly reflect the phonetic structure of speech and often require large vocabularies to maintain adequate coverage. In this work, we are motivated from the phonemic features of Vietnamese to propose a Syllabic-Structure Decoder for ASR, which models speech at the phoneme level instead of the orthographic level. Our approach explicitly captures the phonological composition of syllables, enabling the decoder to generate valid syllabic structures from a compact phonemic inventory. This design more closely aligns with the phonetic realization of speech while significantly reducing vocabulary size. Experimental results on two benchmarks: LSVSC, representing standard speech, and UIT-ViMD, a multi-dialect corpus containing diverse regional pronunciations, show that our method consistently outperforms strong previous baselines, especially pretrained baselines such as PhoWhisper and Wav2Vec2, despite using a substantially smaller vocabulary and no additional training resources. These results highlight the effectiveness of phoneme-based syllabic modeling for ASR in this language. Code for experimental reproducibility will be publicly available upon the acceptance of this paper.

1 Introduction

Automatic Speech Recognition (ASR) has achieved remarkable progress in recent years with the development of end-to-end neural architectures and large-scale pretrained models. Modern systems typically formulate speech recognition as a sequence prediction problem over **orthographic units**, such as characters, subwords, or words. This formulation has proven effective across many languages and datasets, especially when combined with large pretrained acoustic models and external language models. However, such representations primarily reflect the written form of language rather than the phonetic structure of speech. As a result, ASR systems often rely on large vocabularies to ensure lexical coverage and may struggle to capture pronunciation variability present in natural speech.

This limitation becomes particularly relevant for Vietnamese where syllables serve as the fundamental linguistic units and their internal phonological structure governs pronunciation. In this language, speech is naturally organized into syllables composed of constrained combinations of phonemes. Nevertheless, most current ASR systems ignore this structure and directly predict orthographic tokens, which do not explicitly encode phonological composition. Consequently, orthographic modeling may obscure important phonetic information and introduce unnecessary complexity in the output space.

An alternative perspective is to model speech recognition at the **phoneme level**, which more closely reflects the acoustic realization of speech. Phoneme-based representations provide a compact inventory of basic sound units and naturally capture pronunciation variation. Despite these advantages, phoneme-level decoding is often avoided in end-to-end ASR due to the additional complexity required to map phoneme sequences back to written text, typically through pronunciation lexicons or external language models. This issue, however, is significantly mitigated in languages with phonemic orthography such as Vietnamese, where the correspondence between phonemes and written forms is relatively transparent.

In this work, we propose a **Syllabic-Structure Decoder** designed for ASR in Vietnamese. Instead of predicting orthographic tokens, the proposed decoder generates structured phoneme sequences that explicitly represent the internal composition

of syllables. This design aligns more closely with the phonological structure of speech while maintaining a compact output vocabulary. By modeling syllables through their phonemic components, the system can represent a large lexical space using a small set of phonemic units while preserving full vocabulary coverage.

We evaluate the proposed method on two large-scale Vietnamese ASR datasets: **LSVSC** (Tran et al., 2024), which represents standard speech, and **UIT-ViMD** (Dinh et al., 2024), a corpus containing diverse dialectal variations. Moreover, Vietnamese provides a monosyllabic structure and largely phonemic orthography, which allows phoneme sequences to be converted to written text without requiring an additional language model. Experimental results demonstrate that the proposed approach consistently outperforms strong pretrained baselines, including PhoWhisper (Le et al., 2024) and Wav2Vec2 (Baevski et al., 2020), while using a substantially smaller output vocabulary and requiring no additional training resources.

The main contributions of this work can be summarized as follows:

1. **A Phonemic Tokenization Algorithm** for encoding Vietnamese texts as a sequence of phonemes rather than words or subword units with linear time complexity and comprehensive linguistic coverage.

2. **The Syllabic-Structure Decoder** for end-to-end ASR that explicitly models speech at the phonemic level.

These findings suggest that incorporating phonological structure into ASR decoding offers a promising direction for improving recognition accuracy and efficiency in Vietnamese and other similar languages where syllables constitute the core linguistic units.

2 Related Works

Recent advances in Automatic Speech Recognition (ASR) have been driven by end-to-end neural architectures that directly map speech signals to text. Early approaches commonly relied on the Connectionist Temporal Classification (CTC) (Graves et al., 2006) objective or attention-based encoder–decoder models (Radfar et al., 2022; Yao et al., 2024; Gulati et al., 2020; Le et al., 2025). More recently, large-scale pretrained speech models have achieved strong performance across many languages. Notable examples include Whisper (Le et al., 2024) and Wav2Vec (Baevski et al., 2020). These systems typically predict orthographic tokens, such as characters or subwords.

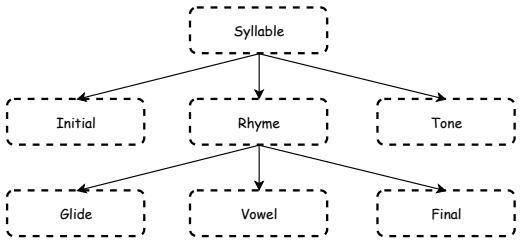

Figure 1: Syllabic Structure in Vietnamese

Phoneme-based representations have long been used in traditional ASR systems (Rabiner, 1989; Jelinek, 1997; Hinton et al., 2012) because they better reflect the acoustic realization of speech and require a compact inventory of units. However, phoneme-level decoding is less common in end-to-end ASR due to the additional complexity of converting phoneme sequences into written text, which often requires pronunciation lexicons or external language models.

3 Methodology

3.1 Syllabic Structure of Vietnamese

Every syllable in Vietnamese share the same structure that includes three main components: initial, rhyme, and tone (Figure 1). Rhyme can be further decomposed into three smaller components: medial, vowel, and tone (Châu, 2002; Xuan, 1985; Đoàn Thiện Thuật, 2016). While medial and tone are optional, the vowel is compulsory for constructing all syllable in Vietnamese. In every syllable, rhyme is the mandatory component while initial and tone can be absent. As a monosyllabic language (Hieu Nguyen et al., 2025; Giáp, 2008, 2011), every Vietnamese word corresponds to at most one syllabic. Space is used to separate any two consecutive words in writing.

Moreover, Vietnamese has a phonemic orthography. This means this language exhibits a high correspondence between grapheme and phoneme. Given the orthographic form of a Vietnamese word, native speakers can easily know how it is pronounced and vice versa. This distinguished feature allows us to develop an algorithm that performs the grapheme-phoneme conversion without any linguistic ambiguity but comprehensive linguistic coverage. Detailed analysis and discussion

Algorithm 1: The algorithm for converting text to phonemes.

Data: Transcript of the audio $w = (w_1, w_2, ..., w_n)$ and the dictionary of IPA (D).

