Unsupervised Speech Recognition at the Syllable Level

Liming Wang[1*], **Kai-Wei Chang**[1], **Kunio Kashino**[2], **David Harwath**[3], **Mark Hasegawa-Johnson**[4], **James R. Glass**[1]
[1]Massachusetts Institute of Technology, [2]NTT, Inc.,
[3]University of Texas at Austin, [4]University of Illinois Urbana-Champaign
limingw@csail.mit.edu

Abstract

Training speech recognizers with unpaired speech and text – known as unsupervised speech recognition (UASR) – is a crucial step toward extending ASR to low-resource languages in the long-tail distribution and enabling multimodal learning from non-parallel data. However, existing approaches based on phones often rely on costly resources such as grapheme-to-phoneme converters (G2Ps) and struggle to generalize to languages with ambiguous phoneme boundaries due to training instability. In this paper, we address both challenges by introducing a syllable-level UASR framework based on masked language modeling, which avoids the need for G2P and the instability of GAN-based methods. Our approach achieves up to a 40% relative reduction in character error rate (CER) on LibriSpeech and generalizes effectively to low-resource languages that have remained particularly difficult for prior methods. Code is publicly available[1].

1 Introduction

Recent advances in self-supervised learning (Baevski et al., 2020; Hsu et al., 2021a; Chen et al., 2022; Chung et al., 2021; Chen et al., 2024; Mohamed et al., 2022) and spoken language modeling (Arora et al., 2025; Chu et al., 2024; Défossez et al., 2024) have made voice assistants increasingly capable. Yet these systems remain far from *language-universal*: most support only a handful of world's languages (Chen et al., 2024; Conneau et al., 2021), largely because large paired speech-text corpora are unavailable for many languages.

A natural step toward language-universal speech technology is to build recognizers from *unpaired* speech and text, i.e., *unsupervised speech recognition* (UASR) (Glass, 2012; Baevski et al., 2021; Liu et al., 2023; Tseng et al., 2024). UASR is important not only for speech recognition itself, but also for downstream applications such as speech synthesis (Ni et al., 2022; Liu et al., 2022), translation (Wang et al., 2023a), and spoken language understanding (Shi et al., 2023). More broadly, it is a core instance of *unpaired multimodal learning* (Artetxe et al., 2018b,a, 2019; Lample et al., 2018a,b; Ma et al., 2019; Hoshen and Wolf, 2018; Goldwasser et al., 2023; Levy et al., 2025), where modalities need to be aligned without parallel data. In the absence of sentence-level alignment, the model must infer higher-level linguistic units–phones, syllables, and words–from raw speech waveforms in conjunction with global text statistics. Thus, UASR provides broader insights into representation learning and multimodal alignment without supervision.

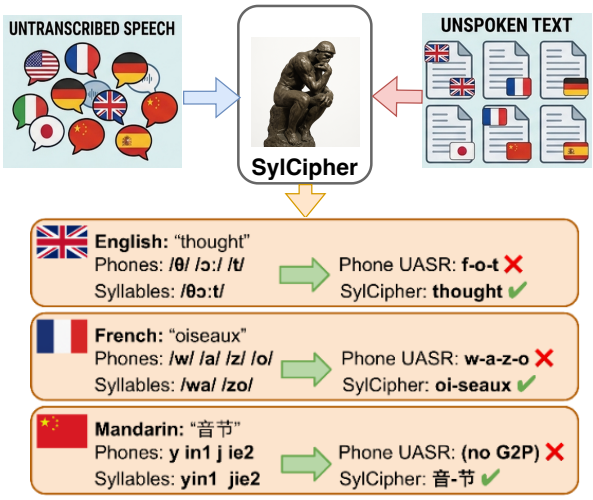
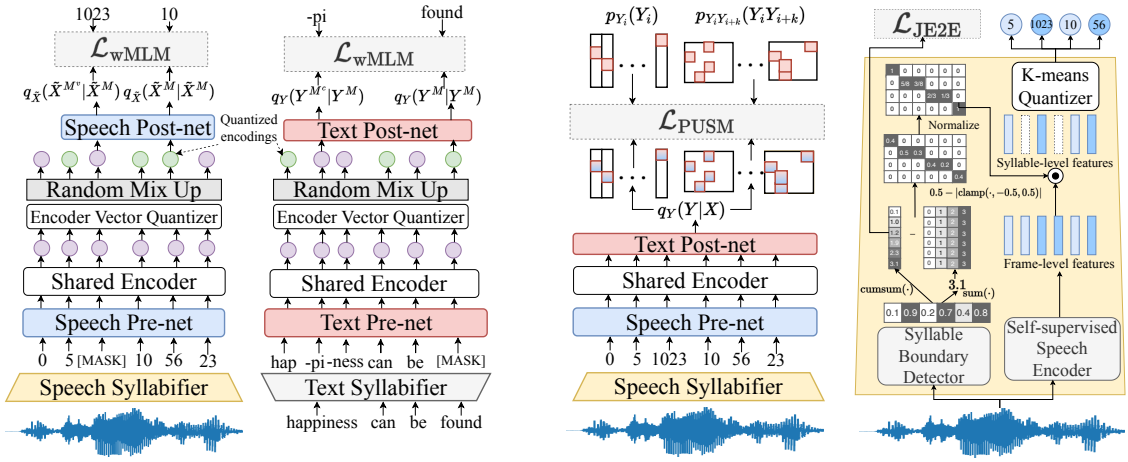


Figure 1: Syllable-level speech-text decipherment.

Most strong UASR systems (Baevski et al., 2021; Liu et al., 2023; Tseng et al., 2024) operate at the *phoneme-level*. This requires converting written symbols (graphemes) into phonemes using a grapheme-to-phoneme convertor (G2P). In many languages, however, G2P is difficult or unavailable

[*]Now at the Chinese University of Hong Kong
[1]https://github.com/cactuswiththoughts/SylCipher

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 2115–2135
October 24-29, 2026 ©2026 Association for Computational Linguistics



(a) MLM-based stages (b) PUSM stage (c) Speech syllabifier

Figure 2: **Overall architecture of SylCipher**. Gray boxes are fixed during training. (a) MLM-based stages: learn a compressed joint semantic space with a shared encoder and random mix-up. (b) PUSM stage: align speech and text spaces by matching lower-order marginals of their distributions.

because writing systems may omit key phonetic information, vary substantially across regions, or fail to represent important sound contrasts (Kandybowicz and Torrence, 2017; Farukh and Vulchanova, 2014; Khan et al., 2024). Even when pronunciation dictionaries exist, building and maintaining them is labor-intensive. Without a G2P, phoneme-based UASR often degrades sharply because speech and raw text become poorly aligned (Liu et al., 2023; Ni et al., 2022). In addition, phone boundaries can be hard to detect in languages with strong coarticulation, such as Mandarin.

A word-level alternative avoids G2P, but introduces a different problem: the vocabulary of rare words is effectively unbounded, making coverage and generalization much harder. Word segmentation also depends on longer-range context, which can destabilize the segmentation mechanisms used in current UASR systems (Wang et al., 2023c).

In this work, we instead study UASR at the *syllable-level*. This choice has three advantages. First, unlike words, the number of syllables is finite, which reduces the long-tail token problem. Second, in many languages, speech and text align more naturally at the syllable level than at the phone or word level. Mandarin is a clear example: written characters often correspond closely to spoken syllables, making syllables a more natural unit than phonemes. Third, recent progress in syllable boundary detection and unit discovery (Baade et al., 2025; Cho et al., 2025) makes it possible to segment speech into syllable-like units without text supervision, often more reliably than unsupervised word segmentation methods while retaining much of the relevant linguistic structure (Peng and Harwath, 2022; Fuchs and Hoshen, 2023).

In this paper, we make the following contributions.

1. We introduce **SylCipher**, to our knowledge the **first syllable-based UASR system**. SylCipher jointly predicts syllable boundaries and discrete speech units from raw speech under a unified self-supervised objective, avoiding adversarial training and improving training stability.

2. We evaluate SylCipher across domains, languages and level of resources. On LibriSpeech, it improves over prior G2P-free UASR systems by up to 40% relative character error rate (CER). On SpokenCOCO, the gains are even larger, demonstrating robustness across domains. On Multilingual LibriSpeech, we build the **first multilingual UASR system** to our knowledge by training SylCipher on multilingual data and outperform prior G2P-free, monolingual approaches by up to 50% relative CER. On Mandarin, it achieves 12.2% phone error rate (PER), outperforming GAN-based UASR methods that fail to even converge. On two low-resource languages, it consistently outperforms phoneme-level approaches.

3. We also apply SylCipher to the **semi-supervised setting** and demonstrate that pseudo-labels from our system can improve semi-supervised ASR performance across

multiple languages with up to 20 hours of paired data.

4. We perform careful ablation and error analysis, examining the effects of token vocabulary size, syllabifier choice, and segmentation mechanisms. In particular, we evaluate **BPE units as a text-derived, noisier approximation** to syllable-like segmental units. These experiments show that the proposed framework can be extended beyond exact syllable inventories, while also highlighting that linguistically or acoustically motivated syllable units provide a stronger inductive bias for UASR.

Paper organizations. Section 2 reviews related work. Section 3 formalizes the syllable-level UASR problem. Section 4 presents SylCipher and its theoretical guarantees. Section 5 reports experiments and ablations. Section 6 concludes with limitations and future work.

2 Related work

Unsupervised speech recognition Early work on UASR assumed the existence of a reliable G2P and formulate the problem as an adversarial game, where a conditional generator predicts a phoneme sequence given a speech waveform, and a discriminator tries to tell real phonemized text apart from the generator's output (Wang et al., 2023b). Within this framework, several works have explored different segmentation mechanisms such as fixed unsupervised phoneme segmentation (Liu et al., 2018), iterative forced alignment (Chen et al., 2019), de-duplication (Baevski et al., 2021) and reinforcement learning (Tseng et al., 2024). (Wang et al., 2023c) proposed an alternative formulation of UASR as an explicit distribution matching problem, by matching the lower-order N-gram distributions of the generated and real texts. Later work further extended this reformulation to use G2P-free tokens such as words (Wang et al., 2023c; Ni et al., 2025), more expressive architecture such as transformers (Ni et al., 2025) and more general text distribution such as masked prediction probabilities (Ni et al., 2025). We adapt this word-level UASR approach to the syllable-level, and supplement it with a simplified training flow, a more stable differentiable boundary detector and additional learning objectives.

Syllable-level self-supervised learning Early work on syllable-level modeling of speech used signal processing techniques (Zhang and Glass, 2009). To discover higher-level structure and more efficient self-supervised learning (SSL) representations from raw speech, (Peng et al., 2023; Cho et al., 2024) proposed to induce syllabic structure from existing SSL models such as HuBERT (Hsu et al., 2021a). To this end, (Peng et al., 2023) probed the self-attention layers of VG-HuBERT (Peng and Harwath, 2022), a visually grounded SSL model to detect syllable-like feature clusters and further refine such clusters using a min-cut algorithm (Shi and Malik, 1997). To sidestep the need for visual data, (Cho et al., 2024) employed a speech-only SSL model trained with utterance-level self-distillation from a HuBERT teacher. Due to the indirect manner of such approaches by which syllabic structures are derived, they are often noisy and unreliable. To cope with this issue, recent works (Baade et al., 2025; Cho et al., 2025) proposed a more direct and targeted approach by performing self-distillation at the syllable-level, which significantly improved syllable boundary detection and unit discovery performance by encouraging sharper contrast between within and between-syllable feature frames.

