DOI: 10.14746/linpo.2025.67.1.8

Sawtone: A universal framework for phonetic similarity and alignment across languages and scripts

Omar Kamali

Omneity Labs
omar@omneitylabs.com | ORCID 0009-0006-5354-0328

Abstract: Processing text across different scripts presents significant hurdles in natural language processing, especially when dealing with non-standardized orthographies and informal writing systems common in low--resource languages. To address this, we introduce *Sawtone*, an integrated framework designed to enable consistent cross-script phonetic alignment and text normalization. At its heart is an architecture built for in-teroperability, combining a unified phonological feature space rooted in linguistic principles with modular, language-specific adapters. This structure allows for robust mapping and comparison between any pair of scripts. Crucially, it enables diverse adapters—developed using different methods or data—to work together cohesively for cross-language tasks. The framework readily supports alloglottographic text and is designed to function with minimal resource requirements. We demonstrate its practicality through implementations for transliteration, cross-script sequence alignment, and text normalization, further illustrated by a case study on preprocessing Moroccan Arabic data for Large Language Model (LLM) training. Initial results are encouraging: transliteration reached an 88% BLEU score, phonetic-based text sequence alignment achieved 87-95% accu-racy across various language and script pairs, and text normalization significantly reduced variations in spel-ling. *Sawtone* offers a structured, interoperable foundation for advancing phonetic-aware NLP across linguistic boundaries.

Keywords: phonetics, cross-script alignment, low-resource languages, text normalization, transliteration, NLP, phonological alignment, phonological similarity, LLM training, Moroccan Arabic

1. Introduction

While the digital age connects us globally, bridging linguistic and cultural barriers, it also underscores the difficulty in processing text across different contexts and writing systems. This is particularly true for low-resource languages that often lack standardized spellings (Bird 2020). Many such languages exhibit alloglottography, where native

languages are written using borrowed scripts, resulting in diverse written forms (Crystal 2011; Unseth 2005).

These challenges affect several areas. Communication on social media is hampered by non-standard conventions and mixed scripts (Crystal 2011). Low-resource languages struggle with limited digital tools and competing writing systems (Bird 2020). Furthermore, digital preservation efforts face difficulties with cross-script search and retrieval of cultural heritage materials (Naji & Allan 2016). Addressing these issues requires robust phonetic handling and solutions designed for interoperability.

We introduce *Sawtone* (derived from Arabic *ṣawt* 'sound' plus *tone*), an integrated framework created for consistent cross-script phonetic alignment and text normalization, with a strong emphasis on interoperability. Its core architecture (Section 3) brings together:

- A unified phonological feature space (Section 3.2) grounded in linguistic principles, allowing for language-neutral sound comparison.
- A consistent phonetic similarity metric (Section 3.3) operating within this space.
- Modular, language-specific adapters (Section 3.4, Section 4.1) that map diverse orthographies—including non-standard and alloglottographic text—to the universal space.

This design separates the universal representation from language-specific processing. It allows adapters developed independently (using different methods or data) to function together seamlessly and therefore facilitating cross-script and cross-language phonetic processing. *Sawtone* is designed to work with minimal resources, making it suitable for low-resource scenarios. We demonstrate its utility through applications like transliteration (Section 5.2), cross-script alignment (Section 5.1), and text normalization (Section 5.3).

The paper unfolds as follows: Section 2 reviews related work. Section 3 outlines the *Sawtone* framework. Section 4 details the implementation methodologies. Section 5 showcases practical applications. Section 6 presents a case study involving Moroccan Arabic for LLM training. Section 7 discusses strengths, limitations, and future directions.

2. Literature review

Processing text across diverse languages and scripts, particularly non-standard varieties like dialects and informal text, presents significant challenges for Natural Language Processing (NLP). *Sawtone* aims to provide a unified framework grounded in phonetic similarity to tackle these issues. Here, we review related work in phonological representation, grapheme-to-phoneme conversion, text normalization, and the specific challenges posed by cross-script and low-resource contexts.

2.1. Phonological representation and feature systems

Achieving consistent sound representation across languages is crucial. Linguistics utilizes phonological features—abstract properties that differentiate phonemes. The Sound Pattern of English (SPE) (Chomsky 1968) introduced an influential set of binary features.

Subsequent theories have refined feature organization, such as Feature Geometry (George N. Clements 1985), and explored their grounding in phonetics (Ladefoged 1996). While linguistically rich, these systems can be complex to implement computationally. They often require expert knowledge and may struggle to capture gradient phonetic details or non-standard pronunciations.

More recent computational approaches often learn representations directly from data (e.g., phonetic word embeddings (Sharma 2021)), but these may lack explicit phonetic grounding. They can also face difficulties generalizing across different scripts or encountering unseen phonological phenomena. *Sawtone* bridges this gap by proposing a computationally tractable, multi-dimensional feature space (Section 3.2) inspired by established phonetic and phonological principles (Chomsky 1968). This provides an interpretable, quantitative foundation for comparing sounds.

2.2. Grapheme-to-Phoneme conversion and cross-script processing

Grapheme-to-Phoneme (G2P) conversion involves mapping written text to its sound representation. Existing tools like Epitran (Mortensen 2018) and Transphone (Li 2022) offer G2P capabilities but often assume standardized orthographies or lack contextual awareness, facing challenges with informal text or dialectal writing. Traditional approaches frequently lack unified mechanisms for handling arbitrary combinations of scripts (Karimi 2011).

Transliteration—converting text between scripts while preserving pronunciation (Knight 1998)—encounters similar hurdles. *Sawtone* addresses these through its adaptable "adapter" mechanism (Section 3.4, Section 4.1). These adapters map diverse orthographies into *Sawtone*'s universal phonetic space, thereby facilitating G2P, transliteration (Section 5.2), and cross-script alignment (Section 5.1) within a single, unified framework.

2.3. Text normalization for non-standard varieties

The prevalence of non-standard orthographies in user-generated content necessitates text normalization: converting variant forms into a canonical representation (Sproat 2016). Existing techniques include graph-based methods (Sonmez 2014), sequence-to-sequence models (Lourentzou 2019), and nearest neighbor approaches that leverage phonetic or contextual similarity (Ansari 2017; Elgeish 2019).

Normalizing highly variable text remains a difficult task, especially in languages like Arabic with complex morphology or diglossia (Darwish 2014; Habash 2010). Rule-based systems often struggle with the sheer scale of variation, while data-driven models might over-correct or fail when encountering unseen variants. *Sawtone*'s approach (Section 5.3) utilizes phonetic similarity within its universal space (Section 3.2) to cluster orthographic variants. This enables normalization based on phonetic equivalence, proving particularly beneficial for variations stemming from phonetic approximation or alloglottography, as demonstrated in the Moroccan Arabic case study (Section 6).

2.4. Low-resource languages and alloglottography

Many languages lack extensive digital resources and standardized orthographies (Bird 2020). Alloglottography – writing one language using another's script (Unseth 2005), common in digital communication like Arabeezi – often results in inconsistent, phonetically-driven spellings where standard NLP tools tend to perform poorly. *Sawtone* is specifically designed for these scenarios. It requires minimal resources and employs flexible adapters (Section 3.4, Section 4.1) capable of handling non-standard input. The "Crescendo Adapter Refinement" strategy (Section 4.3.4) further assists development in low-resource contexts.

2.5. Positioning *Sawtone*'s contribution

While previous work has addressed aspects of phonetic processing, there remains a need for unified, flexible frameworks that can handle diverse scripts, languages (including low-resource ones), and varying orthographic standards. Existing systems often rely on language-pair-specific rules, struggle with extensive variation, or demand large datasets. Although phonetic similarity measures exist, they are less commonly integrated into comprehensive, adaptable frameworks (Kondrak 2003).

Sawtone's contribution therefore lies in:

- Proposing a universal, quantitative phonological feature space (Section 3.2) that is both linguistically grounded and computationally tractable.
- Introducing a flexible adapter architecture (Section 3.4, Section 4.1) that accommodates various implementation strategies (rule-based, machine learning, hybrid) suitable for different languages and resource levels.
- Demonstrating practical applications in transliteration, cross-script alignment, and robust text normalization (Section 5), particularly for non-standard varieties like Moroccan Arabic (Section 6).