Result: A sequence of syllables $p = (p_1, p_2, ..., p_n)$ of the given input transcript $w = (w_1, w_2, ..., w_n)$. Each phoneme $p_i = (p_i^{init}, p_i^{rhyme}, p_i^{tone})$ is a triplet of IPA for the initial, rhyme, and tone.

```
phonemes ← an empty list [];
for W in w do
    S ← D[W];
    p_tone, S ← get_tone(S);
    p_initial, S ← get_initial(S);
    p_glide, S ← get_glide(S);
    p_vowel, S ← get_vowel(S);
    p_final ← get_final(S);
    p_rhyme ← merge(p_glide, p_vowel, p_tone);
    phonemes ← Append
      (p_i^init, p_i^rhyme, p_i^tone);
end
return phonemes;
```

on the phonetics and orthography of Vietnamese are given in the Appendix B, which establish the fundamental theory of our Phonemic Tokenizer.

3.2 Phonemic Tokenizer

In order to represents text at the phonemic level, we design the **Phonemic Tokenizer** that (1) retrieve the phonemic representation of the monosyllabic word; and (2) decompose the syllable into phonemes correspondence to the syllabic structure of the respective language. The phonemic representation retrieval of words can be done using the dictionary.

Retrieved syllables can be analyzed into a structure of phonemes, which can be written using the IPA symbols, by an linear time-complexity algorithm for each syllable as described in Algorithm 1.

Moreover, each of the algorithms *get_tone*, *get_initial*, *get_glide*, *get_vowel*, *get_final*, and *get_tone* iteratively over every phonemes in the given syllable to determine the syllabic components (Appendix C. Therefore all of them have the time complexity of $\mathcal{O}(n)$ that. Accordingly, the time complexity of the Algorithm 1 is linear $\mathcal{O}(n)$ while using a fix sized of vocabulary of phonemes but tokenizing all Vietnamese words.

Compared to other text tokenization approach such as word or subword, this phonemic approach represents two main advantages:

- **Independent from large-corpora training**: As the syllabic structures of Vietnamese are consistent, we can decompose them consistently into sequence of phonemes using rules from previous linguistic studies (Xuan, 1985; Đoàn Thiện Thuật, 2016).
- **Comprehensive Vocabulary Coverage**: There are stable set of phonemes for every syllabic component, which helps the process of determining the phonemes straightforward. Moreover, all syllable in Vietnamese share the same syllabic structure, hence the linguistic coverage is ensure without any data-driven training.
- **Linear Time Complexity**: As the syllabic structure of natural languages is stable, the process of determining the syllabic components is $\mathcal{O}(n)$ (Alg. 1).

The conversion from syllables to words is straightforward for Vietnamese (Appendix B).

3.3 Syllabic-Structure Decoder

To leverage the internal structure of syllables, we introduce a *multi-token autoregressive decoder* that predicts the syllabic components (comp) in parallel at each timestep, while maintaining autoregression across syllables (Figure 2).

3.3.1 Component-Specific Embeddings

At timestep t, the decoder receives 3 syllabic components $(\mathrm{i}_k, \mathrm{r}_k, ..., \mathrm{t}_k)$ of previous syllables k $(1 \leq k \leq t-1)$. Each component is embedded in its own learned subspace, preserving the distinct phonological roles of each one. In particular, the standard embedding layer retrieves the embedding vector of every syllabic component as:

$$emb_k^{comp} = EmbeddingLayer(comp_k) \in \mathbb{R}^{dim} \quad (1)$$

Accordingly, each syllable is represented as the concatenated vector of three components $emb_k = [emb_k^i; emb_k^r; emb_k^t]^T \in \mathbb{R}^{3 \times dim}$. This embedding

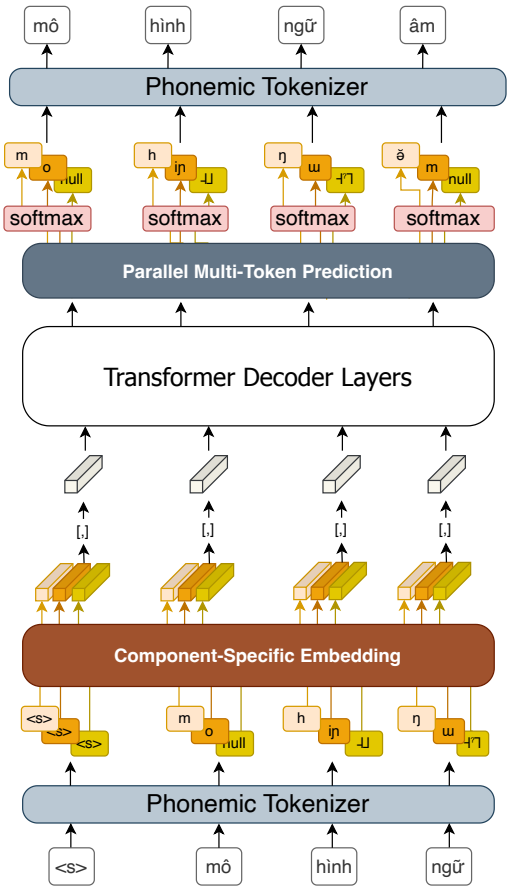

Figure 2: The Syllabic-Structure Decoder.

vector is then projected back to the latent space of the model by a linear function:

$$emb_k = W_e emb_k + b \in \mathbb{R}^{dim} \quad (2)$$

where $W_e \in \mathbb{R}^{dim \times 3 \cdot dim}$ represents the weights of the linear layer and $b \in \mathbb{R}^{dim}$ is the bias vector.

3.3.2 Attention over Acoustic Context

This step is similar to the standard decoding mechanism (Vaswani et al., 2017a), where the unified representation attends to encoder outputs via multi-head attention, thereby integrating acoustic context with phonemic structure and enabling the decoder to capture dialect-sensitive cues and long-range temporal dependencies.

3.3.3 Parallel Multi-Token Prediction

Given the feature vector $f_k \in \mathbb{R}^{dim}$ representing the syllable at timestep k, we aim to extract three respective syllabic components. To this end, we define functions to learn the following probabilistic density

$$P(i^k, r^k, t^k | f_k) = P(i^k | f_k) P(r^k | f_k, i^k) P(t^k | f_k, i^k, r^k)$$

However, as we described in Section 3.1, vowel is the mandatory component of the rhyme, while initial and tone can be absent. Accordingly, we design the method that first learns which is the correct rhyme of the syllable, then given the rhyme, which initial and tone should be given. The above probabilistic density is rewritten as follows:

$$P(i^k, r^k, t^k | f_k) = P(r^k | f_k) P(i^k | f_k, r^k) P(t^k | f_k, r^k)$$

The conditional probabilistic density $P(r^k | f_k)$ is described as the following function:

$$P(r^k | f_k) = softmax(f_k^r) \quad (3)$$

with $f_k^r = \sigma(W_r f_k + b_i) \in \mathbb{R}^{dim}$ where $W_i \in \mathbb{R}^{V_i \times dim}$ V_i is the collection of all initials and $b_i \in \mathbb{R}^{dim}$ is the bias vector.