3 Syllable-level unsupervised speech recognition

In this section, we formulate syllable-level UASR. Let $X = [X_1, \cdots, X_T] \in \mathcal{X}^T$ be a padded sequence of speech feature vectors and $Y = [Y_1, \cdots, Y_L] \in \mathcal{Y}^L$ be a padded sequence of text tokens in the same language. Since tokenized speech is typically longer than its text transcription, we assume $T \geq L$. We further assume X and Y come from two *unpaired* datasets and are therefore statistically independent. Meanwhile, the two modalities are assumed to be *matched*: there exists an ASR function $y^* : \mathcal{X} \mapsto \mathcal{Y}$ such that their distributions p_X and p_Y satisfy

$$p_Y(y) = \int_{x \in \mathcal{X}: y^*(x) = y} p_X(x) \mathrm{d}x, \forall y \in \mathcal{Y}^L. \quad (1)$$

The goal of UASR is to recover y^* given only unpaired X and Y. In the syllable-level setting, the tokens are syllables. The problem is closely related *decipherment*, where one decodes a message written in an unknown script without a lexicon or grammar.

In practice, X is formed from frame-level SSL features (Hsu et al., 2021a), and p_X and p_Y are only

approximately matched because of finite-sample noise and domain mismatch. Moreover, UASR is ill-posed in general. However, (Wang et al., 2023b) show that the mapping y^* in equation 1 becomes identifiable when syllable boundaries are known and the language satisfies mild conditions.

4 SylCipher: Syllable-level UASR via information-constrained masked language modeling

In this section, we present **SylCipher**, a syllable-level framework for UASR. The model first converts continuous speech into syllable-like units, then learns a shared representation space in which speech and text are processed by the same encoder. We first describe the architecture and training objective, then give the theoretical motivation, and finally introduce the practical training modifications.

At a high level, SylCipher has three goals:

- Convert speech into syllable-level units that are easier to align with text;
- Learn a shared latent space for speech and text with a single encoder;
- Make UASR training stable and effective in practice.

4.1 Training: UASR via information compression

As shown in Figure 2, SylCipher is an encoder-only language model with a *shared encoder* for speech and text. Two uni-modal, linear *pre-nets* map the two modalities into a common embedding space: $e_{\tilde{X}} : \mathcal{X}^T \mapsto \mathbb{R}^{L\times d}$ for speech and $e_Y : \mathcal{Y}^L \mapsto \mathbb{R}^{L\times d}$ for text.

Before the speech pre-net, a *speech syllabifier* converts frame-level acoustic features into a syllable-level sequence. It contains (i) a *differentiable soft-pooler* $m : \mathcal{X}^T \mapsto \mathcal{X}^L$ that aggregates frames into syllable segments and (ii) a tokenizer $c : \mathcal{X}^L \mapsto \tilde{\mathcal{X}}^L$ that discretizes the pooled representations. The speech pipeline is therefore

$$e_{\tilde{X}}(\tilde{X}) = e_{\tilde{X}} \circ c \circ m(X), \tag{2}$$

where $\tilde{X} := c \circ m(X) \in \tilde{\mathcal{X}}^L$.

The soft-pooler first employs a trainable boundary detector to predict frame-level boundary probabilities $b(X) \in [0,1]^T$. The boundary detector is pretrained with initial boundaries $[\hat{b}_1(X), \cdots, \hat{b}_T(X)]$ from the unsupervised syllable detector Sylber-(Cho et al., 2025). These frame-level boundary probabilities are converted into a soft pooling mask $a(X)$, which maps frame-level speech features into segment-level representations:

$$a_{it}(X) := \sigma_\epsilon \left(i - \sum_{\tau \le t} b_\tau(X) \right), \tag{3}$$

$$m_i(X) := \sum_{t=1}^{T} \frac{a_{it}(X)}{\sum_{\tau=1}^{T} a_{i\tau}(X)} X_t, \tag{4}$$

where $\sigma_\epsilon(x) := \epsilon - |\mathrm{clamp}(x, -\epsilon, \epsilon)|$. This produces sparse pooling weights and avoids the numerical instability of earlier soft-pooling methods (Bhati et al., 2022; Wang et al., 2024). The segment-level representations are passed to a vector quantizer and speech pre-net.

Both modalities are then processed by the same encoder f:

$$f_{\tilde{X}}(\tilde{X}) := f \circ e_{\tilde{X}}(\tilde{X}), \quad f_Y(Y) := f \circ e_Y(Y). \tag{5}$$

In practice, c is implemented with a differentiable K-means vector quantizer (Ni et al., 2025), and f is a multi-layer transformer (Vaswani et al., 2017).

Distribution matching. Without paired speech-text data, we train the model by matching the unimodal speech and text distributions. Let $g_{\tilde{X}}$ and g_Y be speech and text *post-nets* built on top of the shared encoder, with

$$\begin{aligned} q_Z(Z) &:= g_Z \circ f_Z(Z) = \prod_{i=1}^{L} g_{Z,i}(Z_i | f_Z(Z_{1:(i-1)})) \\ &= \prod_{i=1}^{L} \frac{e^{W^Z_{iZ_i} f_Z(Z_{1:(i-1)})}}{\sum_z e^{W^Z_{iz} f_Z(Z_{1:(i-1)})}}, \end{aligned} \tag{6}$$

where $Z \in \{X, Y\}$, Z_i is the syllable unit at segment i and $g_{Z,i}$ is the linear-softmax post-net prediction head, following JSTTI (Ni et al., 2025). Thus, q_Z defines a sequence-level distribution computed as the product of the softmax probabilities.

Training minimizes *Kullback–Leibler (KL) divergence* to the two unimodal distributions:

$$\min_{q_{\tilde{X}}, q_Y} D_{\mathrm{KL}}(p_{\tilde{X}} || q_{\tilde{X}}) + D_{\mathrm{KL}}(p_Y || q_Y), \tag{7}$$

while **restricting the capacity** of the shared representation to prevent speech and text from drifting into disjoint regions of the latent space. We

prove that Eq. 7 is able to match the true and generated text distributions under regularity conditions (Wang et al., 2023b), and defer the formal objective, assumptions, and theoretical guarantee to Appendix A and C.

Practical training. In practice, SylCipher uses three training components. First, it applies weighted masked language modeling (wMLM) on both speech and text to learn the shared encoder and post-nets to approximate the KL terms in Eq. 7. Given a mask distribution p_M, we minimize the weighted loss:

$$\begin{aligned}&\mathcal{L}_{\text{wMLM}}(\theta_X, \theta_Y) := \\ &\quad - \mathbb{E}_{Z\sim\frac{1}{2}p_{\tilde{X}}+\frac{1}{2}p_Y,\, M\sim p_M}\left[\ln q_Z(Z^M|Z^{M^c})\right] \\ &\quad - \lambda\mathbb{E}_{Z\sim\frac{1}{2}p_{\tilde{X}}+\frac{1}{2}p_Y,\, M\sim p_M}[\ln q_Z(Z^M|Z^M)],\end{aligned}$$

where Z^M is masked portion of the embedding, Z^{M^c} is the unmasked portion of the embedding, θ_X and θ_Y are speech-related and text-related parameters, and λ balances the function of the decoder for mask prediction and reconstruction. Note that for $\lambda = 0$, Eq. 8 approximates Eq. 7 since

$$\begin{aligned}&D_{\text{KL}}(p_Z||q_Z) \propto -\mathbb{E}_{p_Z} \ln q_Z(Z) \\ &\propto -\mathbb{E}_{p_Z, p_M} \ln q_Z(Z^M|Z^{M^c}) = \mathcal{L}_{\text{wMLM}},\end{aligned}$$

where p_Z and q_Z denote the true and estimated unimodal distributions respectively.

Second, since the computations Eq. 3 is differentiable, the boundary detector is updated jointly with the UASR objectives, allowing joint end-to-end (JE2E) training to refine the boundaries using text-distribution matching signals. To discourage both over- and under-segmentation, a soft syllable-count constraint.

$$\mathcal{L}_{\text{JE2E}}(\theta_X) := \mathbb{E}_X \left| \sum_{t=1}^{T} b_t(X) - \hat{b}_t(X) \right|.$$

Third, once MLM saturates, it adds positional unigram and skipgram matching (PUSM) (Wang et al., 2024; Ni et al., 2025) for explicit distribution matching.

The overall training objective is

$$\mathcal{L} := \lambda_1\mathcal{L}_{\text{wMLM}} + \lambda_2\mathcal{L}_{\text{JE2E}} + \lambda_3\mathcal{L}_{\text{PUSM}}. \quad (8)$$

Because the PUSM stage requires different batch sizes, training follows a simple iterative schedule: (1) train with fixed boundaries and wMLM only. (2) enable JE2E to refine segmentation, and (3) switch to PUSM for explicit distribution matching. To restrict the capacity of the shared representation during training, we use a shallow encoder and partial quantization of encoder states before the post-nets with random mix-up (Ao et al., 2022).

4.2 Inference

During inference, the ASR system cascades the speech syllabifier, the speech pre-net, and the text post-net:

$$\begin{aligned}y(X) &:= \arg\max_{Y\in\mathcal{Y}^L} q_Y(Y|X) \\ &:= \arg\max_{Y\in\mathcal{Y}^L} (g_Y \circ f_X(X))(Y). \quad (9)\end{aligned}$$

We also find that replacing each <OOV> token with the second most likely prediction reduces CER by about 1% relative to simply discarding <OOV>s.

5 Experiments

We first describe the datasets and preprocessing, then present the main UASR, boundary-detection, semi-supervised, and ablation results. Additional implementation details are deferred to Appendix D.

5.1 Datasets

We evaluate SylCipher on six datasets: LibriSpeech (460 hours of clean read English) (Panayotov et al., 2015a), SpokenCOCO (742 hours of spoken image captions) (Hsu et al., 2021b), Mandarin AISHELL-3 (85 hours) (Shi et al., 2021), multilingual LibriSpeech (MLS) for German, Dutch and French (Pratap et al., 2020), and a low-resource Cantonese-Taigi setting built from MDCC (Yu et al., 2022) and SuiSiann (Ì-thuân, 2019). For MLS, we subsample 100 hours per language following prior work (Tseng et al., 2024), append a language ID token to each tokens as well as the start of each speech and text unit sequences, train the MLM stage jointly, and run the PUSM stage monolingually.

We follow the LibriSpeech split of (Ni et al., 2025), and the standard splits elsewhere. For LibriSpeech and SpokenCOCO, we report both a *matched* setting, where empirical speech and text probability distributions can be matched exactly, and an *unmatched* setting, where they cannot. In the matched case, pairings are removed from paired corpora; in the unmatched case, LibriSpeech uses LibriLM (Panayotov et al., 2015b) with overlapping text removed, while SpokenCOCO is split into

speech-only and text-only halves. For the remaining datasets, we report only the matched setting. We apply a voice activity detector[2] to all datasets.

5.2 Speech and text syllabification

For English text, we use Pyphen[3], a rule-based hyphenation tool repurposed for syllabification without a G2P, together with a simple fallback for long words that remain unsplit (Appendix F); We refer to this combination as Pyphen+. We also test other G2P-free approaches such as byte-pair encoding (BPE) (Liu et al., 2025; Sennrich et al., 2016), and find SylCipher robust to syllabification noise. For English, we keep only the top-2048 most frequent English syllables and replace the rest with a special <OOV> token (replacing 7% of tokens in LibriSpeech and 2% in SpokenCOCO). Similar syllabification procedure is used for MLS but with top-1024 syllables per language. For Mandarin, we use the pinyin or raw character as syllables. We keep the top-1024 types, covering 99.5% of occurrences in the case of pinyin and 92% in the case of raw character. Similar procedure is used for Cantonese and Taigi. For speech, we cluster syllable-level features obtained by mean pooling within Syllable boundaries, using a codebook size equal to the number of non-<OOV> text tokens.