By bridging theory and practice, *Sawtone* aims to provide a more general and robust solution for phonetic-aware text processing across linguistic boundaries

3. Framework

In this section, we present the *Sawtone* Framework (Fig. 1), a flexible, modular framework for phonetic-aware text processing across linguistic boundaries with various downstream applications.

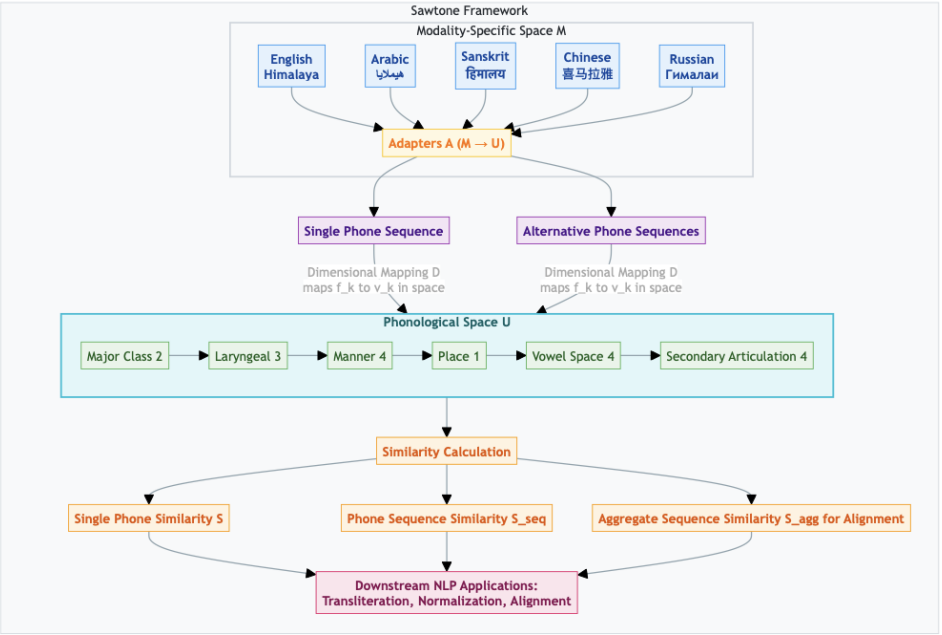

Figure 1: *Sawtone* framework architecture

3.1. Component architecture

The *Sawtone* framework facilitates cross-script phonetic alignment through interacting components centered around a universal phonological representation space.

We define the *Sawtone* framework Φ as follows:

$$\Phi = (U,S,A)$$

where:

- U: Represents the universal phonological space, defined by a feature set and a dimensional mapping .
- S: $U \times U \rightarrow [0,1]$: Is a similarity function that serves as a distance metric between phones within the space .
- A: Denotes a set of modality-specific adapters. These adapters map between the universal space and specific modality spaces M (such as text): $A_{(M \rightarrow U)}: M \rightarrow U$ (encoding) and (decoding).

3.2. Universal phonological representation space (*U*)

At the core of *Sawtone* lies a universal phonological representation space. This is a multi-dimensional vector space designed to quantitatively capture phonetic properties across different languages. Its primary goal is to provide a uniform, comparable representation where spatial proximity directly correlates with perceived phonetic similarity. This structure enables meaningful comparison of sounds originating from different languages or scripts—processed via potentially diverse adapters (Section 4.1)—using standard distance metrics.

The space integrates principles from phonetics and phonology, drawing upon concepts like distinctive features (Chomsky 1968), articulatory phonetics (Ladefoged 1996, 2011), Feature Geometry (George N. Clements 1985, 1995), IPA classifications (IPA 1999), acoustic phonetics (Stevens 1998), and insights from language-specific studies (De Premare 1998; Watson 2002). Features are selected for their orthogonality and are mapped onto a continuous scale suitable for distance calculations. Consequently, each phoneme or sound segment is represented as a vector within this space. We define 18 core feature dimensions based on this synthesis, aiming for comprehensiveness and strong phonetic grounding.

3.2.1. Phonological features (*F*)

The 18 dimensions, categorized below, are numerically represented on a scale from 0.0 to 1.0 (binary or continuous/multi-valued as detailed in Section 3.2.3 Dimensional Mapping (*D*)):

- **Major class (2 dims):** Consonantal, Sonorant (Chomsky 1968).
- **Laryngeal (3 dims):** Voice, Spread Glottis (reflecting aspiration/breathy voice), Constricted Glottis (reflecting glottalization/ejectives) (Gordon 2001; Ladefoged 1996, 2011).
- **Manner (4 dims):** Stricture (ranging from stop to vowel), Nasality, Laterality, TrillTap (Ladefoged 1996, 2011; Stevens 1998).
- **Place (1 dim):** PlaceArticulation (scaled from 0.0 for Bilabial to 1.0 for Glottal) (George N. Clements 1995; Ladefoged 1996, 2011).
- **Vowel space (4 dims):** VowelHeight, VowelBackness, LipRounding, TongueRoot (ATR – Advanced Tongue Root) (Archangeli 1994; International Phonetic Association 1999; Ladefoged 1996, 2011).
- **Secondary articulation (4 dims):** Labialized, Palatalized, Velarized, Pharyngealized (often associated with emphatic sounds) (Ladefoged 1996, 2011; Watson 2002).

Figure 2: *Sawtone* Phonological feature space

This 18-dimensional vector provides a rich, quantitative profile for each sound. Refer to Fig. 2 and Table 1 for an overview.

Table 1: Overview of *Sawtone* phonological feature dimensions

Feature Category	Dimensions	Specific Features	Primary References
Major Class	2	Consonantal, Sonorant	(*Chomsky, 1968*)
Laryngeal	3	Voice, Spread Glottis, Constricted Glottis	(*Gordon, 2001*; *Ladefoged, 1996*)
Manner	4	Stricture, Nasality, Laterality, TrillTap	(*Ladefoged, 1996*, *2011*; *Stevens, 1998*)
Place	1	PlaceArticulation (Scaled 0-1)	(*George N. Clements, 1995*; *Ladefoged, 1996*, *2011*)
Vowel Space	4	VowelHeight, VowelBackness, LipRounding, ATR	(*Archangeli, 1994*; *International Phonetic Association, 1999*; *Ladefoged, 1996*, *2011*)
Secondary Articulation	4	Labialized, Palatalized, Velarized, Pharyngealized	(*Ladefoged, 1996*, *2011*; *Watson, 2002*)
Total	**18**		

3.2.1.1. Suprasegmental features

Features like stress, pitch, and tone (Lehiste 1970), known as suprasegmentals, are generally absent in written text. Therefore, they are excluded from the current feature space. However, future extensions supporting speech modalities could incorporate them.

3.2.1.2. IPA lookup table (LUT)

The IPA lookup table serves as a critical bridge in the *Sawtone* framework, mapping International Phonetic Alphabet (IPA) symbols to their corresponding 18-dimensional feature vectors in the universal phonological space. This mapping is deterministic and language-agnostic, functioning as the foundation that enables cross-linguistic interoperability within our framework. During our work, the LUTs were constructed following the IPA (1999) and Ladefoged's (1996) guidelines.

Each IPA symbol (e.g., [p], [a], [ʃ]) is assigned a precise vector representation based on its phonetic properties across all 18 dimensions. For example, the voiceless bilabial plosive [p] would have high values for Consonantal and Stricture features, a value of 0 for PlaceArticulation (bilabial), and 0 for Voice. This standardized representation ensures that phonetically similar sounds across different languages occupy proximate positions in the universal space.