Using the feature vector for the rhyme component f_k^r, we defined the rhyme-driven syllabic representation f_k^{it} which is the fine-grained version of f_k after obtaining the rhyme feature vector:

$$f_k^{it} = \sigma(W_{it} f_f + b_{it}) \quad (4)$$

where $f_f = \eta(W_k f_k + b_k) \odot tanh(W_k^r f_k^r + b_k^r)$. This feature vector is then used to obtain the corresponding initial and tone of the syllable at step k:

$$P(i^k | f_k^{it}) = softmax(f_k^i) \quad (5)$$

where $f_k^i = \sigma(W_i f_k^{it} + b_i) \in \mathbb{R}^{dim}$, and

$$P(t^k | f_k^{it}) = softmax(f_k^t) \quad (6)$$

where $f_k^t = \sigma(W_t f_k^{it} + b_t) \in \mathbb{R}^{dim}$ with $W_i \in \mathbb{R}^{V_i \times dim}$ and $W_t \in \mathbb{R}^{V_t \times dim}$ and $b_i, b_t \in \mathbb{R}^{dim}$ are two bias vectors.

Given three syllabic components (i^k, r^k, t^k), we convert them back to orthographic form using the proposed Phonemic Tokenizer as when we convert text into the sequence of phonemes.

4 Experiments

4.1 Baselines, Datasets, and Metrics

4.1.1 Datasets

All experiments are conducted on two publicly available datasets: Vietnamese Multi-Dialect (UIT-ViMD) corpus for multi-dialectal speech (Dinh et al., 2024), and Large-Scale Vietnamese Speech Corpus (LSVSC) (Tran et al., 2024) for standard Vietnamese speech. We adopt the official train–validation–test splits for the same evaluation scenario with previous methods. Only minimal preprocessing is applied: non-Vietnamese words are treated as noise and filtered out from the transcripts, and audio is kept in its original mono 16kHz waveform to retain natural pronunciation and recording conditions. As a result, LSVSC represents the general environment of ASR task while UIT-ViMD constitutes a challenging and realistic benchmark for evaluating the robustness of ASR systems across Vietnamese dialects.

4.1.2 Baselines

We compare the proposed phonemic approach against both *subword-level* and *word-level* ASR systems. As representatives of subword-level modeling, we consider publicly available pretrained Vietnamese ASR models, including PhoWhisper (Le et al., 2024) and Wav2Vec 2.0 variants (Baevski et al., 2020), which employ subword tokenization learned from large-scale corpora.

For the UIT-ViMD dataset, we additionally conduct fully supervised experiments using standard Transformer (Vaswani et al., 2017a) and Conformer (Dong et al., 2018; Gulati et al., 2020) architectures with conventional word-based tokenization. These models serve as word-level baselines and allow us to isolate the impact of the proposed phonemic representation under the same training conditions.

For the LSVSC dataset, we compare against the same subword-level pretrained models (PhoWhisper and Wav2Vec 2.0) as well as previously reported word-level systems, including LAS (Chan et al., 2016), Transformer (Vaswani et al., 2017b), Zipformer (Yao et al., 2024), and ChunkFormer (Le et al., 2025). Results for LAS and Transformer are taken from (Tran et al., 2024), while those for the pretrained baselines, Zipformer, and ChunkFormer are obtained from the HuggingFace community(Lab, 2026).

4.1.3 Metrics

We evaluate models from complementary lexical and phonetic perspectives. At the orthographic level, Word Error Rate (WER) is reported as the primary metric, together with Character Error Rate (CER). For the phonemic-level methods, we assess phonetic accuracy using the Phone Error Rate (PER) over the predicted phonetic sequences. We also report component-wise error rates for Vietnamese syllable initials, rhymes, and tones to analyze the phonological modeling behavior. Together, these metrics capture both transcription accuracy, linguistic fidelity, and dialectal robustness of the proposed decoding mechanism.

4.2 Configuration

For all supervised experiments, models were trained from scratch on the LSVSC and UIT-ViMD corpora without using any external language models on a single NVIDIA H100 GPU. All experiments were implemented using a unified pipeline based on the SpeechBrain end-to-end speech processing toolkit (Ravanelli et al., 2021). Speech signals were represented by 80-dimensional log-Mel filterbank (Fbank) features, with a window size of 25ms and a step size of 10ms.

For encoder–decoder architectures, Transformer employed 12 encoder and 6 decoder layers. Conformer consisted of 8 encoders and 4 decoders, each with an attention dimension of 256, 4 self-attention heads, and a feed-forward dimension of 2048. All models were optimized using the Adam optimizer (Kingma and Ba, 2015) with the Noam learning-rate schedule. Warm-up steps (Gotmare et al., 2019) were set to 40k for the Transformer and 20k for the Conformer, with initial learning rates of 10^{-3} and 4×10^{-4}, respectively. Label smoothing (Szegedy et al., 2016) and dropout were both set to 0.1 for regularization. Training used a joint CTC–Attention objective (Kim et al., 2017), with the CTC loss weight set to 0.30 (Transformer) and 0.15 (Conformer). This joint objective promotes monotonic alignment through CTC while preserving the modeling flexibility of attention-based decoding, resulting in more stable training.

4.3 Results

4.3.1 Results on the LSVSC dataset

Table 1 presents the performance comparison on the LSVSC test set. Both of our proposed models achieve the best recognition accuracy

Model	Setup	#Params	CER (%)	WER (%)	Time(s)↓
HuggingFace Published Baslines✳					
PhoWhisper-base	FT	244M	–	11.23	–
PhoWhisper-medium	FT	769M	–	10.25	–
PhoWhisper-large	FT	1.5B	–	10.08	–
Wav2Vec2-base-vi	FT	95M	–	9.89	–
Zipformer	S	65M	–	10.23	–
Paper Published Baslines†					
LAS	S	–	5.79	9.73	–
LAS + fixedSA	S	–	5.05	8.53	–
Transformer	S	26.3M	4.68	8.01	0.16
Transformer + fixedSA	S	26.3M	4.17	7.24	0.16
Transformer + adaptSA	S	26.3M	3.90	6.85	0.16
ChunkFormer	S	110M	–	8.85	–
Our Implementations					
Our-Transformer	S	25.9M	**3.58**	**5.83**	0.15
Our-Conformer	S	28.0M	3.62	5.84	0.14

Table 1: CER and WER (Test) comparison on the LSVSC dataset. Lower is better. Setup: Fine-Tuning (FT), and Fully Suppervise (S). Time(ms): inference time in seconds on the test set. †Results reported from (Tran et al., 2024). ✳Results reported from (Lab, 2026).