5.3 Results: UASR

We compare SylCipher with word-, character-, and syllable-level UASR systems that differ in tokenization, objective, and architecture: word-level JSTTI (Ni et al., 2025); character-level wav2vec-U (Baevski et al., 2021), REBORN (Tseng et al., 2024), and a phone-level JSTTI variant; and syllable-level PUSM (Wang et al., 2023c). We also report *self-training* (Chen et al., 2019; Baevski et al., 2021), where a wav2vec 2.0 (Baevski et al., 2020) student is finetuned on pseudo-labels from each UASR system. We use CER for European languages and PER for Sinitic languages, both are tokenization-independent and easily comparable.

Syllable-level modeling performs best under G2P-free setting. Table 1a summarizes results on LibriSpeech. Among baselines, PUSM performs best in the matched setting, while wav2vec-U is strongest in the unmatched setting. REBORN trained directly on characters degrades substantially, suggesting sensitivity to speech-text misalignment. SylCipher with all three stages (Sylber+JE2E+PUSM) achieves 21.8% CER (matched) and 35.9% (unmatched), outperforming all baselines. Relative to wav2vec-U (35.6%/43.3%), this is a 40%/17% CER reduction. Syllable-level models consistently outperform word- and character-level systems, supporting the view that syllables are the most effective alignment unit. Across stages, JE2E brings modest gains, while PUSM yields the largest improvement (44% and 23% relative over JE2E in matched and unmatched settings), and self-training further reduces CER by 20% relative. Unsupervised Sylber boundaries also perform nearly as well as forced alignment.

Syllables are robust to domain shifts. Table 1b reports results on SpokenCOCO. SylCipher remains strongest under domain shift, outperforming PUSM by 32% and wav2vec-U by 49% relative CER after all stages; even the early stages already beat wav2vec-U by 22%. The gap is larger than on LibriSpeech, suggesting that syllable units are more robust to domain shifts for English. Self-training

Model	Student	Token	Matched CER (↓)	Unmatched CER (↓)
G2P-based approach				
wav2vec-U* (Baevski et al., 2021)	No	Phone	-	13.3
wav2vec-U 2.0* (Liu et al., 2023)	No	Phone	-	12.2
REBORN* (Tseng et al., 2024)	No	Phone	-	8.3
G2P-free approach				
wav2vec-U (Baevski et al., 2021)	No	Char.	35.6	43.3
	Yes	Char.	33.8	42.1
REBORN (Tseng et al., 2024)	No	Char.	37.8	76.6
JSTTI (forced align)	No	Char	81.4	81.1
JSTTI (Ni et al., 2025)	No	Word	49.5	54.2
PUSM (Wang et al., 2023c)	No	Syllable	35.5	57.7
	Yes	Syllable	33.0	54.7
SylCipher (Ours, forced align)	No	Syllable	38.5	46.4
SylCipher (Ours, Sylber)	No	Syllable	43.5	48.6
SylCipher (Ours, Sylber+JE2E)	No	Syllable	39.2	46.8
SylCipher (Ours, Sylber+JE2E+PUSM)	No	Syllable	**21.8**	**35.9**
	Yes	Syllable	**17.5**	**33.3**

(a) UASR results on LibriSpeech (clean subsets)

Model	Student	Token	Matched CER (↓)	Unmatched CER (↓)
wav2vec-U (Baevski et al., 2021)	No	Char.	45.0	45.2
	Yes	Char.	46.3	35.3
JSTTI	No	Char.	78.3	100
JSTTI (Ni et al., 2025)	No	Word	64.5	64.5
PUSM (Wang et al., 2023c)	No	Syllable	41.5	41.3
	Yes	Syllable	34.7	34.3
SylCipher (Ours, Sylber)	No	Syllable	34.9	36.1
SylCipher (Ours, Sylber+JE2E)	No	Syllable	31.2	32.4
SylCipher (Ours, Sylber+JE2E+PUSM)	No	Syllable	**23.4**	**26.8**
	Yes	Syllable	**13.9**	**17.6**

(b) UASR results on SpokenCOCO

Table 1: **UASR results on LibriSpeech (clean subsets) and SpokenCOCO**. Inside the bracket lists the unsupervised boundary used for each model. *Student* stands for the student model used during the self-training stage using pseudo-labels from each model. For tokens used for the text data, *Char.* stands for characters and *Syllable* stands for syllable-level tokens converted using the Pyphen+ syllabifier without using a G2P. CER stands for character error rate. * indicates evaluation on LibriSpeech dev-clean instead.

[2]https://github.com/wiseman/py-webrtcvad.git
[3]https://github.com/Kozea/Pyphen

Model	Student	Token	w/o Tone PER (↓)	w. Tone PER (↓)
wav2vec-U (Baevski et al., 2021)	No	Phone	74.9	76.2
JSTTI	No	Init./Final	96.4	100
JSTTI (Ni et al., 2025)	No	Word	83.2	169
PUSM (Wang et al., 2023c)	No	Syllable	28.4	26.5
	Yes	Syllable	18.5	14.9
SylCipher (Ours, forced align)	No	Syllable	38.1	38.9
SylCipher (Ours, Sylber)	No	Syllable	44.6	48.3
SylCipher (Ours, Sylber+JE2E)	No	Syllable	41.7	45.1
SylCipher (Ours, Sylber+JE2E+PUSM)	No	Syllable	**26.9**	**24.9**
	Yes	Syllable	**15.3**	**12.2**
SylCipher (Ours, Sylber+JE2E+PUSM)	No	Char.	31.9	29.9

Table 2: **UASR results on AISHELL-3 test set**. Inside the bracket lists the unsupervised boundary used for each model. "Student" stands for the student model used during the self-training stage using pseudo-labels from each model. *Init./Final* refers to initials and finals in the Chinese phonetic alphabet. For text data tokens, *Syllable* refers to the pinyin representation of each Chinese character, and *Phone* denotes the individual letters within the pinyin tokens, and *Char.* stands for the raw Chinese character. PER stands for phone error rate.

Model	Token	de (↓)	nl (↓)	fr (↓)
wav2vec-U (Baevski et al., 2021)	Phone	32.5	40.2	39.8
wav2vec-U (Baevski et al., 2021)	Char.	42.9	50.6	72.6
PUSM (Wang et al., 2023c)	Syllable	35.1	38.2	43.6
SylCipher (Ours, Sylber)	Syllable	**38.2**	**40.9**	**46.3**
SylCipher (Ours, Sylber+PUSM)	Syllable	**32.3**	**35.6**	**41.7**

Table 3: **Multilingual UASR results on MLS**. *de*, *nl* and *fr* stand for German, Dutch and French respectively. Performance is measured by phone/character error rate (lower is better), depending on the type of token used.

yields a further 39% relative CER reduction, and the small matched/unmatched gap suggests that the sharper LibriSpeech degradation mainly comes from speech-text domain mismatch.

Syllable-level UASR works under multilingual setting. Table 3 show that multilingual training on related languages is effective: on MLS, SylCipher improves over strong monolingual G2P-free baselines by 25-50% relative to wav2vec-U and 5-8% relative to PUSM, and even beats the G2P-based baseline in Dutch by 12%.

Syllable-level UASR works across (low-resource) languages. Table 2 shows that syllable-level UASR is particularly well-suited to Mandarin: syllable models converge even without boundary refinement, whereas phone- and word-level systems struggle. Relative to PUSM, SylCipher improves PER by more than 5.5% before self-training and 17% after; adding tone labels does not hurt, and often helps after the PUSM stage. Similar trends appears in Table 4 for Cantonese and Taigi, where SylCipher clearly outperforms the GAN-based character baseline and remain competitive even with

Model	Student	Token	Cantonese (↓)	Taigi (↓)
wav2vec-U (Baevski et al., 2021)	No	Phone	75.7	69.0
PUSM (Wang et al., 2023c)	No	Syllable	36.6	37.3
SylCipher (Ours, PUSM+MLM)	No	Syllable	48.8	58.4
SylCipher (Ours, PUSM+MLM+JE2E)	No	Syllable	37.7	38.6
SylCipher (Ours, PUSM+MLM+JE2E+PUSM)	No	Syllable	**35.0**	**35.5**
	Yes	Syllable	**23.5**	**26.5**
SylCipher (Ours, PUSM+MLM+JE2E+PUSM)	No	Char.	46.9	-

Table 4: **Multilingual UASR results on two low-resource languages, Cantonese (MDCC) and Taigi (SuiSiann)**. Inside the bracket lists the unsupervised boundary used for each model. "Student" stands for the student model used during the self-training stage using pseudo-labels from each model. *Syllable* refers to the JyutPing (or analogous notation for Taigi) representation of each Cantonese/Taigi character, and *Phone* denotes the individual letters within the JyutPing tokens, while *Char.* stands for the raw characters of the languages. Performance are measured by phone error rate (lower is better).

	$F1^{50}$ (↑)	$F1^{20}$ (↑)	P. (↑)	Re. (↑)	R (↑)
Feat-Sim (Peng et al., 2023)	47.3	24.7	46.6	48.0	54.4
SDHuBERT (Cho et al., 2024)	66.1	32.2	64.9	67.4	70.7
SylBoost (Baade et al., 2025)	73.2	44.6	72.1	74.4	76.9
Sylber (Cho et al., 2025)	83.4	44.1	84.8	84.1	86.4
SylCipher (Ours, Sylber+JE2E)	**86.1**	**50.8**	**86.6**	**86.1**	**88.1**

(a) Boundary detection results on LibriSpeech

	$F1^{50}$ (↑)	$F1^{20}$ (↑)	P. (↑)	Re. (↑)	R (↑)
Feat-Sim (Peng et al., 2023)	60.3	-	57.4	63.6	64.3
SylBoost (Baade et al., 2025)	55.6	30.2	48	**65.2**	50.8
Sylber (Cho et al., 2025)	53.5	28.6	52.3	54.9	59.6
SylCipher (Ours, Sylber+JE2E)	**62.3**	**31.0**	**61.1**	63.6	**67.4**

(b) Boundary detection results on SpokenCOCO

Table 5: **Syllable boundary detection results on LibriSpeech and SpokenCOCO**. The superscript for F1 scores is tolerance threshold in ms and the tolerance is 50ms for other metrics. P., Re. and R stand for precision, recall and R-value respectively. SylCipher is trained under unmatched settings.

raw characters. In these extremely low-resource settings, however, we find it necessary to pretrain with PUSM before MLM (more details in Appendix D), and the gap to PUSM narrows, likely because limited data makes higher-order speech-text matching harder. Multilingual training also stabilizes the MLM stage; for example, Taigi-only training diverges.

5.4 Results: Unsupervised syllable boundary detection

To isolate the effect of JE2E, we compare its boundary predictions against its teacher Sylber (Cho et al., 2025) and other unsupervised approaches including Feat-Sim (Peng et al., 2023), SDHuBERT (Cho et al., 2024), SylBoost (Baade et al., 2025). Sylber is the strongest baseline on most LibriSpeech metrics, while SylBoost is strongest on LibriSpeech $F1^{20}$ and three of five SpokenCOCO metrics. How-

Domain	Ratio	CER
None	0.0	33.0
News	0.5	39.4
News	0.75	48.3
Wikipedia	0.5	42.8
Wikipedia	0.75	57.1
Image Caption	0.25	37.0
Image Caption	0.5	75.6

(a) **Effect of speech-text domain mismatch.** A PUSM with XEUS backbone is trained on mixed LibriSpeech text with text from three out-of-domain (OOD) sources: news (from NewsCrawl), Wikipedia and image captions (from MS COCO), using different proportions of OOD text.

Backbone	LibriSpeech (↓)	MLS (german) (↓)
HuBERT	35.5	45.5
XEUS	**33.4**	**35.1**

(b) **Effect of speech encoder backbone choices.** The model used is a syllable-level PUSM and the evaluation metric used is CER.