The lookup table is particularly valuable for:

- **Interpretability**: The LUT as an intermediate step allows for more interpretable adapters
- **Cross-script compatibility**: Enabling meaningful comparisons between sounds from distant writing systems (e.g., Arabic, Latin, Cyrillic)

- **Adapter interoperability**: Allowing adapter outputs to be comparable through a shared representation, regardless of their implementation details
- **Phonological analysis**: Providing a quantitative basis for studying sound patterns across languages

This standardized mapping layer ensures that all phonetic information, regardless of source language or script, can be represented in a consistent format within the universal phonological space, forming the backbone of *Sawtone*'s language-agnostic approach to phonological representation.

3.2.1.3. Phonetic inventory

Each language has its own phonetic inventory—the set of distinct sounds (phonemes) that can be expressed in that language. These inventories are typically defined using IPA symbols and can be sourced from multilingual dictionaries (De Premare 1998) or introductory linguistic materials for specific languages.

While collecting a language's complete inventory is valuable, it isn't always mandatory within our framework. When parallel text-to-IPA data is available, the inventory can be reconstructed automatically by extracting the IPA symbols that appear in the data. Regardless of the collection method, the inventory should contain only valid IPA phonemes to ensure accurate representation in the universal phonological space.

3.2.2. Universal phonological space (*U*)

The universal phonological space encompasses all possible 18-dimensional feature vectors where each dimension ranges from 0 to 1 ($[0,1]^{18}$). This space forms the domain for the similarity function:

$$U = \{\phi \in [0,1]^{18} \mid \phi = D\ (f)\ f \text{ or } f \in F\}$$

Representing sounds as vectors in this shared space allows for the calculation of phonetic distance (e.g., using weighted cosine distance, Section 3.3). The underlying hypothesis is that sounds perceived as similar will be located closer together within this space. Adapters (Section 4.1) map diverse orthographies into comparable locations within , ensuring cross-linguistic applicability. Notably, IPA features can be deterministically mapped to and from the *Sawtone* space, reinforcing its universality. The space itself is language-agnostic, accommodating language-specific variations through adapters and optional similarity weights. Special partial or superposition phones handle incomplete information (Section 3.4.1).

3.2.3. Dimensional mapping (*D*)

This process translates the 18 abstract features (f_k) into numerical values v^k within the range [0, 1], forming the vector $\phi_p = (v^1,...,v18)$ for a given phone p.

- **Binary features** (e.g., Consonantal): Mapped directly to either 0.0 or 0.1.
- **Categorical features** (e.g., VowelHeight, Stricture): For features with ordered categories (indexed $i = 0$ to $N = 1$), these are mapped to uniformly spaced points: $v^k = i/(N–1)$. For instance, 4 categories would map to .
- **Gradient features** (e.g., PlaceArticulation): Mapped linearly onto the [0, 1] range based on an established phonetic scale (e.g., Bilabial maps to 0.0, Glottal to 1.0).
- **Standard implementation:** Unless otherwise specified, these standard mappings are used. Range bounding is applied to adapter outputs to ensure numerical stability.
- **Limitations & future work:** Assuming uniform spacing for categorical features is a simplification. Future research could explore non-uniform mappings derived from acoustic or perceptual data (Kondrak 2003), or through data-driven learning, balancing enhanced accuracy with interpretability. This paper concentrates on the framework architecture using the standard mapping.

3.3. Similarity metric (*S*)

To compare individual phones and , we use their vector representations v_{p1} and v_{p2} with a **weighted cosine similarity** measure (Kondrak 2003) illustrated in Fig. 3:

$$S(p_1, p_2) = \frac{\sum_{k=1}^{18} w_k v_{p_1}^k v_{p_2}^k}{\sqrt{\sum_{k=1}^{18} w_k (v_{p_1}^k)^2} \sqrt{\sum_{k=1}^{18} w_k (v_{p_2}^k)^2}}$$

Here:
- v_p^k is the k-th feature value for phone *p*.
- w_k represents the weight for the k-th feature ($0 \leq w_k \leq 1$), reflecting its relative importance.

This calculation yields a score between 0 and 1, indicating the phonetic closeness of the two phones.

Feature weights (w_k): These weights allow for nuanced control over the similarity calculation. Using uniform weights (w_k =1 for all k) provides a strong baseline. Optimizing these weights, perhaps based on acoustic or perceptual data, remains an area for potential future work. For this paper, we assume uniform weights unless stated otherwise.

3.3.1. Phonetic sequence similarity (S_{seq})

To compare sequences of phones, $A = (p_{a_1},...,p_{a_m})$ and $B = (p_{b_1},...,p_{b_n})$, we employ **sequence alignment**, specifically the Needleman-Wunsch algorithm (Needleman, n.d.) illustrated in Fig. 3:
- **Match/Mismatch score:** Determined by the pairwise *Sawtone* similarity $S(p_{a_i}, p_{b_j})$ (eq. 3) between aligned phones.

- **Gap penalty (g):** A constant penalty (e.g., $g = -0.5$) applied when aligning a phone with a gap. This is a tunable hyperparameter.

This process yields a raw alignment score $S_{raw\_seq}(A, B)$. For comparison across sequences of different lengths, this score can be normalized, for instance, by the average sequence length:

$$S_{seq}(A, B) = \frac{S_{raw\_seq}(A, B)}{(m + n)/2}$$

This provides a robust measure of the overall phonetic resemblance between two sequences.

3.4. Modality-specific adapters

Adapters serve as the bridge between the universal phonological space and modality-specific spaces , such as text or potentially speech. They perform two key functions: encoding representations from a specific modality into , and decoding representations from back into a modality (Mortensen, 2018). See Methodology (Section 4.1) for implementation details.

$$A_{U \to M} : U \to M(Decode)$$

$$A_{M \to U} : M \to U(\text{Encode})$$

3.4.1. Handling phonetic ambiguity

Writing systems often provide incomplete phonological information, meaning the mapping performed by adapters may not be perfectly reversible (bijective) and can sometimes be lossy (Bird, 1994). To manage this, we introduce two special elements within :

1. Partial phone (ϕ_p): Represents a placeholder where certain features are unspecified (nullified) due to missing information. Formally, a partial phone contains undefined values for one or more feature dimensions in the vector. We use an underscore character to represent Partial Phones in phone sequences in this paper.

2. Superposition phone (ϕ_s): Represents multiple possible phonological interpretations simultaneously, useful for cases like heteronyms or inconsistent spelling conventions. Formally, a superposition phone is a weighted distribution over a finite set of possible phones $\{\phi_1, \phi_2, ..., \phi_n\}$ with associated weights $\{p_1, p_2, ..., p_n\}$ where $\sum_i p_i = 1$ for adapters that support it.

These elements allow the framework to represent ambiguity explicitly, carrying it through calculations, albeit potentially at the cost of some precision.

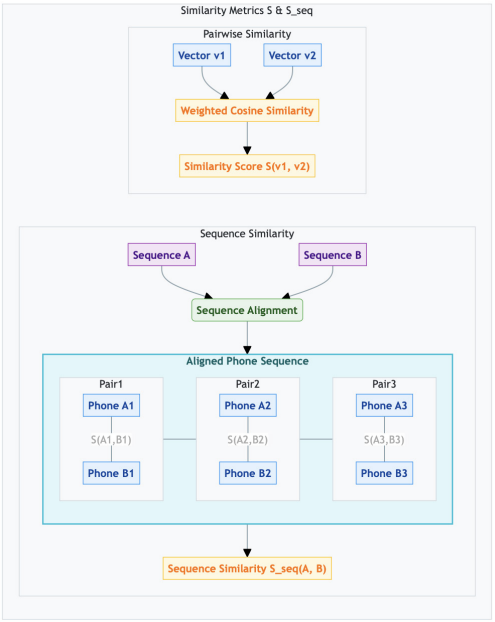

Figure 3: Similarity metrics S & S_{seq}

3.4.2. Calculating similarity with ambiguous phones (S_{agg})

Due to ambiguity, an adapter might generate multiple potential phonological sequences for a given input (e.g., a set $A = \{A_1, ...\}$ for input X_A, and $B = \{B_1, ...\}$ for input X_B). In such cases, we need a method to calculate an aggregated similarity score between these *sets* of sequences.