Model	Setup	Params	WER↓	WER_N	WER_C	WER_S	Time(s)↓
Published Baselines†							
PhoWhisper-small	–	74M	21.78	18.17	27.18	23.80	0.73
PhoWhisper-base	–	244M	17.11	15.25	18.50	18.38	1.21
wav2vec2-base-vi	–	95M	22.48	19.87	27.15	23.82	0.03
wav2vec2-base-vi-160h	–	95M	31.12	27.09	40.49	32.83	0.03
wav2vec2-base-vi-250h	–	95M	16.83	14.51	19.73	18.23	0.07
PhoWhisper-base	FT	74M	16.30	–	–	–	–
wav2vec2-base-vi	FT	95M	15.80	–	–	–	–
wav2vec2-base-vi-160h	FT	95M	17.49	–	–	–	–
wav2vec2-base-vi-250h	FT	95M	13.56	–	–	–	–
Our Implementations							
Transformer	S	29M	16.66	14.68	18.64	17.95	0.37
Conformer	S	31M	17.56	16.05	19.92	18.39	0.29
Our-Transformer	S	26M	12.59	**10.77**	14.80	13.68	0.33
Our-Conformer	S	28M	**12.58**	10.88	**14.69**	**13.58**	**0.28**

Table 2: Performance comparison on Vietnamese multi-dialect ASR. Metrics include word error rate (WER, %) and phone error rate (PER, %). WER_N, WER_C, and WER_S denote WER of Northern, Central, and Southern speech. Setup: Zero-shot (-), Fine-Tuning (FT), and Fully Suppervise (S). † Results reported from (Dinh et al., 2024).

among all compared systems. Specifically, **Our-Transformer** obtains a CER of 3.58% and a WER of 5.83%, while **Our-Conformer** achieves a CER of 3.62% and a WER of 5.84%. These results outperform all previously reported baselines, including the strongest fully supervised model, Transformer+adaptSA, which reports a CER of 3.90% and a WER of 6.85%.

The improvements are even larger when compared to publicly available pretrained models such as Wav2Vec2-base-vi (9.89% WER) and PhoWhisper-large (10.08% WER), despite using substantially fewer parameters. These results demonstrate the effectiveness of the phonemic approach for Vietnamese ASR.

Furthermore, the proposed models maintain competitive inference speed, requiring only 0.15 s and 0.14 s per utterance for the Transformer and Conformer variants, respectively. We attribute these gains to the proposed phonemic formulation, which decomposes each Vietnamese syllable into four components and predicts them in parallel. By explicitly modeling the internal structure of Vietnamese syllables, the model learns more discriminative linguistic representations, leading to consistent improvements across different encoder architectures.

Model	Overall (%)	Initial (%)	Rhyme (%)	Tone (%)
UIT-ViMD dataset				
Transformer	**8.21**	8.32	9.53	6.76
Conformer	8.27	8.29	9.59	6.89
LSVSC dataset				
Transformer	**3.57**	3.53	4.13	3.04
Conformer	3.59	3.58	4.14	3.02

Table 3: Phoneme Error Rate (PER) on UIT-ViMD and LSVSC. PER is broken down by Vietnamese syllable component: *Initial* (initial consonant), *Rhyme*, and *Tone*. Lower is better.

Dataset	Modeling Unit	Architecture	Unique Correct Words↑	Pearson r	Spearman ρ
UIT-ViMD	Word-level	Transformer	1,696	0.79	0.76
		Conformer	1,718	0.77	0.74
	Phonemic-level	Transformer	**1,983**	**0.63**	**0.54**
		Conformer	**1,965**	0.64	0.54
LSVSC	Word-level	Transformer	2,211	0.78	0.66
	Phonemic-level	Transformer	2,766	0.47	0.12
		Conformer	2,748	0.48	0.14

Table 4: Lexical diversity and frequency bias analysis for word-level and phonemic-level ASR models on the UIT-ViMD and LSV datasets.

4.3.2 Results on the UIT-ViMD dataset

On the multi-dialect domain, our proposed phonemic models still achieve significant results. According to Table 2, our proposed phonemic models achieve the best overall performance, with **Our-Conformer** obtaining the lowest WER of 12.58% and **Our-Transformer** achieving a comparable WER of 12.59%. Both models substantially outperform their corresponding character-based baselines, reducing WER by more than 4 absolute percentage points.

The improvements are consistent across all dialect regions. Compared with the baseline Transformer, Our-Transformer reduces WER from 14.68% to 10.77% on Northern speech, from 18.64% to 14.80% on Central speech, and from 17.95% to 13.68% on Southern speech. Similar gains are observed for the Conformer backbone, indicating that the proposed representation generalizes well across dialectal variations.

Similar to performance on the LSVSC dataset, the two proposed models remain computationally efficient, requiring only 0.33s and 0.28s per utterance for Our-Transformer and Our-Conformer, respectively. These results suggest that representing Vietnamese speech at phonemic level provides a more robust linguistic representation, enabling better handling of pronunciation differences across dialects while maintaining efficient inference.

4.4 Results Analysis

4.4.1 Analysis on the Syllabic Components

For deeper observation on how Our-Transformer and Our-Conformer obtained at phonemic level, we provided analysis on their PER on initial, rhyme, and tone as in Table 3.

In particular, for both datasets and architectures, the *Tone* component consistently achieves the lowest error rates, with values around 3% on LSVSC and below 7% on UIT-ViMD. This suggests that tonal information can be learned reliably from the acoustic signal and remains relatively robust across different model architectures. In contrast, the *Rhyme* component exhibits the highest error rates, reaching 9.53–9.59% on UIT-ViMD and approximately 4.14% on LSVSC. Rhymes include glide, vowel, and final, hence they involve greater phonetic variability and a larger prediction space, making more challenging to recognize accurately.

Furthermore, all component-level error rates are substantially higher on UIT-ViMD than on LSVSC, reflecting the increased difficulty introduced by dialectal variation. The largest degradation occurs in the rhyme component, suggesting that regional pronunciation differences primarily affect vowel and coda realizations. These findings support the motivation of explicitly modeling syllable structure, as analyzing errors at the component level provides finer-grained insights into the challenges of Viet-

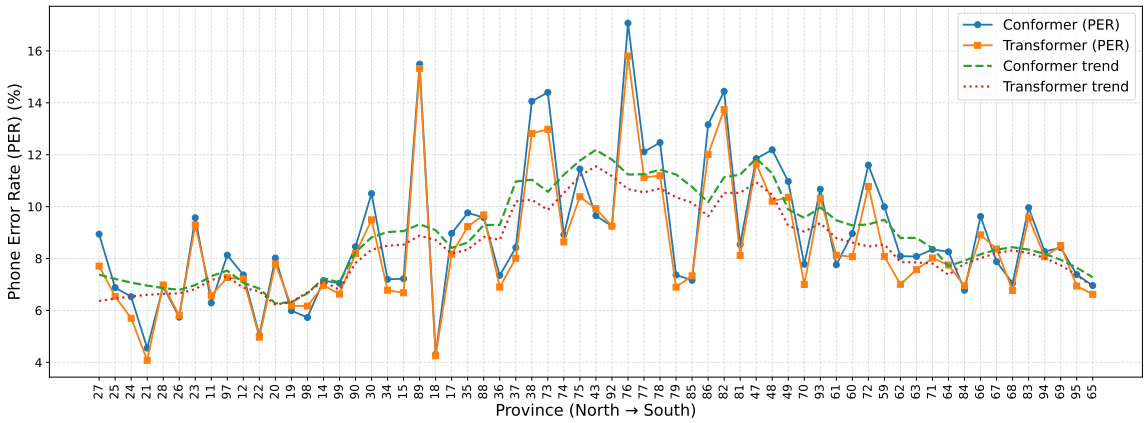

Figure 3: Phone error rate (PER, %) by province for Transformer and Conformer models using the proposed Vietnamese phonetics method.

namese ASR and highlights rhyme modeling as a key direction for future improvement.