Figure 3: **Ablation studies** on the effect of speech-text domain mismatch and speech encoder backbone choices for syllable-level UASR.

ever, SylBoost often over-segments, which hurts cross-modal alignment and destabilizes SylCipher, whereas Sylber predicts syllable counts closer to ground truth and is therefore a better initializer. JE2E improves over Sylber by +14% relative $F1^{20}$ and +3% relative $F1^{50}$ on LibriSpeech, and by +15% F1 and +11% R-value on SpokenCOCO. It also outperforms the best SpokenCOCO baseline, Feat-Sim, by +3% relative F1 score and +4.6% R-value. This suggests that unpaired text provides useful boundary supervision. Speech-text alignments are shown in Appendix H.

5.5 Results: Semi-supervised ASR

We also evaluate a semi-supervised setting with limited paired data by finetuning the self-training student models on labeled subsets and comparing them with the same students before self-training. At inference time, we optionally use a 4-gram LM and tune dataset-specific decoding hyperparameters; a beam size of 5 is sufficient throughout. Figure 4 shows that, without an LM, SylCipher improves over the baselines by 20–100% on English with up to 20 hours of labeled data and by roughly 7–10% on Cantonese and Taigi with up to 2 hours. The same trend holds with an LM, although the gap closes faster as more labeled data becomes available. These results highlight the value of decipherment-based pretraining in low-resource supervised ASR.

5.6 Ablation studies

We ablate the effects of speech-text domain mismatch and speech encoder choice on syllable-level UASR in Figure 3 and the effects of syllabifier, soft-pooler, and vocabulary size in Figure 5.

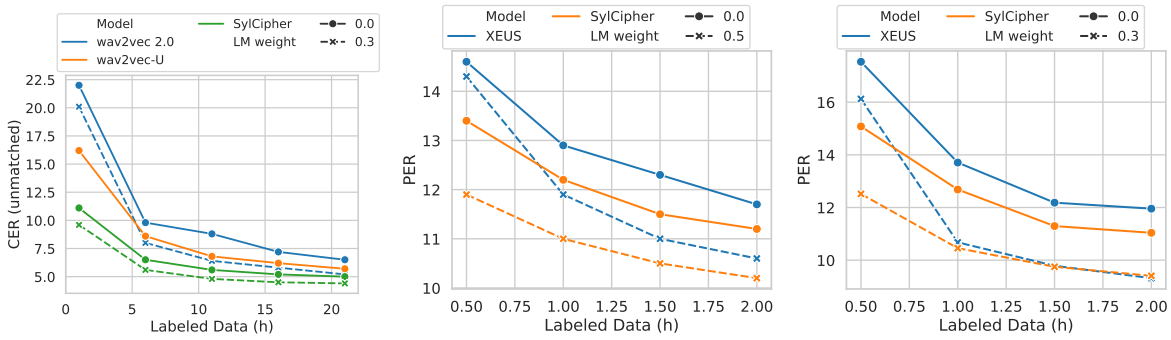
Level of speech-text domain mismatch. As shown in Figure 3a, replacing part of the training corpus with out-of-domain text consistently degrades performance, with the deterioration becoming more pronounced as the proportion of out-of-domain text increases. News text raises the CER to 39.4 and 48.3 at ratios of 0.5 and 0.75, respectively, while Wikipedia text produces even larger increases to $42.8 \rightarrow 57.1$. Image captions cause the most severe degradation: although a ratio of 0.5 and 0.75, respectively, while Wikipedia text produces even larger increases to $42.8 \rightarrow 57.1$. Image captions cause the most severe degradation: although a ratio of 0.25 results in a moderate CER of 37.0, increasing the ratio to 0.5 raises the CER dramatically to 75.6. These results show that syllable-level UASR is sensitive not only to the amount of mismatched text but also to the nature and severity of the domain mismatch.

Speech backbone choice. As shown in Figure 3b, replacing HuBERT with XEUS reduces the CER from 35.5 to 33.4 on LibriSpeech and from 45.5 to 35.1 on German MLS. The considerably larger improvement on German MLS suggests that XEUS provides more robust and transferable speech representations, particularly in the cross-lingual setting.

Syllabifier choice. We compare Pyphen+ with the phoneme-based Syllabify[4] and a lightweight approach BPE-based syllabifier (BPE+) derived from the first stage of SuperBPE (Liu et al., 2025) with a vocabulary of 17k, roughly matching the number of syllable types in LibriSpeech. More details are in Appendix G. All variants are trained only in the *fixed-boundary stage*, and report syllable error rate (SER) to focus on syllable-level behavior. Pyphen+ gives the best SER, outperforming even Syllabify, while BPE+ remains competitive, showing that SylCipher is robust even with simple, resource-light tokenizers.

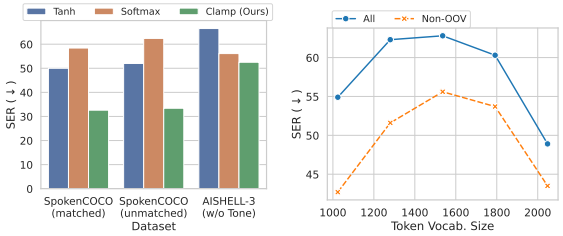
Soft-pooler design. Figure 5a compares our clamp-based soft-pooler with $\tanh$- and softmax-based alternatives (Bhati et al., 2022; Wang et al., 2024). Using $\epsilon = 0.5$ for Clamp and 0.1 for the others,

[4] https://github.com/kylebgorman/syllabify



(a) LibriSpeech (English) (b) MDCC (Cantonese) (c) SuiSiann (Taigi)

Figure 4: **Semi-supervised ASR results on various spoken languages**. A small subset of labeled data is randomly sampled from the training set to simulate the low-resource setting. "LM weight" stands for the fusion weight of a 4-gram language model during beam search. For each dataset, all methods share the same model architecture and hyperparameter settings, except our approach uses an additional unpaired speech-text self-training stage before the supervised finetuning. Lower is better for all metrics.



(a) SER vs. pooler type across datasets (b) SER vs. token vocabulary size

Model	Tokenizer	Language	SER (↓)	CER (↓)
SylCipher (Sylber)	Pyphen+	English	48.9	-
SylCipher (Sylber+JE2E)	Pyphen+	English	44.9	-
SylCipher (Sylber)	Syllabify	English	51.3	-
PUSM	Pyphen+	German	46.3	35.1
PUSM	Pyphen+	Dutch	48.7	38.2
PUSM	Pyphen+	French	52.7	43.6
SylCipher (Sylber)	BPE+	English	52.5	-
SylCipher (Sylber+JE2E)	BPE+	English	49.9	-
PUSM	BPE+	German	54.3	39.7
PUSM	BPE+	Dutch	54.4	39.8
PUSM	BPE+	French	64.3	48.3

(c) Effect of syllabifier type

Figure 5: **Ablation studies** on the effect of syllabifier type, soft-pooler design and token vocabulary size on SylCipher UASR performance.

Clamp consistently improves SER by 6-36% on SpokenCOCO and AISHELL-3, indicating more stable training.

Vocabulary size. Figure 5b shows that SylCipher is robust across a wide range of non-<OOV> vocabulary sizes, with the best overall SER and best non-<OOV> SER at 2048 and 1024 tokens.

6 Conclusion

In this work, we introduced SylCipher, a UASR system that avoids phoneme-level resources such as G2P by recognizing speech at the *syllable level*. Under the G2P-free setting, SylCipher outperforms the best existing systems by 17-40% relative CER on LibriSpeech and shows generalizability across other domains on SpokenCOCO, narrowing the gap with G2P-based systems. It also demonstrates cross-lingual robustness, including Mandarin and two low-resource languages where phoneme-based methods fail to converge, SylCipher achieves a PER of 13%. These results suggest that syllable-level modeling is a viable alternative to phoneme-level UASR and can push the horizon of more accessible and inclusive spoken language technology.

Acknowledgments

This work was supported in part by NTT, Inc. We thank Junrui Ni for his assistance with setting up the JSTTI codebase.

Limitations

SylCipher is not yet language-universal, since different languages use different writing systems and require linguistic knowledge to properly syllabify. For example, languages such as Hebrew and Arabic omits vowels in their writing system, which could pose challenges to existing syllabifiers. While our experiments show that the method can be adapted to more resource-efficient tokenizers such as BPE with minimal modifications, coming up with a language-universal tokenization method remains an open problem. Further, the iterative training procedure can be further simplified into an end-to-end approach. Lastly, improving the robustness of SylCipher under domain mismatch between speech

and text remains an open challenge.

Use of AI assistants. We used AI assistants only for language polishing and improving clarity. All technical content, claims, experiments, and conclusions were developed and verified by the authors.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5723–5738. Association for Computational Linguistics.

Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. 2025. On the landscape of spoken language models: A comprehensive survey. *TMLR*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Alan Baade, Puyuan Peng, and David Harwath. 2025. Syllablelm: Learning coarse semantic units for speech language models. In *ICLR*.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27826–27839.

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*, pages 12449–12460.

Saurabhchand Bhati, Jesús Villalba, Piotr Zelasko, Laureano Moro-Velázquez, and Najim Dehak. 2022. Unsupervised speech segmentation and variable rate representation learning using segmental contrastive predictive coding. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:2002–2014.

Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-yi Lee, and Lin-Shan Lee. 2019. Completely unsupervised phoneme recognition by a generative adversarial network harmonized with iteratively refined hidden markov models. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1856–1860. ISCA.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.*, 16(6):1505–1518.

William Chen, Wangyou Zhang, Yifan Peng, Xinjian Li, Jinchuan Tian, Jiatong Shi, Xuankai Chang, Soumi Maiti, Karen Livescu, and Shinji Watanabe. 2024. Towards robust speech representation learning for thousands of languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10205–10224. Association for Computational Linguistics. Preprint available on arXiv: arXiv:2407.00837.

Cheol Jun Cho, Nicholas Lee, Akshat Gupta, Dhruv Agarwal, Ethan Chen, Alan W. Black, and Gopala K. Anumanchipalli. 2025. Sylber: Syllabic embedding representation of speech from raw audio. In *ICLR*, Singapore.

Cheol Jun Cho, Abdelrahman Mohamed, Shang-Wen Li, Alan W Black, and Gopala K Anumanchipalli. 2024. Sd-hubert: Sentence-level self-distillation induces syllabic organization in hubert. In *ICASSP*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech recognition. In *Interspeech 2021*, pages 2426–2430.

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*, 2 edition. Wiley-Interscience.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Eleni Drinea and Adam Kirsch. 2007. Directly lower bounding the information capacity for channels with i.i.d. deletions and duplications. In *2007 IEEE International Symposium on Information Theory*, pages 1731–1735.

A. Farukh and Mila Vulchanova. 2014. Predictors of reading in urdu: Does deep orthography have an impact? *Dyslexia*, 20(2):146–166.

Tzeviya Sylvia Fuchs and Yedid Hoshen. 2023. Unsupervised word segmentation using temporal gradient pseudo-labels. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

James Glass. 2012. Towards unsupervised speech processing. In *International Conference on Information Sciences, Signal Processing and their Applications*.

Shafi Goldwasser, David F. Gruber, Adam Tauman Kalai, and Orr Paradise. 2023. A theory of unsupervised translation motivated by understanding animal communication. In *Neural Information Processing Systems*.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021a. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460.

Wei-Ning Hsu, David Harwath, Chunxi Song, and James Glass. 2021b. Text-free image-to-speech synthesis using learned segmental units. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5820–5831.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP*, pages 7669–7673.