We first compute the pairwise sequence similarities S_{seq} (A_i, B_j) (Section 3.3.1) for all combinations and then aggregate these scores:

1. Average similarity: $S_{avg}(A,B) = \frac{1}{|A||B|}\sum_{i,j} \quad S_{seq}(A_i, B_j)$

2. Maximum similarity: $S_{max}(A,B) = max_{i,j} S_{seq}(A_i, B_j)$

3. Weighted average similarity: If adapters provide confidence scores (A_i) and (B_i) for each potential sequence:

$$S_{weighted\_avg}(A,B) = \frac{\sum_{i,j} \quad w(A_i)w(B_j)S_{seq}(A_i,B_j)}{\sum_{i,j} \quad w(A_i)w(B_j)}$$

Computational approximation: When the number of potential sequences $|A|$ and $|B|$ is large, computing all pairwise similarities can be expensive. We can approximate A_{agg} by:

1. Selecting representative subsets $R_A \subseteq A$ and $R_B \subseteq B$ (of size and , respectively) through sampling or clustering (e.g., using k-medoids).

2. Computing S_{seq} only for pairs within the Cartesian product $R_A \times R_B$.

3. Aggregating these $N_A \times N_B$ scores using one of the methods above.

The sizes N_A and N_B control the trade-off between computational cost and approximation accuracy.

3.5. Extension points

The framework's architecture is designed to be extensible. It allows for the integration of new components, such as different distance metrics or adapters for modalities like speech (Goldsmith 1990). Furthermore, individual components can potentially be used independently; for instance, the universal space could be employed solely for phonetic similarity calculations in Information Retrieval (IR).

Additional features (like suprasegmental or prosodic features) can be incorporated into the *Sawtone* space, and new adapters can be developed to handle them without altering the core framework structure.

4. Methodology

This section delves into the practical aspects of *Sawtone*, detailing the construction of its universal phonological space (Section 3.2) and outlining flexible strategies for implementing the adapters (Section 3.3). These adapters are crucial for mapping text to and from this universal space, accommodating diverse languages and varying resource levels. A primary consideration during adapter development is feasibility, especially given the resource constraints often associated with low-resource languages.

4.1. Modality-specific adapters ($A_{M\to\Phi}$ and $A_{\Phi\to M}$)

The effectiveness of *Sawtone* hinges on reliably mapping between diverse orthographies (graphemes) and the universal phonological space (encoding via $A_{M\to\Phi}$, (Fig. 4)), and potentially mapping back to a target script (decoding via $A_{\Phi\to M}$, (Fig. 5)). Writing systems exhibit significant variation—from alphabetic to syllabic, with spelling regularities ranging from highly consistent to quite irregular. Likewise, the availability of resources like dictionaries, corpora, and linguistic expertise varies widely. Consequently, no single adapter implementation strategy proves universally optimal.

Sawtone's flexibility allows for the integration of various techniques. We explored several complementary approaches:

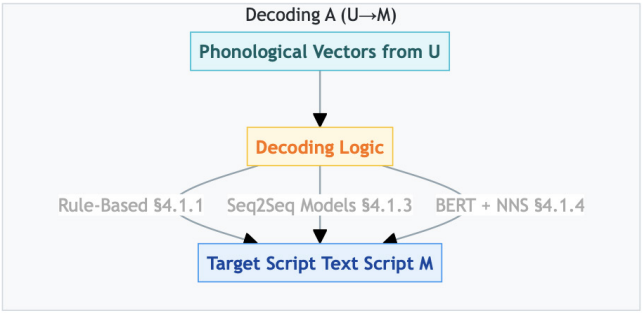

Figure 4: Encoding adapter architecture

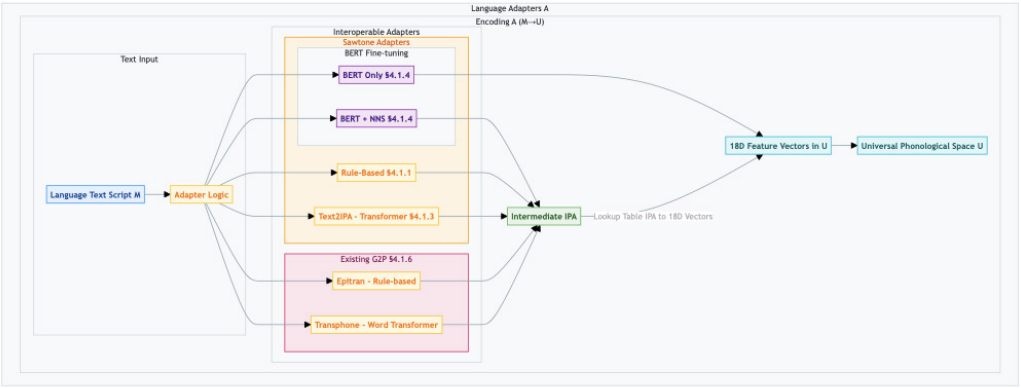

Figure 5: Decoding adapter architecture

4.1.1. Rule-based systems

- *Principle*: These systems encode expert linguistic knowledge into sets of context-sensitive rewrite rules (Fig. 6). These rules map sequences of graphemes to an intermediate phonetic representation (e.g., IPA), which is then converted to 18D feature vectors using a lookup table. We take inspiration from FST (Finite State Transducer) algorithms (Koskenniemi 1983) in the implementation of our rule-based adapters (Fig. 7).
- *Implementation*: Typically involves multi-layered, ordered rules formatted like *[ContextBefore, MatchSeq, ContextAfter, Replacement]*, performing text-to-text or text-to-IPA transformations. Successive layers handle progressively more complex phenomena, starting with basic mappings and moving to context-dependent rules and phonological processes. These systems can generate multiple interpretations to manage ambiguity (Section 3.4.1).

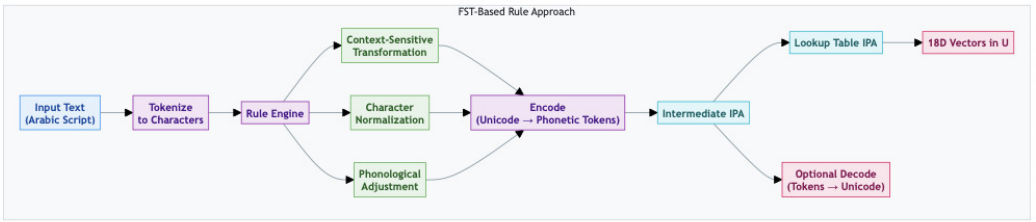

Figure 6: Rule-based adapter architecture

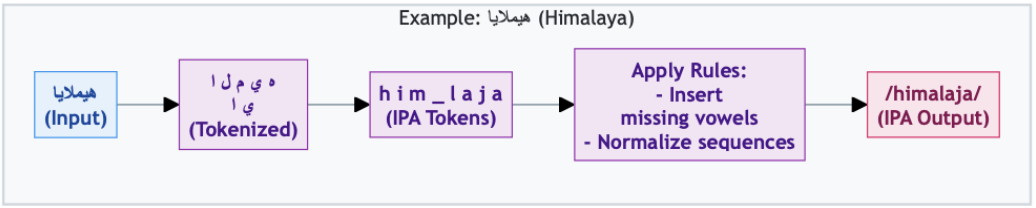

Figure 7: Rule-based adapter example (هيملايا → /himalaja/)

- *Decoding*: While rule-based decoding (IPA back to graphemes) is feasible, it can be a lengthy and costly process, especially for non-standard orthographies. Data-driven methods are often preferred for decoding.
- *Applicability*: Best suited for languages with regular orthographies or where linguistic expertise is readily available. Ensures outputs align well with the IPA-based features of the *Sawtone* space. However, development is labor-intensive, and the rules can be brittle when faced with irregular or noisy text.