4.4.2 Analysis on the Lexical Diversity

As discussed in Section 3.1, Vietnamese has a highly regular relationship between pronunciation and orthography, allowing speakers to infer a word's spelling directly from its pronunciation. We therefore hypothesize that phonemic ASR models rely more on acoustic evidence and less on memorized word-frequency patterns. To examine this hypothesis, Table 4 reports lexical diversity and frequency-bias analyses for word-level and phonemic-level models. Across both datasets, phonemic models correctly recognize substantially more unique words than word-level models, increasing from 1,696–1,718 to 1,965–1,983 on UIT-ViMD and from 2,211 to 2,748–2,766 on LSVSC. These results suggest improved lexical coverage and better generalization to a wider range of words.

This advantage is further reflected in the reduced correlation between word frequency and recognition accuracy. On UIT-ViMD, Pearson's r decreases from 0.77–0.79 to 0.63–0.64, while Spearman's ρ drops from 0.74–0.76 to 0.54. The effect is even more pronounced on LSVSC, where Pearson's r of Transformer decreases from 0.78 to 0.47 and Spearman's ρ from 0.66 to 0.12. Lower correlation values indicate that recognition performance is less dependent on training word frequency, suggesting that phonemic representations reduce frequency bias and improve recognition of infrequent or unseen words relying on acoustic evidence.

4.4.3 Performance on Dialect Speech Recognition

The most interesting results of the proposed phonemic methods are their performance on the dialectal speech. We examine PER of the Our-Transformer and Our-Conformer across Vietnamese provinces and provide their PER in Figure 3, ordered approximately from North to South. These two models highly similar error patterns, indicating that the proposed phonemic representation behaves consistently across different encoder architectures. The PER remains relatively low in many northern provinces, increases noticeably in the central region, and gradually decreases again toward the southern provinces, suggesting that speech from central dialect regions poses greater challenges for phoneme recognition. At the same time, the elevated PER observed in several central provinces highlights the greater phonetic variability of Central Vietnamese, making it a promising direction for future dialect-aware modeling and adaptation.

5 Conclusion

We introduced a phonemic approach to Vietnamese ASR that represents each syllable through its initial, rhyme, and tone components, together with a Syllabic-Structure Decoder that explicitly models this organization. Results on two benchmarks demonstrate consistent gains over word- and subword-based methods, while analyses show improved lexical coverage, reduced frequency bias, and robustness to dialectal variation. Overall, our findings highlight the benefits of incorporating phonological structure into neural ASR models.

Limitations

The proposed approach is motivated by the phonological structure of Vietnamese, where syllables can be systematically decomposed into an initial, a rhyme, and a tone. Consequently, the representation assumes that phonological information is primarily organized at the syllable level. While this assumption is well suited to Vietnamese, its applicability to languages with more complex syllable structures, productive morphology, or extensive phonological processes spanning multiple syllables remains to be investigated.

In addition, the proposed Syllabic-Structure Decoder explicitly models the internal structure of individual syllables but does not explicitly encode phonological relationships across syllable boundaries. Such relationships are instead learned implicitly through the Transformer attention mechanism. As a result, the current framework does not provide a direct account of cross-syllable phenomena such as sandhi, coarticulation, prosodic interactions, or morphophonological alternations. Extending phonemic representations beyond the syllable level to capture these phenomena constitutes an important direction for future work.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

Hoàng Thị Châu. 2002. *Phương ngữ học tiếng Việt*. Nhà xuất bản Đại học Quốc gia Hà Nội.

Alexandre de Rhodes. 1651. *Dictionarium Annamiticum Lusitanum et Latinum*.

Nguyen Dinh, Thanh Dang, Luan Thanh Nguyen, and Kiet Van Nguyen. 2024. Multi-dialect vietnamese: Task, dataset, baseline models and challenges. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7476–7498. Association for Computational Linguistics.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5884–5888. IEEE.

Nguyễn Thiện Giáp. 2008. *Từ vựng học tiếng Việt*. Nhà xuất bản Giáo dục Việt Nam.

Nguyễn Thiện Giáp. 2011. *Vấn đề "từ" trong tiếng Việt*. Nhà xuất bản Giáo dục Việt Nam.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *21st Annual Conference of the International Speech Communication Association, Interspeech 2020, Virtual Event, Shanghai, China, October 25-29, 2020*, pages 5036–5040. ISCA.

Nghia Hieu Nguyen, Dat Tien Nguyen, and Ngan Luu-Thuy Nguyen. 2025. Vietnamese words are not constructed from syllables: Rethinking the role of word segmentation in natural language processing for vietnamese texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):24069–24077.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Frederick Jelinek. 1997. Statistical methods for speech recognition.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 4835–4839. IEEE.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

G-Group AI Lab. 2026. Gipformer - efficient vietnamese speech recognition.

Khanh Le, Tuan Vu Ho, Dung Tran, and Duc Thanh Chau. 2025. Chunkformer: Masked chunking conformer for long-form speech transcription. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Thanh-Thien Le, Linh The Nguyen, and Dat Quoc Nguyen. 2024. Phowhisper: Automatic speech recognition for vietnamese. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 11, 2024*. OpenReview.net.

L.R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Martin Radfar, Rohit Barnwal, Rupak Vignesh Swaminathan, Feng-Ju (Claire) Chang, Grant Strimel, Nathan Susanj, and Thanasis Mouchtaris. 2022. Convrnn-t: Convolutional augmented recurrent neural network transducers for streaming speech recognition.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, and 2 others. 2021. Speechbrain: A general-purpose speech toolkit. abs/2106.04624.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Linh Thi Thuc Tran, Han-Gyu Kim, Hoang Minh La, and Su Van Pham. 2024. Automatic speech recognition of vietnamese for a new large-scale corpus. *Electronics*, 13(5).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Neural Information Processing Systems*.

Hao Cao Xuan. 1985. *Phonologie Et Linearite. Reflexions Critiques Sur Les Postulats De La Phonologie Contemporaine*. Societe d'Etudes Linguistiques et Anthropologiques de France (SELAF).

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *International Conference on Learning Representations*, volume 2024, pages 44440–44455.

Đoàn Thiện Thuật. 2016. *Ngữ âm tiếng Việt*. Nhà xuất bản Đại học Quốc gia Hà Nội.