Jason Kandybowicz and Harold Torrence, editors. 2017. *Africa's Endangered Languages: Documentary and Theoretical Approaches*. Oxford University Press.

Zohaib Ahmad Khan, Yuanqing Xia, Fiza Khaliq, Javed Ali Khan, and Nek Dil Khan. 2024. From tradition to technology: A systematic survey on navigating pashto in modern NLP. SSRN. Available at SSRN.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Ido Levy, Orr Paradise, Boaz Carmeli, Ron Meir, Shafi Goldwasser, and Yonatan Belinkov. 2025. Unsupervised translation of emergent communication. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)*, Philadelphia, PA, USA. AAAI Press.

Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2023. Towards end-to-end unsupervised speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 221–228.

Alexander H. Liu, Cheng-I Jeff Lai, Wei-Ning Hsu, Michael Auli, Alexei Baevski, and James Glass. 2022. Simple and effective unsupervised speech synthesis. In *Interspeech 2022*, pages 843–847. ISCA.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A Smith, and Yejin Choi. 2025. SuperBPE: Space travel for language models. In *Second Conference on Language Modeling*.

Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee. 2018. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings. In *Interspeech*.

Shuang Ma, Daniel McDuff, and Yale Song. 2019. Unpaired image-to-speech synthesis with multimodal information bottleneck. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7515–7524.

Abdelrahman Mohamed, Hung yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, and 1 others. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Junrui Ni, Liming Wang, Heting Gao, Kaizhi Qian, Yang Zhang, Shiyu Chang, and Mark Hasegawa-Johnson. 2022. Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition.

In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 461–465. ISCA.

Junrui Ni, Liming Wang, Yang Zhang, Kaizhi Qian, Heting Gao, Mark Hasegawa-Johnson, and Chang D. Yoo. 2025. Towards unsupervised speech recognition without pronunciation models. *Speech Communication*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015a. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*, pages 5206–5210.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015b. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE.

Puyuan Peng and David Harwath. 2022. Word discovery in visually grounded, self-supervised speech models. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2823–2827. ISCA.

Puyuan Peng, Shang-Wen Li, Okko Räsänen, Abdelrahman Mohamed, and David Harwath. 2023. Syllable segmentation and cross-lingual generalization in a visually grounded, self-supervised speech model. In *Interspeech*.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *ArXiv*, abs/2012.03411.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jianbo Shi and Jitendra Malik. 1997. Normalized cuts and image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 731–737.

Jiatong Shi, Chan-Jan Hsu, Holam Chung, Dongji Gao, Paola Garcia, Shinji Watanabe, Ann Lee, and Hung yi Lee. 2023. Bridging speech and text pre-trained models with unsupervised asr. In *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2021. Aishell-3: A multi-speaker mandarin tts corpus. In *Interspeech 2021*, pages 2756–2760.

Liang-Hsuan Tseng, En-Pei Hu, David Cheng-Han Chiang, Yuan Tseng, Hung-yi Lee, Lin-Shan Lee, and Shao-Hua Sun. 2024. REBORN: reinforcement-learned boundary segmentation with iterative training for unsupervised ASR. In *Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*, pages 5998–6008.

Changhan Wang, Hirofumi Inaguma, Peng-Jen Chen, Ilia Kulikov, Yun Tang, Wei-Ning Hsu, Michael Auli, and Juan Pino. 2023a. Simple and effective unsupervised speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10771–10784, Toronto, Canada. Association for Computational Linguistics.

Liming Wang, Mark Hasegawa-Johnson, and Chang Yoo. 2023b. A theory of unsupervised speech recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1215, Toronto, Canada. Association for Computational Linguistics.

Liming Wang, Mark Hasegawa-Johnson, and Chang D. Yoo. 2024. Unsupervised speech recognition with n-skipgram and positional unigram matching. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 10936–10940. IEEE.

Liming Wang, Junrui Ni, Heting Gao, Jialu Li, Kai Chieh Chang, Xulin Fan, Junkai Wu, Mark Hasegawa-Johnson, and Chang Dong Yoo. 2023c. Listen, decipher and sign: Toward unsupervised speech-to-sign language recognition. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6785–6800. Association for Computational Linguistics.

Tiezheng Yu, Rita Frieske, Peng Xu, Samuel Cahyawijaya, Cheuk Tung Shadow Yiu, Holy Lovenia, Wenliang Dai, Elham J. Barezi, Qifeng Chen, Xiaojuan Ma, Bertram E. Shi, and Pascale Fung. 2022. Automatic speech recognition datasets in cantonese: A survey and new dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

Yaodong Zhang and James R. Glass. 2009. Speech rhythm guided syllable nuclei detection. In *ICASSP*, pages 3797–3800.

Ì-thuân. 2019. Suísiann Dataset. https://suisiann-dataset.ithuan.tw/. Accessed: 2026-01-22.

A Theoretical guarantee of SylCipher

We state our main theorem below.

Theorem A.1. *Let $(f^*_{\tilde{X}}, g^*_{\tilde{X}}, f^*_Y, g^*_Y)$ be a minimizer of equation 7 under the constraint that $H(f_Z(Z)) \leq H(Y)$, and suppose the following assumptions hold:*

1. *The speech feature sequence X is a sequence of one-hot vectors with $|\mathcal{X}| = |\tilde{\mathcal{X}}| = |\mathcal{Y}|$;*
2. *The true syllable boundaries are used, i.e., $b_t(X) = \mathbb{1}[y^*(X_t) \neq y^*(X_{t+1})]$;*
3. *The tokenizer c is an optimal K-means quantizer with Euclidean distance metric and cluster size $|\tilde{\mathcal{X}}|$;*
4. *The true ASR y^* is decomposable, i.e., $y^*(\tilde{X}) = [y^*(\tilde{X}_1), \cdots, y^*(\tilde{X}_L)]$;*
5. *$f_{\tilde{X}}$ and f_Y are decomposable;*
6. *Assumption 1-2 in (Wang et al., 2023b) hold for $(\tilde{X}, Y)$.*

Then $f^*_{\tilde{X}}$ and f^*_Y are invertible and $q^*_{Y|X}(y|x) := \mathbb{1}[f^{*-1}_Y \circ f^*_{\tilde{X}} \circ c \circ m(x) = y]$ satisfies

$$D_{\mathrm{KL}}(p_Y || \mathbb{E}_X[q^*_{Y|X}]) = 0,$$
$$y^*(x) = \arg\max_{y \in \mathcal{Y}^L} q^*_{Y|X}(y|x), \quad \forall x \in \mathcal{X}^T.$$

The proof relies on the following lemma proven in Appendix B.

Lemma A.2. *Suppose Assumption 5 of Theorem A.1 and Assumption 1 and 2 of (Wang et al., 2023b) hold for $(\tilde{X}, Y)$, and functions $g_X : \tilde{\mathcal{X}}^L \mapsto [0,1], g_Y : \tilde{\mathcal{Y}}^L \mapsto [0,1]$ of the form in equation 6 be such that $p_X = g_X \circ f_X$ and $p_Y = g_Y \circ f_Y$. Then f_X and f_Y are both invertible.*

Now we are ready to prove the main theorem.

Proof. The proof consists of two main parts: (i) First, we prove that at least one minimizer $(f_{\tilde{X},1}, g_{\tilde{X},1}, f_{Y,1}, g_{Y,1})$ can achieve a minimum of 0 for equation 7; (ii) then we establish that for any minimizers $(f^*_{\tilde{X}}, g^*_{\tilde{X}}, f^*_Y, g^*_Y)$, the marginal distribution of the text posterior $q^*_{Y|X}$ matches the true text distribution p_Y. As a result, by Assumption 4, Theorem 1 of (Wang et al., 2023b) guarantees that $\arg\max_{y \in \mathcal{Y}^L} q^*_{Y|X}(y|X)$ achieves zero-error UASR.

To prove (i), by the definition of y^* and Assumption 1, y^* is a invertible mapping between $\mathcal{X}$ and $\mathcal{Y}$. Then let $t_i = \min\{\tau : \sum_{t=1}^{\tau} b_t(X) \geq i-1\}$ be the *starting time* of each syllable and apply Assumption 2, we have

$$X_s = X_t, \forall t_i \leq s < t < t_{i+1}, 1 \leq i \leq L.$$

As a result, the output of the soft-pooler is simply $m_i(X) = X_{t_i}$. Further, by Assumption 3, c is optimal, and we claim that this is achievable if and only if for any $(x, x') \in \mathcal{X}^2$,

$$c(x) = c(x') \Longleftrightarrow y^*(x) = y^*(x').$$

Otherwise, suppose for some $(x, x') \in \mathcal{X}^2$ such that $c(x) = c(x') =: c_0$ but $y^*(x) \neq y^*(x')$, then for any centroid μ_{c_0} of cluster c_0 and triangle inequality,

$$\begin{aligned}&\|x - \mu_{c(x)}\|_2 + \|x' - \mu_{c(x')}\|_2 \\ =&\|x - \mu_{c_0}\|_2 + \|x' - \mu_{c_0}\|_2 \geq \|x - x'\|_2 > 0,\end{aligned}$$

and therefore the K-means objective:

$$\mathcal{L}_{\mathrm{km}}(\mu_1, \cdots, \mu_{|\mathcal{X}|}) := \mathbb{E}_{X \sim p_X} \|X - \mu_{c(X)}\|_2^2 > 0.$$

However, by simply setting $c(x) = y^*(x)$ for all $x \in \mathcal{X}$, we have $\mu_{c(x)} = x$ for any $x \in \mathcal{X}$ and $\mathcal{L}_{\mathrm{km}} = 0$ due to the invertibility of y^*. This contradicts with the optimality of c and thus proves the "$\Longrightarrow$" direction. Further, this implies that the cluster size is at least $|\mathcal{Y}|$ and since we set the cluster size to $|\mathcal{Y}|$, $c \circ m$ is equivalent to y^* up to a permutation. This then proves the other direction. Let $c \circ m = \pi \circ y^*$ where π is a permutation, then the syllable-level features $\tilde{X} = \pi \circ y^*(X)$ and thus by the permutation-invariance of discrete entropy,

$$H(\tilde{X}) = H(y^*(X)) = H(Y).$$

Therefore, set $f_{\tilde{X},1}$ to be a permutation on $\tilde{\mathcal{X}}$, $f_{Y,1} = f_{\tilde{X},1} \circ \pi$, $g_{\tilde{X},1} = p_{\tilde{X}} \circ f^{-1}_{X,1}$ and $g_{Y,1} = p_Y \circ f^{-1}_{Y,1}$, we have

$$\begin{aligned}&D_{\mathrm{KL}}(p_{\tilde{X}} || g_{\tilde{X},1} \circ f_{\tilde{X},1}) + D_{\mathrm{KL}}(p_Y || g_{Y,1} \circ f_{Y,1}) \\ =&D_{\mathrm{KL}}(p_{\tilde{X}} || p_{\tilde{X}}) + D_{\mathrm{KL}}(p_Y || p_Y) = 0,\end{aligned}$$

and

$$\begin{aligned}&f_{Y,1}(y(X)) \\ =&f_{\tilde{X},1} \circ \pi \circ y(X) = f_{\tilde{X},1} \circ c \circ m(X) \\ =&f_{\tilde{X},1}(\tilde{X}).\end{aligned}$$

The latter implies that the probability distribution of $f_{Z,1}(Z)$ satisfies

$$\begin{aligned} p_{f_{Z,1}(Z)} &= \frac{1}{2}p_{f_{\tilde{X},1}(\tilde{X})} + \frac{1}{2}p_{f_{Y,1}(Y)} \\ &= \frac{1}{2}p_{f_{\tilde{X},1}(\tilde{X})} + \frac{1}{2}p_{f_Y(y(X)),1} = p_{f_{\tilde{X},1}(\tilde{X})}, \end{aligned} \tag{*}$$

and thus

$$H(f_{Z,1}(Z)) = H(f_{Y,1}(Y)) = H(f_{X,1}(\tilde{X})) \leq H(Y).$$

Therefore, by the nonnegativity of D_{KL}, $(f_{\tilde{X},1}, g_{\tilde{X},1}, f_{Y,1}, g_{Y,1})$ is an optimal solution of equation 7 with a minimum of 0, which proves (i).