4.1.2. A note on tokenization

Statistical methods generally require explicit text tokenization. For scalability, we focus on statistical tokenizers. Tokenization itself is complex due to the variable mapping between graphemes and phonemes (e.g., 'sh' mapping to a single phoneme /ʃ/ versus 's' and 'h' representing separate sounds in *misheard*) (Ni 2018). This presents a trade-off:

1. Single-character tokens: Risk splitting multi-character representations of single phonemes (like 'sh' into 's' and 'h').

2. Multi-character tokens: Risk merging adjacent characters that represent distinct phonemes (like 's' and 'h' in *misheard*).

Achieving perfect resolution typically requires language-specific contextual tokenizers, which is often intractable for language-agnostic systems. We approximate phoneme boundaries using multi-character tokenization, preferring potential ambiguity (merging) over definite information loss (splitting), relying on the adapters' contextual capabilities to interpret the resulting tokens correctly.

Since phonemes rarely span more than 2-3 characters, while standard tokenizers like BPE or WordPiece might create overly long tokens, we trained custom Unigram tokenizers for each language. We limited the maximum token length (typically 3 characters; 1 for ideographic scripts) and pruned rare tokens. This approach balances vocabulary size (around to tokens) with the ability to capture common multi-character phonemes. While cross-lingual tokenizers are possible, per-language tokenizers reduce the demands placed on the adapter models. Our Unigram approach offers a simple solution that mitigates the risk of splitting phonemes.

4.1.3. Sequence-to-Sequence (Text2IPA) models

- *Principle*: These models (Fig. 8) leverage standard transformer architectures (like T5 (Raffel, 2020), GPT-2 (Radford, 2019), Llama 3 (L. 3. Team, 2024), Qwen 2.5 (Q. Team, 2024)) trained on parallel corpora mapping text to IPA sequences. Large Language Models (LLMs) generally demonstrated better handling of out-of-domain input and performed better on longer sequences.
- *Implementation*: Typically involves training one model per language or language pair. The model predicts an IPA sequence from the input text; these IPA symbols are then mapped to 18D *Sawtone* vectors via a LUT (Section 3.2.1.2). Training data often comes from parallel corpora (e.g., Wikipedia, social media) automatically converted to IPA using phonetic dictionaries (Doherty, n.d.). Our largest dataset comprised 10 million sentences.

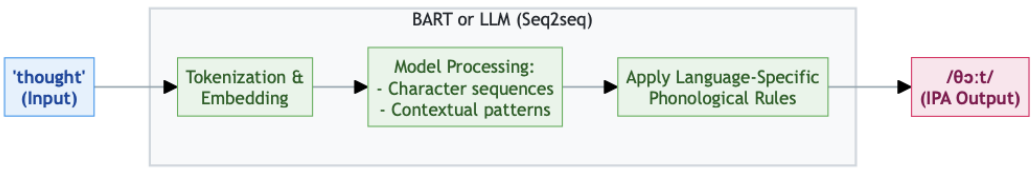

Figure 8: Text2IPA Adapter Example (‘thought’ → /θɔ:t/)

- *Decoding*: A reverse model (IPA -> Text) can be trained using the same data and methodology. Performance varies depending on the language pair and dataset quality.
- *Applicability*: Highly effective when sufficient aligned text-IPA data is available. These models can learn complex mappings and handle irregularities present in the training data. Outputs naturally align with the IPA-based *Sawtone* features via the lookup table mapping. Requires less manual effort than rule-based systems but is data-dependent and less interpretable.

4.1.4. BERT embedding regression and fine-tuning

- *Principle*: This approach refines existing text embeddings by training on pairs of phonetically related strings, such as homophones, transliterations, or rhymes. It aims to learn phonetic exceptions and nuances from large volumes of such paired data.

Languages with strict poetic conventions (like Arabic *buḥūr*, French metres, Sanskrit *chandas*) can particularly benefit. This method is empirically driven and capable of capturing subtle phonetic variations.

- *Decoding*: Decoding often involves a nearest neighbor search on the resulting vectors using the *Sawtone* phonetic similarity metric, selecting the closest phoneme from the target language's phonetic inventory.
- *Implementation*: We utilized a compact BERT-like model (based on MicroBert (Gessler & Zeldes 2022), with 64/128 dimensions), assuming phonetic information is less complex than full semantics. The model was first pretrained using Masked Language Modeling (MLM) on target language data. It was then fine-tuned using one of two strategies:
  1. *Embedding regression*: Training the model to project its BERT embeddings into the 18D *Sawtone* space using Mean Squared Error (MSE) loss against target vectors generated by another adapter (e.g., a rule-based or Text2IPA model acting as a "teacher"). The quality depends on the teacher model, but this allows for manual correction of teacher errors.
  2. *Semantic Textual Similarity (STS) adaptation*: Fine-tuning the model on pairs known to be phonetically close, such as homophones, transliterations, or lines from poetry/lyrics (e.g., Moroccan *zajal*/*melhoun*, Chinese *ci*, English rap). Using pairs that sound similar (not just identical) provides a potent signal for learning subtle phonetic relationships.
- *Applicability*: Useful for enhancing the phonetic awareness of embeddings, especially when large annotated datasets are scarce but lists of phonetically related pairs are available.

4.2. Comparative overview of adapter strategies

All adapter types ultimately map text to vectors in the universal phonological space, either directly or via an intermediate IPA representation. Rule-based and Text2IPA adapters typically perform this mapping via IPA. Text2IPA models are generally preferred for their ability to handle context effectively. BERT fine-tuning focuses on refining embeddings using phonetic similarity signals derived from paired data, which can be valuable for bootstrapping high-quality Text2IPA adapters. Table 2 summarizes key properties of these adapter strategies:

Table 2: Alignment with IPA-based features, resource requirements and performance

Method	Aligned with IPA Features?	IPA-Interpretability	Expert Rules?	Text-IPA Data?	Pair Lists?	Hours to Train	Converted Tokens/s
Rule-Based	Yes	Yes	Yes	No	No	Low	0.5M
Text2IPA (Seq2seq)	Yes	Yes	No	Yes	No	High	200
BERT Fine-tuning	Yes	Partial	No	Yes (as vectors)	Yes	High	15K

Computational resources: Figures are based on a single NVIDIA A100 GPU (SXM 80GB). Rule-based execution measured on a single core of an AMD EPYC 7551 CPU. **Performance**: Values provide a rough guide and depend heavily on data quality and specific implementation details.

4.2.1. Using existing G2P tools

Any method capable of producing IPA sequences from graphemes can be integrated with *Sawtone*. We explored Epitran (Mortensen, 2018) (which uses rules and lookup tables) and Transphone (Li 2022) (a word-level transformer trained on Wiktionary). Epitran behaves similarly to our rule-based adapters. Transphone resembles Text2IPA adapters but operates with limited context and wasn't trained on noisy text. Both are compatible with the *Sawtone* framework. A comprehensive evaluation across many languages would require suitable cross-lingual, phonetically-annotated datasets that reflect real-world variation, which is a direction for future work.

4.3. Practical considerations

4.3.1. Universality of the IPA features

Ensuring that the outputs from different adapters align consistently within the universal, IPA-based space requires two conditions:

1. The vector dimensions must be interpretable as (combinations of) IPA features.
2. The value ranges used for these features must be comparable across different adapters and languages.

This consistency is achieved by using a common LUT (Section 3.2.1.2) that maps IPA symbols to predefined *Sawtone* feature vectors. This LUT is used by methods involving an intermediate IPA representation (rule-based, Text2IPA), ensuring consistency. BERT vectors capture phonetic similarity implicitly but are not directly aligned with specific IPA features and cannot be interpreted as such.

4.3.2. Handling orthographic noise

Real-world text often contains typos and non-standard spellings. We employed an unsupervised filtering technique: clustering words with high orthographic similarity (low edit distance) that appear in similar contexts, and then filtering out low-frequency variants within these clusters as likely noise. This approach aims to balance normalization benefits with the preservation of legitimate rare forms.