A Appendix: Experimental Details

A.1 Metrics

We evaluate model performance using complementary metrics that capture orthographic transcription accuracy, phonetic fidelity, dialectal robustness, and lexical generalization behavior.

Word Error Rate (WER). End-to-end transcription quality is primarily measured using *Word Error Rate* (WER), defined as

$$\mathrm{WER} = \frac{S + D + I}{N}, \tag{7}$$

where S, D, and I denote the numbers of word-level substitutions, deletions, and insertions obtained via minimum-edit-distance alignment, and N is the number of words in the reference transcription. In addition to overall WER, we report dialect-wise WER for Northern, Central, and Southern Vietnamese, following the regional grouping defined in the UIT-ViMD benchmark.

Phone Error Rate (PER). For phonetic-level systems, we report *Phone Error Rate* (PER), computed analogously at the phone level:

$$\mathrm{PER} = \frac{S_p + D_p + I_p}{N_p}, \tag{8}$$

where S_p, D_p, and I_p are phone-level substitutions, deletions, and insertions, and N_p is the number of reference phones. PER provides a linguistically grounded assessment of pronunciation modeling, which is particularly relevant for Vietnamese due to systematic dialectal variation at the phone level.

Component-wise Phonetic Error Rates. To analyze phonological modeling behavior with respect to Vietnamese syllable structure, we further compute component-wise error rates for syllable *initials*, *rhymes*, and *tones*. For a component

$c \in \{\text{initial}, \text{rhyme}, \text{tone}\}$, the error rate is defined as

$$\mathrm{ER}_c = \frac{S_c + D_c + I_c}{N_c}, \tag{9}$$

where S_c, D_c, I_c, and N_c are computed over the aligned component sequence.

Dialect Classification Metrics. For multi-task models that jointly perform ASR and dialect identification, we report *classification accuracy* and *macro-averaged F1-score*. Given a set of dialect classes C, macro-F1 is computed as

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \mathrm{F1}_c, \tag{10}$$

ensuring equal weighting across dialects regardless of class imbalance.

Lexical Diversity and Frequency Bias. To assess lexical generalization beyond aggregate error rates, we report the number of *unique word types* correctly recognized at least once in the test set. We further analyze frequency bias by measuring the relationship between training-set word frequency and per-word recall.

For a word type w, its training-set frequency is defined as

$$f_{\text{train}}(w) = \sum_{u \in \mathcal{D}_{\text{train}}} \sum_{i=1}^{|u|} \mathbf{1}\left(u_i = w\right), \tag{11}$$

where $\mathcal{D}_{\text{train}}$ denotes the training corpus and $\mathbf{1}(\cdot)$ is the indicator function. To mitigate the heavy-tailed frequency distribution, we apply a logarithmic transformation:

$$\tilde{f}_{\text{train}}(w) = \log\left(1 + f_{\text{train}}(w)\right). \tag{12}$$

Per-word recall is defined as

$$\mathrm{recall}(w) = \frac{\sum_{(x,y) \in \mathcal{D}_{\text{test}}} \sum_{i=1}^{|\hat{y}|} \mathbf{1}\left(\hat{y}_i = w\right)}{\sum_{(x,y) \in \mathcal{D}_{\text{test}}} \sum_{i=1}^{|y|} \mathbf{1}\left(y_i = w\right)}, \tag{13}$$

where (x, y) is a test utterance–reference pair and $\hat{y}$ is the predicted word sequence after optimal alignment.

The relationship between training frequency and recognition performance is quantified using Pearson and Spearman correlation coefficients:

$$r = \mathrm{Pearson}\left(\tilde{f}_{\text{train}}(w), \mathrm{recall}(w)\right), \tag{14}$$

$$\rho = \mathrm{Spearman}\left(\tilde{f}_{\text{train}}(w), \mathrm{recall}(w)\right), \tag{15}$$

computed over all word types appearing at least once in the test set. Lower correlation values indicate reduced dependence on frequency-driven memorization and stronger structural generalization.

B Appendix: Phonetics and Orthography in Vietnamese

Traditionally, the Vietnamese had the Nom alphabet as the main orthography system. However, the Nom alphabet was developed based on the ancient Han alphabet. This means we have to be fluent in the ancient Han to have the ability to use the Nom alphabet for reading and writing. Later on, (de Rhodes, 1651) used the Latin alphabet to describe the speech sound of this language. This orthography system is simple and effective for describing almost all phonetic phenomena in Vietnamese. With its advantages that counterbalance its disadvantages, this Latin alphabet is gradually improved over time and finally becomes the national alphabet system of modern Vietnamese.

Although having the Nom alphabet or the Latin alphabet, these orthography systems all reflect two consistent characteristics of Vietnamese:

1. Vietnamese is a monosyllabic language.
2. The correspondence between graphemes and phonemes in Vietnamese is consistent.

That is, in this language, we do not have the linking pronunciation as in English, and every phoneme has persistent writing forms. In Vietnamese, each syllable has three components: initials, rhymes, and tones. Rhyme has smaller components, which are glide, vowel, and final. We provide the list of all phonemes according to the syllabic structure of Vietnamese:

- 22 phonemes as the initials:
  - Plosive consonants: /b, t, tʰ, k/.
  - Fricative consonants: /f, d, ɣ, z, j, s, ʂ, c͡ɕ, t͡ʂ, ŋ, x, v/.
  - Nasal consonants: /n, m, ŋ, ɲ/.
  - Vibrant consonants: /r, l/.
- 01 phonemes as the glide: /u̯/.
- 15 phonemes as the vowels:
  - Diphthongs: /ie, uo, ɯə/.

- Monophthongs: /a, ă, ɤ̆, i, ɛ, e, u, ɯ, o, ɔ, ɔː, ə/.

- 10 phonemes as the finals:
  - Nasal consonants: /n, t/.
  - Labial consonants: /m, p/.
  - Velar consonants: /ŋ, k/.
  - Palatal consonants: /ɲ, c/.
  - Semivowels: /u̯, i̯/.

- 6 phonemes denote tones:
  - Flat tone is denoted by nothing.
  - Low falling tone: /˧˩/.
  - Mid raising tone: /˧˥/.
  - Mid falling tone: /˧˩/.
  - Mid glottalized-falling tone: /˧ʔ˩/.
  - Mid glottalized-raising tone: /˧ʔ˥/.

The writing form of these phonemes is consistent regardless of grammar. In particular:

- 6 tones are denoted by a mark above or below the graphemes of vowels:
  - Flat tone is denoted by nothing (a).
  - Low falling tone /˧˩/ is denoted by a grave accent (à).
  - Mid raising tone /˧˥/ is denoted by an acute accent (á).
  - Mid falling tone /˧˩/ is denoted by a hook above (ả).
  - Mid glottalized-falling tone /˧ʔ˥/ is denoted by a tilde above (ã).
  - Mid glottalized-raising tone /˧ʔ˩/ is denoted by a dot below (ạ).