To prove (ii), we first use (i) to conclude that

$$p_{\tilde{X}} = g^*_{\tilde{X}} \circ f^*_{\tilde{X}},\ p_Y = g^*_Y \circ f^*_Y.$$

Then it amounts to prove that any minimizer $(f^*_{\tilde{X}}, g^*_{\tilde{X}}, f^*_Y, g^*_Y)$ of equation 7 satisfies

$$p_{f^*_Z(Z)} = p_{f^*_X(X)} = p_{f^*_Y(Y)}. \tag{*}$$

Since if this is the case, by Lemma A.2, f^*_Y is invertible and thus for any $y \in \mathcal{Y}^L$,

$$\begin{aligned} & p_{f^{*-1}_Y \circ f^*_X(X)}(y) \\ &\overset{(a)}{=} p_{f^*_X(X)}(f^*_Y(y)) \overset{(b)}{=} p_{f^*_Y(Y)}(f^*_Y(y)) \\ &\overset{(c)}{=} p_Y(y) \Longrightarrow D_{\mathrm{KL}}(p_Y || \mathbb{E}_X q^*_{Y|X}) = 0, \end{aligned}$$

where (b) uses (*) and $(a)(c)$ uses Lemma A.2.

To prove (*), we use the concavity of the discrete entropy $h(p) := -\sum_x p(x)\log p(x)$ and Lemma A.2,

$$\begin{aligned} H(f^*_Z(Z)) &= h\left(\frac{1}{2}p_{f^*_X(X)} + \frac{1}{2}p_{f^*_Y(Y)}\right) \\ &\overset{(d)}{\geq} \frac{1}{2}h(p_{f^*_X(X)}) + \frac{1}{2}h(p_{f^*_Y(Y)}) \\ &\overset{(e)}{=} \frac{1}{2}(H(X) + H(Y)) = H(Y), \end{aligned}$$

with equality if and only if $p_{f^*_Y(Y)} = p_{f^*_X(X)}$, where (d) uses the concavity of the entropy function and (e) uses Lemma A.2. This concludes the proof of (ii). □

B Proof of Lemma A.2

Proof. We prove the lemma by contradiction and focus on proving the invertibility of f_X since the proof is analogous for f_Y. Suppose otherwise f_X is not invertible, then by Assumption 5, there exists $(x', x'') \in \tilde{\mathcal{X}}^2$ such that $f_X(x') = f_X(x'')$ but $x' \neq x''$, then by the definition of g_X,

$$\begin{aligned} p_{X_i}(x') &= \sum_{x_{-i} \in \tilde{\mathcal{X}}^{L-1}} p_X(x_{1:i-1}, x', x_{i+1:L}) \\ &= \sum_{x_{-i} \in \tilde{\mathcal{X}}^{L-1}} \prod_{j<i} g_X(x_j \mid f_X(x_{1:j-1})) \cdot \\ &\qquad g_X\big(x' \mid f_X(x_{1:i-1})\big) \cdot \\ &\prod_{k>i} g_X\big(x_k \mid f_X(x_{1:i-1}), f_X(x'), f_X(x_{i+1:k-1})\big) \\ &= \sum_{x_{-i} \in \tilde{\mathcal{X}}^{L-1}} \prod_{j<i} g_X(x_j \mid f_X(x_{1:j-1})) \cdot \\ &\qquad g_X\big(x'' \mid f_X(x_{1:i-1})\big) \cdot \\ &\prod_{k>i} g_X\big(x_k \mid f_X(x_{1:i-1}), f_X(x''), f_X(x_{i+1:k-1})\big) \\ &= \sum_{x_{-i} \in \tilde{\mathcal{X}}^{L-1}} p_X(x_{1:i-1}, x'', x_{i+1:L}) = p_{X_i}(x''). \end{aligned} \tag{10}$$

Therefore, the positional unigram matrix (PUM)

$$P^X := \begin{bmatrix} p^\top_{X_1} \\ \vdots \\ p^\top_{X_L} \end{bmatrix}$$

is column-rank-deficient. However, Theorem 1 of (Wang et al., 2023b) asserts that if Assumption 1 and 2 of (Wang et al., 2023b) holds, P^X has full column-rank, which is a contradiction. Therefore, $f_X(x') \neq f_X(x'')$ if $x' \neq x''$ and f_X is invertible. □

C Theoretical analysis of SylCipher under clustering and segmentation noise

In practice, the speech cluster sequence X can be noisy such that clusters within each syllabic segment $\tilde{X}_i$ may not all correspond to the true syllable Y_i. These type of cluster noise can be well-modeled by an N-ary symmetric channel (Cover and Thomas, 2006) defined below.

Definition C.1. (N-ary symmetric channel (NSC)) An N-ary symmetric channel $\mathrm{NSC}(N, p)$ with input X and output Y satisfy

$$\Pr[Y = y] = \begin{cases} 1 - p, \text{ if } y = X, \\ \frac{p}{N}, \text{ otherwise.} \end{cases}$$

(a) Number of predicted vs. gold syllables in single utterances

(b) Closest predicted vs. gold syllable segment index

(c) Distribution of the gold-predicted segment index gap

Figure 6: **Reliability analysis of Syllable-level boundaries**. The syllable boundaries are extracted from the LibriSpeech test sets under the matched setup. The true segment index of a timestamp is the position index of the true segment that it belongs to, and only timestamps corresponding to a predicted syllable boundary is considered in the plots. The index gap is the difference between the true and predicted segment index.

Further, define the *transition matrix* of the channel as

$$\Pi_{N,p} := \begin{bmatrix} 1-p & p/N & \cdots & p/N \\ p/N & 1-p & \cdots & p/N \\ \vdots & \vdots & \ddots & \vdots \\ p/N & p/N & \cdots & 1-p \end{bmatrix}.$$

To simplify our analysis, we choose the soft pooler $m(\cdot)$ to output a random feature frame in each predicted segment i, $X_{t'_i}$, rather than a weighted average of all the frame. Then we pass the feature frames independently into an $\mathrm{NSC}(|\mathcal{X}|, \epsilon_c)$ channel with error probability $\epsilon_c > 0$ to create $\tilde{X}_i$. Therefore by definition,

$$\Pr[\tilde{X}_i = x] = \begin{cases} 1-\epsilon_c, \text{ if } x = X_{t'_i}, \\ \frac{\epsilon_c}{|\mathcal{X}|}, \text{ otherwise.} \end{cases}$$

Assuming no segmentation noise, the new speech distribution $\tilde{p}_{\tilde{X}}$ satisfies

$$\tilde{p}_{\tilde{X}}(x) = \sum_{x' \in \mathcal{X}^L} p_{\tilde{X}}(x') \Pi^{\otimes L}_{|\mathcal{X}|,\epsilon_c}(x', x),$$

where $\otimes$ denotes the Kronecker product. We can bound the KL divergence between the $\tilde{P}_{\tilde{X}}$ and $P_{\tilde{X}}$ using Jensen's inequality as:

$$\begin{aligned} D_{\mathrm{KL}}(p_{\tilde{X}} || \tilde{p}_{\tilde{X}}) &\leq \mathbb{E}_{p_{\tilde{X}}} D_{\mathrm{KL}}(I_{|\mathcal{X}|^L} || \Pi^{\otimes L}_{|\mathcal{X}|,\epsilon_c}) \\ &= L D_{\mathrm{KL}}(I_{|\mathcal{X}|} || \Pi_{|\mathcal{X}|,\epsilon_c}) \\ &= L|\mathcal{X}| \log \frac{1}{1-\epsilon_c} = O(L|\mathcal{X}|\epsilon_c). \end{aligned}$$

Thus, by data processing inequality,

$$D_{\mathrm{KL}}(p_Y || \mathbb{E}_{\tilde{p}_{\tilde{X}}} q^*_{Y|X}) \leq D_{\mathrm{KL}}(p_{\tilde{X}} || \tilde{p}_{\tilde{X}}) = O(L|\mathcal{X}|\epsilon_c).$$

Further, for small enough $\epsilon_c > 0$, We can prove a variant of Lemma A.2 using the noisy PUM in place of the clean PUM, the former of which is invertible if and only if the latter is. Therefore, the model is robust against NSC noise and both conclusions from Theorem A.1 hold approximately.

To further verify that the assumptions in Theorem A.1 approximately hold in practice and to develop a more general theory of UASR under segment noise, we visualize the key statistics of the syllable boundaries predicted by Sylber before and after JE2E refinement used in SylCipher, as shown in Figure 6. Figure 6a indicates that the number of predicted and true syllables in each utterance is approximately the same, even for long utterances with more than 80 syllables. Further, Figure 6b and Figure 6c show that the positions of the true and predicted syllables tend to match or within 5 syllables from each other, regardless of the absolute positions of the syllables, suggesting that the insertion/deletion probability of the syllables is low. Motivated by these observations, we propose the following *deletion-replication channel* (Drinea and Kirsch, 2007) model for the syllable segmentation noise.

Definition C.2. (deletion-replication channel (DRC)) A deletion-replication channel $\mathrm{DRC}(P_T)$ with input sequence $X = (X_1, \cdots, X_N)$ and output sequence $Y = (Y_1, \cdots, Y_T)$ satisfies

- $T = \sum_{i=1}^N T_i$, where $T_i \in \{0, \cdots, T_{\max}\}$'s are sampled i.i.d from some distribution P_T;
- For any $i \in \{1, \cdots, N\}$ and $\sum_{j<i} T_j \leq t \leq \sum_{j \leq i} T_j$,
$$Y_t \equiv X_i.$$

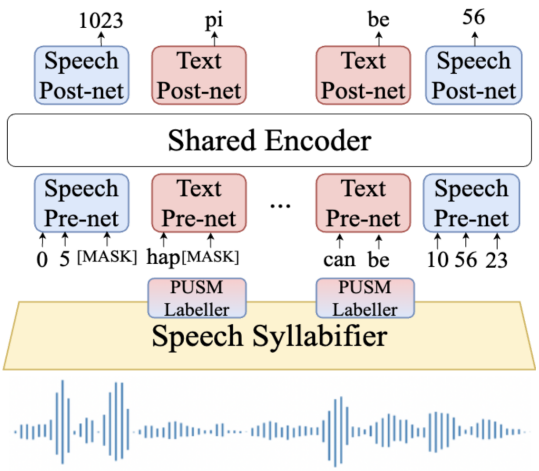


Figure 7: Initialization of SylCipher using interleaving speech tokens and PUSM-predicted text tokens to stabilize training for extreme low-resource scenario.

Further, denote $\epsilon_d := P_T(0)$ as the *deletion probability* and $\epsilon_r := \sum_{t>1} P_T(t)$ as the *replication probability*.