4.3.3. What about diacritics?

Languages like Arabic utilize optional diacritics for determining correct pronunciation. Their absence leads to a loss of phonetic information. We evaluated several strategies to handle this:

- *Deterministic recovery*: Requires sophisticated POS tagging and morphological analysis, often infeasible or inaccurate for noisy, informal text.
- *Statistical inference*: Requires large, diacritized corpora matching the target text register; we attempted this for Arabic, finding it promising but unstable.
- *Treat absence as ambiguity (partial/superposition phone)* (Section 3.4.1): This was our choice due to its simplicity and effectiveness at handling various scenarios. It trades accuracy for robustness.

4.3.4. Crescendo adapter development strategy

The choice of adapter strategy depends on the target language and available resources. Rule-based systems (Section 4.1.1) are a good starting point if linguistic descriptions exist. Text2IPA models (Section 4.1.3) are viable if phonetic dictionaries or aligned corpora are available. BERT fine-tuning (Section 4.1.4) becomes an option if homophones, transliterations, or relevant poetic data can be compiled.

These methods can be employed iteratively in what we term the "Crescendo" strategy. For instance, simple rules can generate initial phonetic data to bootstrap BERT training; the improved embeddings from BERT can then be used to train more sophisticated Text2IPA models. This strategy allows for robust adapter development even for low-resource languages by progressively leveraging different types of data and techniques, starting with human expert knowledge. This contrasts with approaches requiring large, perfectly annotated datasets from the outset. Our development process involved iterative refinement and qualitative validation.

5. Applications of *Sawtone* to NLP tasks

This section illustrates how the *Sawtone* framework can be applied to core NLP tasks that involve cross-script interactions and non-standard text, namely: sequence alignment, transliteration and normalization.

5.1. Cross-script sequence alignment

Task Definition: This task involves aligning text sequences written in different scripts based on their underlying phonetic similarity, identifying corresponding sound segments (Figs. 9 & 10). It is fundamental for determining how phonetically related two strings are across script boundaries. The objectives are twofold: 1) Align sequences of phones, potentially introducing gaps where necessary. 2) Calculate an overall phonetic similarity score based on the alignment.

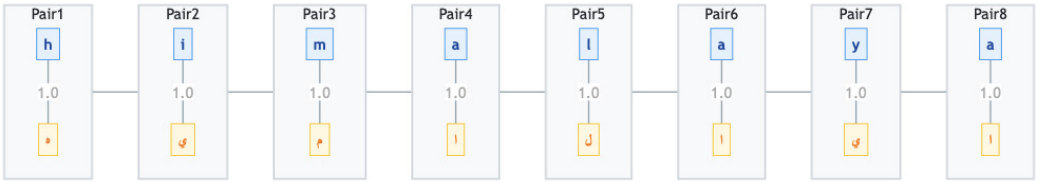

Figure 9: *Perfect Alignment, Score 1.0:* Comparing *Himalaya* with its Arabic transliteration هيملايا (himalaya).

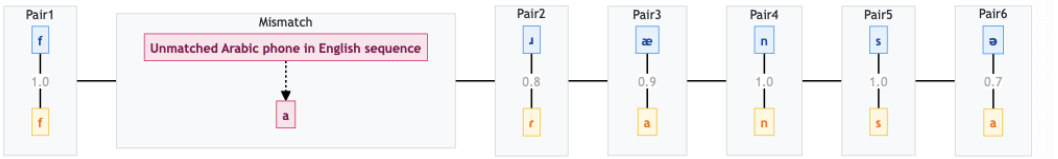

Figure 10: *Imperfect Alignment, Score < 1.0:* Comparing *France* with its common Arabic transliteration فرنسا (faransa). Note the vowel mismatch and the different 'r' sounds.

***Sawtone* Application**:

- *Encoding*: Encode the input sequences (Sequence X in Script A, Sequence Y in Script B) into *Sawtone* sequences using their respective adapters.
- *Sequence-Aligned Scoring*: Apply the Needleman-Wunsch algorithm (Section 3.3.1), using the *Sawtone* similarity metric (eq.3) to calculate match/mismatch scores between pairs of phone vectors with gap penalty .
- *Similarity Score*: Calculate the average similarity of the aligned vector pairs (excluding pairs involving gaps).

5.1.1. Experimental setup

Datasets: We used manually curated word pairs covering various languages and challenges:

- ParaNames (Sälevä, 2022) (Arabic/English names)
- Google Transliteration Dataset (*Rosca, 2016*) (Arabic/English general words)
- ANETAC (Ameur, 2019) (Moroccan Arabic/Arabeezi named entities)
- Internal French Homophone List
- Internal Japanese/English Homophone List
- Sentence pairs were generated where possible. These datasets typically averaged around 500 pairs per language pair.

5.1.2. Note on choosing adapters

For most languages, we utilized Text2IPA adapters (Section 4.1.3) as parallel script-to-IPA corpora were available for training. However, for Moroccan Arabic (MA), characterized by high variation and lack of initial parallel IPA data, we employed the Crescendo strategy (Section 4.3.4). This involved starting with a rule-based adapter (Section

4.1.1) and progressively refining it towards a Text2IPA model. Text2IPA adapters generally offer good flexibility and resistance to noise.

5.1.3. Moroccan Arabic adapters: Application of the Crescendo strategy

Developing effective adapters for MA was challenging due to its diglossic nature, non-standard spellings, and noisy data sources (Kamali & Abchir 2024). The goal was to map text written in both standard Arabic script and varied Latin-based Arabeezi scripts to the *same* underlying IPA sequence for consistent alignment. Since parallel script-IPA corpora were unavailable, the Crescendo strategy (Section 4.3.4) was essential. The multi-step development process (Rule-based -> BERT -> Text2IPA) is an iterative process that allowed us to generate progressively better training data (MA/Arabeezi -> IPA pairs), ultimately enabling the training of a robust Text2IPA adapter capable of handling the complexities of MA.

5.1.4. Adapters for other languages (English, Arabic, French, Japanese)

For other languages like English, Standard Arabic, French, and Japanese, we used Text2IPA adapters (Section 4.1.3) trained directly on existing parallel script-IPA corpora (averaging around 10 million words per language). These adapters proved effective, although their quality is inherently limited by the training data (e.g., word-level IPA conversion might miss phonetic changes occurring between words). Despite these minor limitations, they achieved good results. Further fine-tuning with homophone lists could offer marginal gains but was not deemed necessary for these initial experiments.

Evaluation: We assessed the system's ability to assign high similarity scores to phonetically related pairs (like transliterations and homophones) and low scores to unrelated pairs. Alignment accuracy was measured by calculating the edit distance between the predicted alignment and a ground truth alignment (counting gaps, normalized by the length of the longer sequence, with vowel/silent letter gaps potentially down-weighted).

5.1.5. Metrics and results

Alignment accuracy: Measured as 1 minus the normalized edit distance between the predicted and ground truth alignments. **F1 score**: The harmonic mean of precision and recall, evaluating the correctness of matched phone pairs in the alignment.

Table 3: Alignment system evaluation results across language pairs

Language pair	Alignment accuracy	F1 score
Arabic/English	0.92	0.94
Moroccan Arabic/Moroccan Arabeezi	0.90	0.92
Japanese/English	0.87	0.88
French/French (Homophones)	0.95	0.97

Discussion: The high accuracy and F1 scores reported in (Table 3) across diverse language pairs demonstrate *Sawtone*'s effectiveness in identifying phonetic relatedness, both across different scripts and within the same script (for homophones), by interpreting the underlying sounds represented by the graphemes. The examples below (Tables 4-8) illustrate the nuanced similarity scores produced by the system.

5.1.5.1. Arabic/English

Table 4: Arabic/English homophone pairs with Sawtone similarity scores

Native	Romanized	English	Sawtone similarity
سيل	sail	Sail	1.00
كمال	kamal	Camel	0.96
قرين	qarin	Karen	0.74
فم	fam	Fam	0.98
بر	bar	Bar	0.92
قلم	qalam	Column	0.63
كلب	kalb	Calf	0.75
دارت	darat	Dart	0.86
زهر	zahr	Tsar	0.78
لوم	lawm	Loom	0.83
أمنيتي	umniyati	Omneity	0.89

5.1.5.2. Moroccan Arabic/Arabeezi

Similarity scores between MA words written in Arabic script and their Arabeezi counterparts, including various spelling variants.