  In the following texts, we provide writing forms of vowels regarding the mentioned phonemes. These examples might include the tone mark above or below the graphemes. Readers can discard these marks to see the true writing form of the vowels in Vietnamese. For instance, the dot below **iê** /iə/ in word **kiệm** /kiem˧ʔ˥/ denotes the mid glottalized-raising tone /˧ʔ˥/.

- 22 initials have 26 writing forms:
  - b /b/. Eg: **b**a mẹ, **b**ánh kẹo, **b**uôn **b**án.
  - t /t/. Eg: **t**âm **t**ư, **t**ịnh **t**iến, **t**ính cách.
  - th /tʰ/. Eg: **th**ách **th**ức, **th**ành **th**ạo.
  - k, c, or q /k/. Eg: **c**ách mạng, **q**uan hệ, hiện **k**im.
  - ph /f/. Eg: **ph**ụ huynh, **ph**ong cách, **ph**ân định.
  - đ /d/. Eg: **đ**ưa **đ**ón, **đ**ậm **đ**à.
  - gh or g /ɣ/. Eg: **g**a tàu, **g**ánh hát, **g**anh **gh**ét.
  - gi /z/. Eg: **gi**ếng nước, **gi**ống loài.
  - đ /d/. Eg: **đ**ứng **đ**ắn, **đ**êm tối, **đ**èn **đ**uốc.
  - d /j/. Eg: **d**a **d**ẻ, tiêu **d**ùng, **d**ụng cụ.
  - x /s/. Eg: sản **x**uất, **x**uất **x**ứ, **x**e cộ.
  - s /ʂ/. Eg: xác **s**uất, **s**o **s**ánh, **s**ao chép.
  - ch /c͡ɕ/. Eg: **ch**ứa **ch**an, **ch**e **ch**ở/.
  - tr /t͡ʂ/. Eg: **tr**anh chấp, tiệt **tr**ùng.
  - ng or ngh /ŋ/. Eg: mong **ng**óng, tình **ngh**ĩa.
  - nh /ɲ/. Eg: **nh**à cửa, nỗi **nh**ớ, **nh**ung lụa.
  - l /l/. Eg: **l**ấm **l**em, **l**ung **l**inh, **l**ối về.
  - r /ɾ/. Eg: **r**ậm **r**ạp, **r**ón **r**én, **r**ực **r**ỡ.
  - kh /x/. Eg: **kh**ó **kh**ăn, **kh**ởi sắc, **kh**ấm **kh**á.
  - v /v/. Eg: **v**ui **v**ẻ, **v**ương **v**ấn, **v**ẫy **v**ùng.
  - m /m/. Eg: **m**ong **m**ỏi, **m**ay **m**ắn, **m**ênh **m**ông.
  - n /n/. Eg: đất **n**ước, **n**úi **n**on, **n**ông cạn.

- The glide /u̯/ has two writing forms: u or o. Eg: q**u**ê nhà, h**o**a cỏ, kh**u**yến khích.

- 03 diphthongs have 8 writing forms:
  - iê, yê, ia, or ya /iɛ/. Eg: kh**iế**m thị, **yê**n ắng, ch**ia** sẻ, khu**ya** khoắt.
  - uô or ua /uo/. Eg: kh**uô**n khổ, m**ua** bán.
  - ươ or ưa /ɯə/. Eg: kh**ươ**u giác, dây d**ưa**.

- 12 monophthongs have 13 writing forms:
  - a /a/. Eg: b**a** mẹ, tr**a**nh vẽ, ng**ã** b**a**, m**ải** miết, làng ch**ài**.
  - ă or a /ă/. Eg: ánh **ắ**ng, n**ắ**m tay, n**ă**m tháng, **áy** n**áy**, chạy nhảy.
  - â /ɤ̆/. Eg: n**â**ng niu, **ấ**n định, ng**â**n vang.
  - i or y /i/. Eg: th**i** cử, tr**ĩ**u nặng, b**ĩ**u môi.
  - ê /e/. Eg: k**ế**t quả, thể hi**ệ**n, mân m**ê**.
  - e /ɛ/. Eg: mùa h**è**, x**e** cộ, t**é** ngã.
  - u /u/. Eg: th**u** mua, m**ủ**m mỉm, l**u**ng lay, tr**u**ng thành.
  - ư /ɯ/. Eg: tr**ư**ng cầu, xây d**ự**ng, **ư**ng ý.

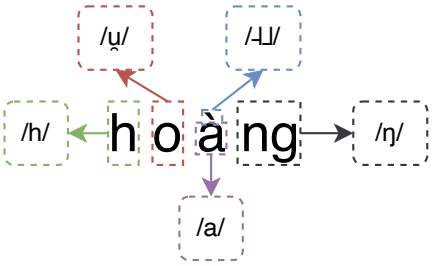

Figure 4: Example for the consistency between graphemes and phonemes in Vietnamese.

  - o /ɔ/. Eg: chăm s**ó**c, m**o**ng ng**ó**ng, tr**o**ng veo.
  - oo /ɔː/. Eg: x**oo**ng chảo.
  - ô /o/. Eg: tr**ô**ng đồng, **ô**ng hút, c**ô** nhân.
  - ơ /ə/. Eg: m**ơ** mộng, c**ơ** nh**ỡ**, ch**ơ**i b**ờ**i.

- 10 final consonants have 12 writing forms:
  - i or y /i̯/. Eg: làng chà**i**, mỏ**i** mệt, chạ**y** đua, bay nhả**y**.
  - m /m/. Eg: ê**m** ấ**m**, nhiệ**m** màu, mâ**m** cỗ.
  - n /n/. Eg: na**n** giải, no**n** nớt, tả**n** mạ**n**.
  - ng /ŋ/. Eg: sa**ng** trọ**ng**, trố**ng** trải, su**ng** túc.
  - nh /ɲ/. Eg: nha**nh** nhẹn, bê**nh** vực, bi**nh** quyền.
  - p /p/. Eg: phậ**p** phồng, thấ**p** thỏm, thá**p** tùng, gượng é**p**, ức hiế**p**, tẩm ướ**p**.
  - t /t/. Eg: lấn á**t**, bá**t** đĩa, kế**t** quả, hớ**t** hả.
  - c /**k**/. Eg: cú**c** áo, chự**c** chờ, bố**c** vá**c**.
  - ch /**c**/. Eg: cá**ch** thứ**c**, chí**ch** ngừa.
  - u or o /u̯/. Eg: tra**u** chuốt, chị**u** đựng, xoong chả**o**, xiêu vẹ**o**.

It is important to note that the writing form of phonemes in Vietnamese is consistent in all usage cases, regardless of grammar (tense, aspect, mood). This feature defines Vietnamese as an isolating language whose morphology is an aspect of phonetics rather than grammar, as in inflectional languages. To this end, given a Vietnamese word, native speakers can find no difficulty in pronouncing it. On the other hand, Vietnamese natives can write down any Vietnamese word by listening to its speech sound without knowing how to spell it. For instance, given the word **hoàng**, its phonetic representation is /hu̯aŋ˧˩˧/. We can determine the grapheme of the initial /h/ is **h**, of the glide /u̯/ is **o**, of the vowel /a/ is **a**, of the /ŋ/ is **ng**, and tone is denoted by the grave accent above **à** (Figure 4).