Intuitively, DRC models how the boundary detector i) fails to detect a boundary, leading to deletion and ii) detects multiple boundaries per syllable, leading to replication. Further, from Figure 6c, we can see that ϵ_d and ϵ_r of the syllable segment sequence are both very low. To analyze the effect of DRC on distribution matching, denote the distribution of any N-gram subsequence of the predicted segment sequence $\tilde{X}$ as $\tilde{P}_{\tilde{X}_{-N}\cdots\tilde{X}_0}$ then we have

$$\tilde{p}_{\tilde{X}_{-N}\cdots\tilde{X}_0} = (1-\epsilon_r-\epsilon_d)^N p_{\tilde{X}_{-N}\cdots\tilde{X}_0} + (1-(1-\epsilon_r-\epsilon_d)^N)q,$$

where $p_{\tilde{X}_{-N}\cdots\tilde{X}_0}$ is the true N-gram distribution under true syllable boundaries, and q is some residual distribution created when there is at least one insertion or replication in the N-gram sequence. Then, as long as $(1-\epsilon_r-\epsilon_d)^N \approx 1$, we have $D_{\mathrm{KL}}(P_{\tilde{X}_{-N}\cdots\tilde{X}_0} || \tilde{P}_{\tilde{X}_{-N}\cdots\tilde{X}_0})$ to be low. This also indicates that lower-order N-grams are easier to match than higher-order ones even with infinite data under DRC segmentation noise, as the weight for the true N-gram distribution, $(1-\epsilon_r-\epsilon_d)^N$, decays exponentially as N increases. Nonetheless, using similar arguments, we can show that given enough samples and with sufficiently low ϵ_d and ϵ_r, one can achieve approximate distribution matching for lower-order N-grams and almost zero UASR error rate.

D Implementation details of SylCipher

For English experiments, SylCipher uses a HuBERT-large[5] model pretrained on Libri-Light (Kahn et al., 2020) as the SSL encoder in the speech syllabifier, and initialize the soft-pooler with unsupervised boundary labels from Sylber (Cho et al., 2025). For languages other than English, we instead use XEUS[6] (Chen et al., 2024), which is pretrained on Mandarin (among other languages) and significantly outperforms HuBERT. We also finetune Sylber on AISHELL-3 to ensure stable convergence. For the pre-nets, shared encoder, post-nets, MLM parameters, and most of the optimizer hyperparameters, we follow JSTTI (Ni et al., 2025), as modifying them did not yield consistent improvements. More implementation details can be found in Table 6, 7 and 8.

E Additional experiments: multiple random seeds and comparison with GAN

Table 9 evaluates the robustness of the syllable-level PUSM to random initialization. Across four random seeds, the model obtains CERs between 33.0 and 34.1, with a mean of 33.4±0.5. The small standard deviation and narrow performance range indicate that the proposed training procedure is stable and that its performance is not sensitive to the choice of random seed.

Table 10 compares SylCipher with a syllable-level GAN based on the wav2vec-U architecture and objective. The GAN performs poorly across all tested combinations of code-penalty and gradient-penalty weights, yielding SERs between 115.7 and 126.5. Increasing the code-penalty weight from 2 to 4 further degrades performance, while varying the gradient-penalty weight has a comparatively smaller effect. In contrast, SylCipher achieves an SER of 25.5, corresponding to a 78.0% relative error reduction over the best GAN configuration. These results suggest that directly extending the adversarial wav2vec-U framework to syllable-level recognition is ineffective, whereas SylCipher provides substantially more reliable syllable-level transcription.

When training SylCipher on Taigi, an extreme low-resource language, we use an interleaving language modeling strategy shown in Figure 7 to ini-

[5] https://dl.fbaipublicfiles.com/hubert/hubert_large_ll60k.pt
[6] https://huggingface.co/espnet/xeus/blob/main/model/xeus_checkpoint_new.pth

Component	Configuration
Acoustic features	Layer 21 of HuBERT-Large pretrained on 60k-hour LibriLight or last layer (18) of XEUS encoder
Syllable-boundary detecter	Five-layer 1-D CNN/TDNN with stride 1 and same padding. Kernel widths are $[11, 9, 7, 5, 3]$; channel dimensions are $1024 \rightarrow 500 \rightarrow 500 \rightarrow 500 \rightarrow 500 \rightarrow 500$; ReLU follows each convolution. Two linear heads map 500 dimensions to (i) one sigmoid boundary probability and (ii) 100 frame-level acoustic classes.
Soft-pooler	soft segment assignment/pooling uses sharpness $\epsilon = 0.5$. A fixed K-means codebook converts each pooled word representation to a straight-through hard one-hot word token.
Speech pre-net	Learned speech-token embedding plus scaled positional encoding; embedding/output dimension 768; no additional hidden layer. The input inventory contains the K acoustic syllable tokens and special symbols.
Text pre-net	Learned text-token embedding plus scaled positional encoding; embedding/output dimension 768; no additional hidden layer. The input inventory contains the K written-word types and special symbols.
Shared encoder	Two Transformer encoder layers; model dimension 768; feed-forward dimension 3072; 12 attention heads; GELU activation; dropout 0.1; attention dropout 0.1; activation dropout 0; layer-drop 0.05; relative-position embeddings; BERT-style initialization. Maximum speech/text sequence length is 100 tokens.
Shared mix-up quantizer	Gumbel–Softmax product quantizer with 3 groups and 100 entries per group. Each encoder position is independently replaced by a shared codebook vector with probability 0.3 in the static-boundary SylCipher setting.
Speech post-net	One linear projection from 768 encoder dimensions to speech-token logits (K acoustic types plus special symbols). It is used to reconstruct masked and unmasked speech inputs.
Text post-net	One linear projection from 768 encoder dimensions to text-word logits (K word types plus special symbols). It is used for text reconstruction and speech-to-text prediction.

Table 6: **Architecture used by SylCipher.**

Stage	Supervision	Procedure and output
Codebook preparation	Sylber boundaries	Mean-pool features within Sylber segments; train one K-means model and fix it during training.
Behavior cloning	Sylber boundaries	Train the five-layer CNN with boundary BCE (positive-class weight 1.5) and frame-level acoustic cross-entropy. The acoustic targets are 100-means labels from HuBERT-base layer 6. Both loss weights are 1.

Table 7: **Boundary and codebook initialization**.

tialize the model. First, we train an PUSM model and use it to generate pseudo-text labels for the utterances. Then we uniformly randomly interleave the PUSM text tokens with the speech tokens from the speech syllabifier with an interleaving probability of 0.5. We found that this approach allows the model to learn a more aligned shared latent space for speech and text than naive cross-entropy training.

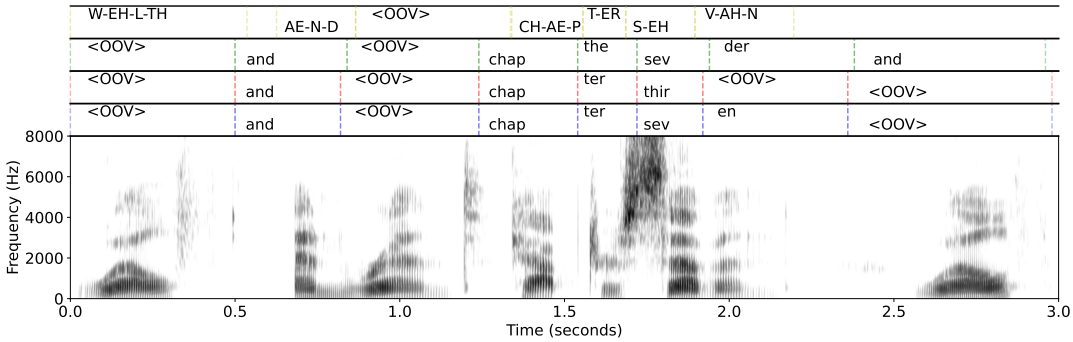
F Pseudo-code for the Pyphen+ syllabifier

See Algorithm 1.

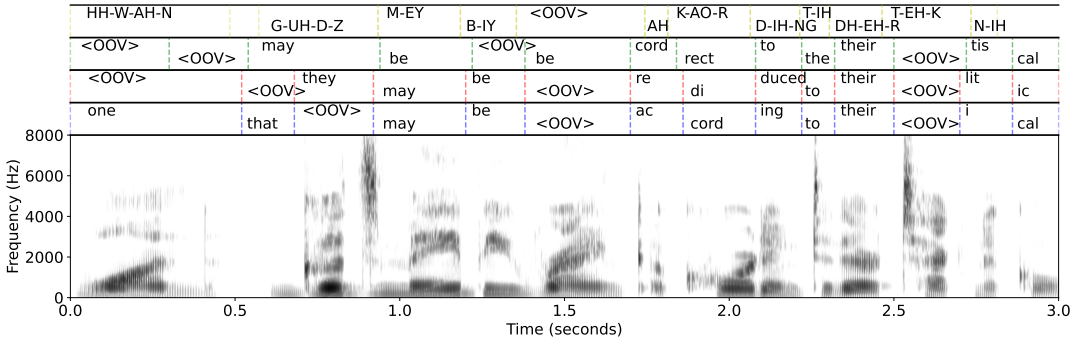
G Pseudo-code of BPE+ syllabifier

See Algorithm 2.

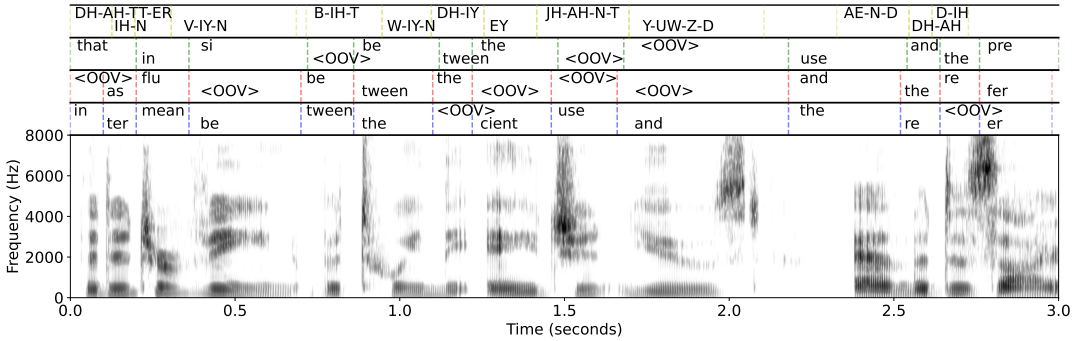
H Spectrogram examples on LibriSpeech



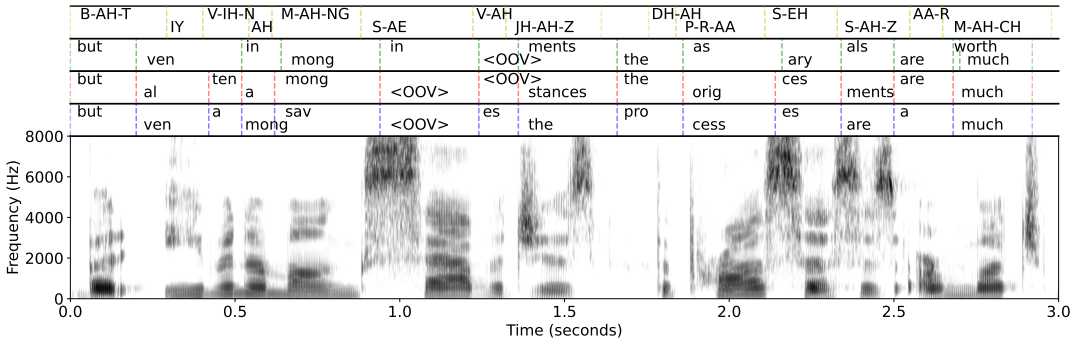
(a) Reference transcript: “Wealth and rent chap-ter sev-en wealth”.



(b) Reference transcript: “One goods may be ranked ac-cord-ing to their tech-ni-cal”.