Table 5: Phonetic similarity between Arabeezi spellings of *tbarkellah* (تبارك الله) 'God bless'

Word	tbarkellah	tabarklah	tbraklah	tbrklh	tabaraka allah	tbar9ela7
tbarkellah	1.00	0.95	0.95	0.92	0.92	0.76
tabarklah	0.95	1.00	0.92	0.87	0.92	0.71
tbraklah	0.95	0.92	1.00	0.96	0.86	0.77
tbrklh	0.92	0.87	0.96	1.00	0.79	0.80
tabaraka allah	0.92	0.92	0.86	0.79	1.00	0.67
tbar9ela7	0.76	0.71	0.77	0.80	0.67	1.00

Table 6: Phonetic similarity between spellings of *ghanmchi* (غانمشي) 'I will go' in MA and Arabeezi

Word	hanmchi	ghanmchi	ghnmchi	ghannamchi	ghanemechi	غانمشي	غامشي	غنمشي
hanmchi	1.00	0.94	0.60	0.83	0.81	0.94	0.46	0.60
ghanmchi	0.94	1.00	0.61	0.87	0.87	1.00	0.53	0.61
ghnmchi	0.60	0.61	1.00	0.48	0.58	0.61	0.67	1.00
ghannamchi	0.83	0.87	0.48	1.00	0.90	0.87	0.45	0.48
ghanemechi	0.81	0.87	0.58	0.90	1.00	0.87	0.50	0.58
غانمشي	0.94	1.00	0.61	0.87	0.87	1.00	0.53	0.61
غامشي	0.46	0.53	0.67	0.45	0.50	0.53	1.00	0.67
غنمشي	0.60	0.61	1.00	0.48	0.58	0.61	0.67	1.00

5.1.5.3. Japanese/English

Table 7: Japanese/English homophone pairs with *Sawtone* similarity scores

Native	Romanized	English	Sawtone similarity
ユーセイ	yusei	You say	0.85
飴	ame	Amy	0.90
箸	hashi	Hushy	0.88
蜜	mitsu	Meets	0.87
待つ	matsu	Match	0.83
買う	kau	Cow	0.89
家	ie	Yeah	0.86
足	ashi	Ashy	0.91
波	nami	Nummy	0.82
夢	yume	You May	0.83
鳥	tori	Tory	0.80
花	hana	Honor	0.81
山	yama	Yammer	0.79
音	oto	Auto	0.92
竹	take	Tacky	0.75
餅	mochi	Mushy	0.76
見る	miru	Mirror	0.78
来る	kuru	Crew	0.77

5.1.5.4. French Homophones

Table 8: French homophone pairs with *Sawtone* similarity scores

Word 1	Word 2	Sawtone similarity
mer	mère	0.99
maire	mère	1.00
sain	saint	0.97
sein	ceint	0.93
vers	verre	0.92
vert	ver	0.94
saut	seau	0.94
sot	seau	0.91
sang	sans	0.97
cent	sans	0.88
fois	foie	0.93
foi	foie	0.98
pair	père	0.95
paire	père	0.98
pot	peau	0.97
champ	chant	0.89
compte	conte	0.91
comte	conte	0.96

Task Definition: Converting text from one writing script to another (Fig. 11) while aiming to preserve the original pronunciation as closely as possible (Knight 1998). It's crucial for tasks like Named Entity Recognition (NER), Cross-Lingual Information Retrieval (CLIR), and handling alloglottography.

5.2.1. *Sawtone* application

Transliteration using *Sawtone* involves two stages:

- **Encoding ($A_{(M\rightarrow\Phi)}$)**: The source text (in Script A) is first processed by its corresponding *Sawtone* adapter (Section 3.3) to generate a sequence of phonological vectors (Section 3.2).
- **Decoding ($A_{(\Phi\rightarrow M)}$)**: These phonological vectors are then fed into the adapter for the target script (Script B), which generates the corresponding grapheme sequence. The quality of the final transliteration depends on the accuracy of both the encoding and decoding adapters.

5.2.2. Experimental Setup

Dataset: We used the ParaNames dataset (Sälevä 2022), filtering it to obtain 1000 direct Arabic-to-English transliteration pairs, primarily focusing on named entities. **Adapter**:

Language-specific Text2IPA adapters were employed for both encoding (Arabic -> IPA/*Sawtone* vectors) and decoding (IPA/*Sawtone* vectors -> English), trained as described in the Methodology section.

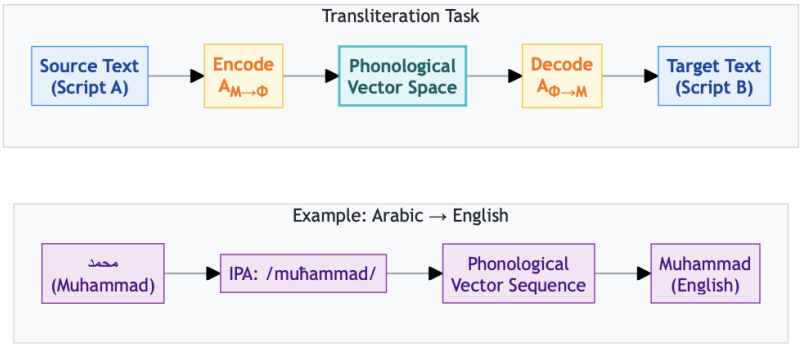

Figure 11: Transliteration with *Sawtone*

We allowed the decoding adapter to sample multiple potential outputs, ranked by perplexity. **Evaluation**: We generated English transliterations for the Arabic names in the dataset and compared them against the provided reference transliterations.

5.2.3. Metrics and results

Performance was assessed using several standard metrics (summarized in Table 9):

- **WER (Word error rate)**: Calculated using Levenshtein distance at the word level (treating each name as a single word). Lower scores indicate better performance.
- **CER (Character error rate)**: Levenshtein distance calculated at the character level. Lower is better.
- **BLEU score (Mean)**: Measures the overlap of character n-grams between the generated transliteration and the reference (treating the name as a sentence). Higher scores are better.
- **MRR (Mean reciprocal rank)**: Evaluates the quality of the ranked list of potential transliterations generated by the decoder. It measures how high up the correct reference appears in the ranked list, on average. Higher is better.

Table 9: Results for Arabic to English transliteration on ParaNames

Metric	Score
Word Error Rate	0.24
Character Error Rate	0.15
Mean BLEU Score	0.88
Mean MRR	0.67

Discussion: The results in Table 9 indicate reasonable performance. The high character-level similarity (reflected in BLEU and CER) suggests the phonetic mapping is largely successful, although there is room for improvement in generating the exact target word form consistently (reflected in WER and MRR).

5.3. Text normalization

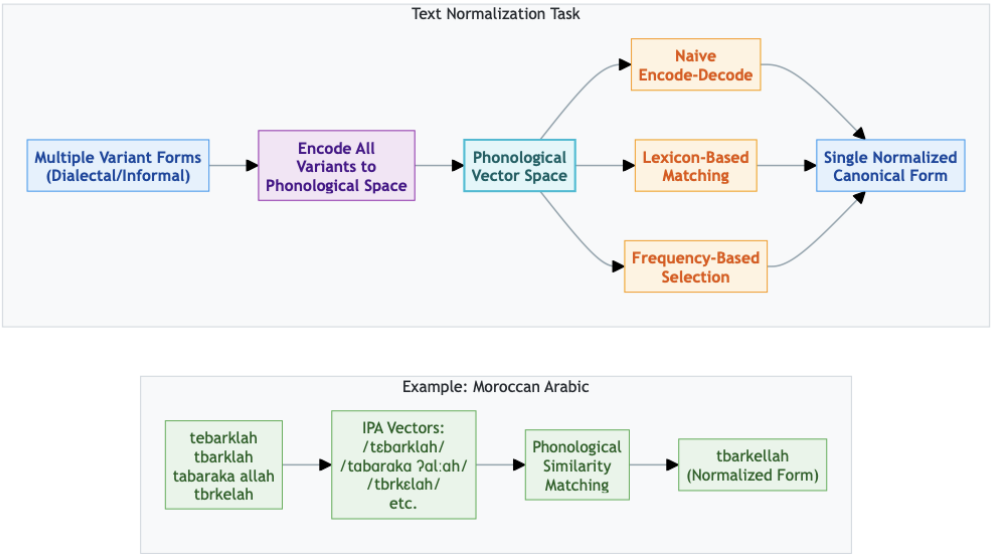

Figure 12: Text normalization and an example in Moroccan Arabic

Task definition: Converting non-standard text forms—arising from typos, dialectal spellings, informal abbreviations, etc.—into a standardized or canonical representation (Sproat 2016) as illustrated in Fig. 12.