However, although the conversion from grapheme to phonemes is straightforward (that is, many graphemes correspond to unique phonemes), the inversion is not all ways many-to-one mapping. There are some phonemes having one-to-many mapping with graphemes, such as the initial /ɣ/ can be written as **g** or **gh**, or the diphthong /ie/ can be written as **iê, yê, ia** or **ya**. Actually, a particular writing form of such phonemes is determined consistently via the neighbor phonemes. We detail here rules for determining writing forms of phonemes having multiple respective graphemes:

- The initial /k/ is written as:
  - **k** if followed by **i** /i/, **ê** /e/, **e** /ɛ/, **iê** /ie/. Eg: **ki**p thời, **kiể**u cách, **kè**m cặp.
  - **q** if followed by **u** /u̯/ as the glide. Eg: **qu**ê hương, **qu**à cáp.
  - **c** otherwise. Eg: **cứ**ng **cỏ**i, **cở**i mở, hạt **cá**t, rau **củ**.

- The initial /ɣ/ is written as:
  - **gh** if followed by **i** /i/, **ê** /e/, **e** /ɛ/, **iê** /ie/. Eg: bàn **ghế**, **ghi** chép, **ghe** tàu.
  - **g** otherwise. Eg: thanh gươm, gồng gánh, gọi điện.

- The initial /ŋ/ is written as:
  - **ngh** if followed by textbfi /i/, **ê** /e/, **e** /ɛ/, **iê** /ie/. Eg: **nghe** ngóng, **nghiê**m nghị, **nghệ** sĩ.
  - **ng** otherwise. Eg: **ngà**nh nghề, **ngỗ** nghịch, **ngọ**t **ngà**o.

- The diphthong /ie/ is written as:
  - **iê** if the rhyme has a final consonant and no glide. Eg: k**iến** thức, tiết k**iệm**.
  - **yê** if the rhyme has a final consonant and the glide written as **u**. Eg: kh**uyên** bảo, **uyển** ch**uyển**, câu ch**uyện**.
  - **ya** if the rhyme has no final consonant and the glide written as **u**. Eg: đêm kh**uya**.
  - **ia** if the rhyme has no final consonant and no glide. Eg: b**ìa** sách, ch**ia** sẻ.

- The diphthong /uo/ is written as:

- **uô** if the rhyme has a final consonant. Eg: nỗi b**uồn**, m**uối** biển, m**uộn** màn, ch**uồn** ch**uồn**.
- **ua** if the rhyme has no final consonant. Eg: ch**ùa** chiền, nhảy m**úa**.

- The diphthong /ɯə/ is written as:
  - **ươ** if the rhyme has a final consonant. Eg: bia r**ượu**, h**ưởng** thụ, chiêm ng**ưỡng**.
  - **ua** if the rhyme has no final consonant. Eg: ch**ùa** chiền, nhảy m**úa**.

- The monophthong /ă/ is written as:
  - **a** if is is followed by character **y**. Eg: m**áy** bay, c**ay** nồng, t**ay** chân.
  - **ă** otherwise. Eg: b**ắt** tay, b**ằng** lòng, may m**ắn**.

- The monophthong /i/ is written as:
  - **i** if the rhyme has a final consonant. Eg: l**íu** r**ít**.
  - **y** if the rhyme has no consonant. Eg: k**ỷ** luật, l**ý** do.

- The final /i̯/ is written as:
  - **i** if the rhyme has the front vowel /a/, the central vowel /ɯ, ɯə/ or the back vowels /u, o, ɔ, uo/ as the vowel. Eg: m**ải** mê, m**ui** thuyền, h**ỏi** han, m**ồi** chài, g**ửi** gắm, l**ười** biếng, n**uôi** nấng.
  - **y** if the rhyme has /ă, ɤ̆/ as the vowel. Eg: b**ay** lượn, ch**ạy** nh**ảy**, c**ấy** c**ày**.

Following these orthographic rules in Vietnamese, there is no ambiguity in converting phonemes to graphemes and vice versa.

C Appendix: Detailed Implementation of Sub-Procedures in Phonemic Tokenization Algorithm

We provide here the detailed implementation of *get_tone*, *get_initial*, *get_glide*, *get_vowel*, and *get_final* mentioned in Section 3.2, Algorithm 1. According to Algorithm 5, 2, 6, 3, and 4, they defined a stable set of phonemes, then iteratively determine whether the given input S is starting with any phoneme in the defined set. We can easily see that these algorithms have the linear time complexity $\mathcal{O}(n)$. Together with the algorithm 1, the overall time complexity of our tokenizer is $\mathcal{O}(n)$.

Algorithm 2: Algorithm for determining initial of a syllable.

```
Function get_initial(S):
    Input: A syllable S in Vietnamese
    Output: The initial i of S
    initials ← [b, t, tʰ, k, f, d, ɣ, z, j, s, ʂ,
      c͡ɕ, t͡ʂ, ŋ, x, v, n, m, ɲ, ɲ, r, l];
    i ← None;
    for i in initials do
        if S starts with i then
            Remove i from the prefix of S;
            return i, S;
        end
    end
    return None, S
end
```

Algorithm 3: Algorithm for determining vowel of a syllable.

```
Function get_vowel(S):
    Input: A syllable S in Vietnamese
    Output: The vowel v of S
    vowels ← [ie, uo, ɯə, a, ă, ɤ̆, i, ɛ, e, u,
      ɯ, o, ɔ, ɔː, ə, n, t, m, p, ŋ, k, ɲ, c, u̯, i̯];
    for v in vowels do
        if S starts with v then
            Remove v from the prefix of S;
            return v, S;
        end
    end
    return None, S;
end
```

Algorithm 4: Algorithm for determining final of a syllable.

```
Function get_final(S):
    Input: A syllable S in Vietnamese
    Output: The final f of S
    finals ← [n, t, m, p, ŋ, k, ɲ, c, u̯, i̯];
    if f in finals then
        return f;
    end
    return None;
end
```

Algorithm 5: Algorithm for determining tone of a syllable.

```
Function get_tone(S):
    Input: A syllable S in Vietnamese
    Output: The tone t of S
end
tones ← [˧˩, ˧˥, ˧˩, ˧ˀ˩, ˧ˀ˥];
for t in tones do
    if S ends with t then
        Remove t from suffix of S;
        return t, S;
    end
end
return None, S;
```

Algorithm 6: Algorithm for determining glide of a syllable.

```
Function get_glide(S):
    Input: A syllable S in Vietnamese
    Output: The glide g of S
    glides ← [u̯, o̯];
    for g in [u̯, o̯] do
        if S starts with g then
            Remove g from the prefix of S;
            return u̯, S;
        end
    end
    return None, S;
end
```