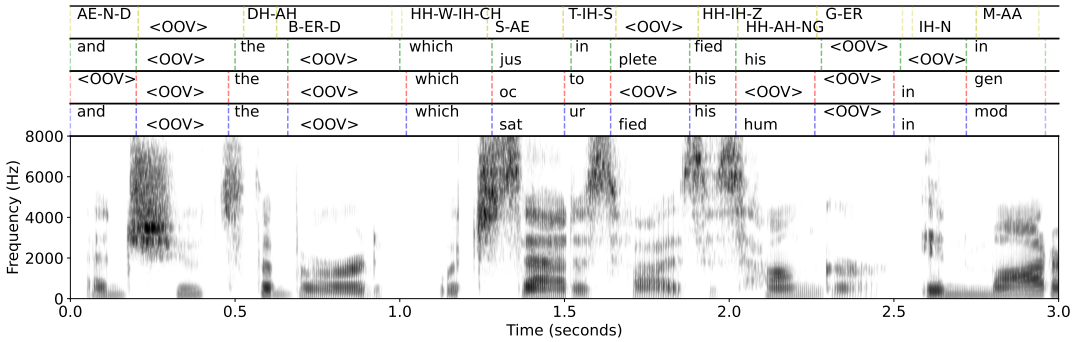


(c) Reference transcript: “That in-ter-vene be-tween the a-gent used and the de-sired”

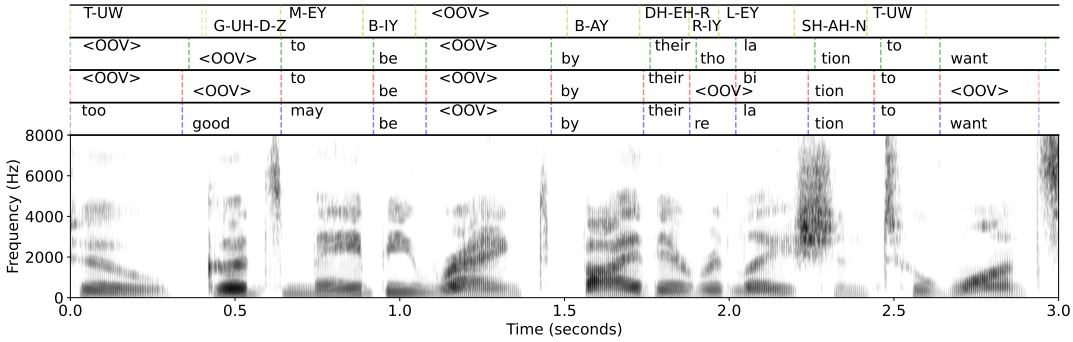


(d) Reference transcript: “But e-ven a-mong sav-ages the pro-cess-es are much”

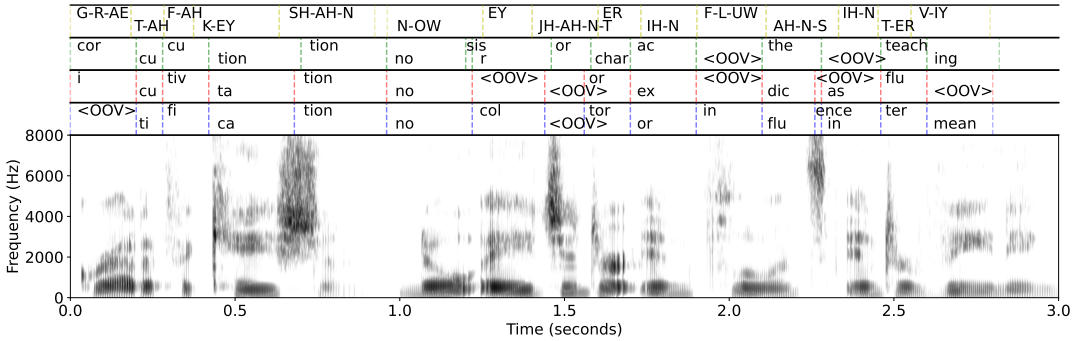
Figure 8: **Spectrograms of audio examples in our test split of LibriSpeech clean subsets (matched setting) and the predicted speech-text alignment by SylCipher after different training stages.** Audios are truncated to the 3-second mark for better visualization. The alignment bars from top to bottom: Forced alignment, Sylber, Sylber+JE2E, Sylber+JE2E+PUSM.



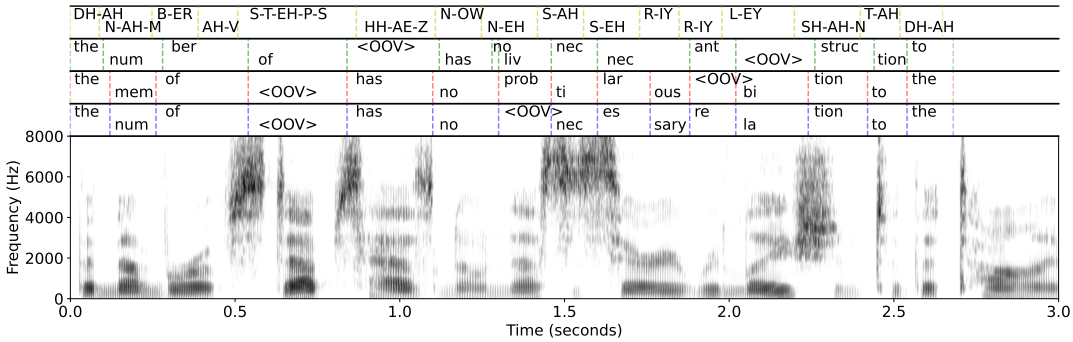
(a) Reference transcript: "And shoots the bird which sat-is-flies his hun-ger".



(b) Reference transcript: "two goods may be ranked by their re-la-tion to wants".



(c) Reference transcript: "Grat-i-fi-ca-tion no a-gent or in-flu-ence in-ter-ven-ing".



(d) Reference transcript: "The num-ber of steps has no nec-es-sary re-la-tion to the".

Figure 9: **Spectrograms of audio examples in our test split of LibriSpeech clean subsets (matched setting) and the predicted speech-text alignment by SylCipher after different training stages.** Audios are truncated to the 3-second mark for better visualization. The alignment bars from top to bottom: Forced alignment, Sylber, Sylber+JE2E, Sylber+JE2E+PUSM.

Experimental setting	Optimization	Schedule, batching, and selection
Fixed-boundary MLM-based training	Adam; learning rate 2×10^{-4}; $\beta = (0.9, 0.98)$; $\epsilon = 10^{-6}$; weight decay 0; gradient-norm clipping 20; polynomial decay (power 1); 150 warm-up updates; maximum/total updates 10^6. Poisson span masking with mean length 3.5 and a mask budget below 30%. Replace 90% of selected positions by <MASK> and 10% by a random token. Masked-token loss weight 1; unmasked-token loss weight 0.5; speech and text modalities weighted equally.	Eight 32-GB V100 GPUs; at most 500 tokens per GPU; update frequency 16; speech:text batch ratio 1:1; seed 0. Random speech and text utterances are paired in each training batch. Shared codebook mix-up probability is 0.3. Train the fixed-boundary stage from scratch for fewer than 7000 epochs; initialize later JE2E stage from the fixed-boundary checkpoint and train for fewer than 1600 epochs and PUSM stage for 2000 epochs (or 2000 steps by using the whole dataset as a single batch). Stop when paired validation SER flattens.
JE2E boundary refinement	Initialize from converged static-boundary SylCipher and CNN checkpoints. Adam with learning rate 2×10^{-4}; text-loss ratio 10; end-to-end policy-loss weight 1 and policy-pretraining weight 0.	Update frequency 890, chosen to produce approximately one optimizer update per epoch for the 2048-word setup. Stop before 700 epochs when validation WER converges.
PUSM training	Optimize the PUSM objective at learning rate 10^{-3}.	Use an update of 7200 to approximately use the whole dataset as a single batch; Codebook mix-up probability is set to 0.
Pseudo-text CTC student	Initialize the unfinetuned 60k-hour wav2vec 2.0 checkpoint. Adam; learning rate 10^{-4}; $\beta = (0.9, 0.98)$; $\epsilon = 10^{-8}$;	One GPU; a batch size of 32 with a one-layer bi-directional LSTM classifier network with 512 hidden units. Select the checkpoint by validation WER.

Table 8: **Optimization parameters and training schedule.**

Seed	CER (↓)
0	33.0
1	33.4
2	33.2
3	34.1
mean±std	33.4±0.5

Table 9: **Effect of random seeds.** The model is a syllable-level PUSM with XEUS backbone trained on LibriSpeech.

Model	λ_{cp}	λ_{gp}	**SER (↓)**
Syllabic GAN	2	1.5	115.8
Syllabic GAN	2	2	115.7
Syllabic GAN	4	1.5	123.7
Syllabic GAN	4	2	126.5
SylCipher	-	-	**25.5**

Table 10: **Comparison with syllable-level GAN model on LibriSpeech**. The GAN uses the same architecture and training objective as wav2vec-U. λ_{cp} stands for the code penalty regularization loss and λ_{gp} stands for the gradient penalty loss, both used by wav2vec-U. SER stands for syllable error rate (lower is better).

Algorithm 1 Merging No-Vowel Segments and Naive Syllabification

```
function MERGE_NOVOWEL_SEGMENTS(segs)
    new_segs ← [ ]
    n_seg ← len(segs)
    i ← 0
    while i < n_seg do
        seg ← segs[i]
        if seg has no vowels then
            if seg is the last segment then
                merge seg into previous segment (if any)
            else
                merge seg into the next segment
                i ← i + 1
            end if
        else
            append seg to new_segs
        end if
        i ← i + 1
    end while
    return new_segs
end function
function NAIVE_SYLLABIFY(w)
    w0 ← w
    if w ends with a silent 'e' (not "-le") then
        drop the final 'e' in w
    end if
    if w ends with "-ed" and root does not end with 't' or 'd' then
        drop the 'e' in "-ed" in w
    end if
    Apply rule to w: split V C-CC... V into VC-CC...V
    Apply rule to w: split VCCV into VC-CV
    Apply rule to w: split VCV into V-CV
    if modifications were made then
        if w0 ended with 'e' then
            add back 'e' to w
        else if w0 ended with 'ed' then
            restore 'ed' to w
        end if
    end if
    return syllabified w
end function
function PYPHEN_PLUS_SYLLABIFY(w)
    syls ← []
    segs ← wordsegment.segment(w)
    for each seg in segs do
        syls_ ← pyphen.inserted(seg)
        syls_ ← merge_novowel_segments(syls_)
        if syls_ has only one syllable and syllapy.count(syls_) > 1 then
            syls_ ← naive_syllabify(syls_)
        end if
        extend syls with syls_
    end for
end function
```

Algorithm 2 Splitting and Processing BPE Tokens with Vowel Constraints

```
function SPLIT_ON_NONCONSECUTIVE_VOWELS(token)
    parts ← [ ]
    current ← [ ]
    vowel_positions ← [ ]
    for each character ch in token do
        append ch to current
        if ch ∈ VOWELS then
            record position of ch in current
        end if
        if two or more vowels are non-consecutive then
            cut before the last vowel
            append left substring to parts
            reset current and vowel_positions accordingly
        end if
    end for
    if current is not empty then
        append current to parts
    end if
    return parts
end function
function ENFORCE_VOWEL_CONSTRAINT(parts)
    remove empty parts
    new_parts ← [ ]
    buffer ← empty string
    for each part p in parts do
        buffer ← buffer + p
        if buffer contains a vowel then
            if next part starts with a vowel or "E" then
                continue without breaking
            else
                append buffer to new_parts
                reset buffer
            end if
        end if
    end for
    if buffer not empty then
        if new_parts not empty then
            merge buffer into the last part
        else
            append buffer as a new part
        end if
    end if
    return new_parts
end function
function BPE_PLUS_SYLLABIFY(tokens)
    first_split ← apply split_on_nonconsecutive_vowels to each token
    final_parts ← enforce_vowel_constraint(first_split)
    return final_parts
end function
```