***Sawtone* application**: *Sawtone* enables phonetic-based normalization. We encode words into the *Sawtone* space and then map phonetically similar variants to a single, chosen canonical form. We explored three approaches:

- *Naive encoding-decoding*: Simply encode the input word using its script's adapter ($A_{M\to\Phi}$) and immediately decode it back using the same adapter ($A_{\Phi\to M}$). The output serves as the normalized form. This is akin to transliterating a script onto itself via the intermediate phonetic space.
- *Lexicon-based*: Encode the input word. Search a predefined lexicon of canonical forms to find entries that are phonetically similar (above a certain threshold) using the *Sawtone* similarity metric (Elgeish 2019). Replace the input word with the most similar canonical form found.
- *Frequency-based*: First, build a frequency dictionary from a relevant corpus. Then, encode the input word. Find phonetically similar words (above a threshold) within the frequency dictionary. Replace the input word with the most frequent word among the similar candidates.

5.3.1. Experimental setup

Datasets:

- *Naive encoding-decoding*: We used a manually created dataset of 200 Moroccan Arabic (MA) sentence clusters. Each cluster contained 10 sentences (3-5 words each) representing the same sentence written differently, along with a ground truth canonical sentence form.
- *Lexicon*: We used a standard Arabic social media dataset (manually annotated) and a Standard Arabic lexicon compiled from various sources (containing ~100K words) (al-Khalīl 8th century; Ibn Ḥammād al-Jawharī 10th-11th century; Ibn Sīda 11th century).
- *Frequency*: We utilized a subset of a 1-million-word MA corpus gathered from online sources (Section 5.2) to build the frequency list.

Adapter: The same adapters developed for previous tasks were used here. **Thresholds**: Similarity thresholds for the lexicon and frequency-based methods were determined empirically via grid search. **Evaluation**: Output compared against ground truth (for Encoding/Decoding and lexicon) or against the most frequent word in the corpus (for Frequency).

5.3.2. Metrics & results

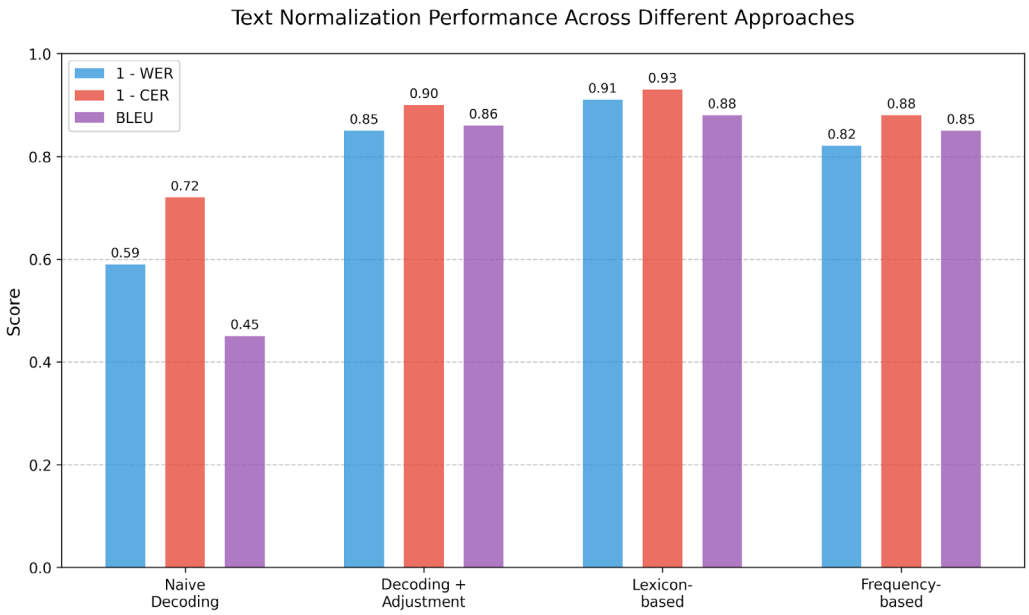

Figure 13: Comparison of accuracy for different normalization approaches

WER/CER: Word/character differences compared to the reference. Lower is better. **BLEU score**: Character sequence overlap compared to the reference. Higher is better.

Coverage (Lexicon/Frequency): The percentage of input words for which a phonetically similar variant was found above the threshold, allowing normalization to be attempted. **Processing speed**: Measured in tokens processed per second (after the initial build phase for lexicon/frequency methods).

Interpretation: It is important to note that the "ground truth" reference differs between methods (manual annotation vs. corpus frequency), making direct comparisons across methods challenging and the choice of methodology a circumstantial choice. Contextual normalization, which considers the surrounding sentence context, was not implemented due to dataset limitations and remains an area for future work. (Fig. 13) provides a visual comparison.

5.3.2.1. Normalizing by encoding and decoding

Comparing the output of the naive encode-decode method to the manual ground truth yielded less than satisfactory results (Table 10). This is likely because the adapter's inherent writing style might differ significantly from typical human writing conventions—a limitation of the specific adapter implementation rather than the *Sawtone* framework itself.

Table 10: Performance metrics for decoding-based normalization approach (vs. Human Ground Truth)

Metric	Value
Word Error Rate	0.41
Character Error Rate	0.28
BLEU Score	0.45
Processing Speed	15,000 tokens/second

However, evaluating the adapter against its *own* writing style as the ground truth reveals its intrinsic consistency (Table 11). Further rule-tuning could align the adapter's output more closely with human style.

Table 11: Performance metrics using adapter's writing style as ground truth

Metric	Value
Word Error Rate	0.15
Character Error Rate	0.10
BLEU Score	0.86
Processing Speed	15,000 tokens/second

5.3.2.2. Lexicon-based text normalization

This method encodes words and then searches a lexicon for the phonetically closest canonical form using the *Sawtone* similarity measure. It can be robust if a high-quality lexicon is available, although this is often not the case for low-resource languages (Bird 2020). Results on a Standard Arabic dataset are shown in (Table 12):

Table 12: Results for lexicon-based normalization on Standard Arabic dataset[1]

Metric	Value
Character Error Rate	0.07
Word Error Rate	0.09
Coverage	0.78
BLEU Score	0.88
Processing Speed	80,000 tokens/second

5.3.2.3. Frequency-based text normalization

This approach does not need a formal lexicon but is sensitive to the clustering threshold, noise within the corpus, and potential ambiguities (e.g., multiple common spellings, or morphologically related words that are phonetically similar but semantically incorrect matches). Results on the 1M-word MA corpus are presented in (Table 13):

Table 13: Performance metrics for frequency-based normalization[2]

Metric	Value
Character Error Rate	0.12
Word Error Rate	0.18
Coverage	0.91
BLEU Score	0.85
Processing Speed	80,000 tokens/second

5.3.3. Error analysis

We manually analyzed 100 errors from each normalization method. Table 14 shows an estimated breakdown of error types *within* each method, aggregated across the methods:

[1] Coverage indicates the percentage of input words matched to a lexicon entry above the similarity threshold.

[2] Speed measured after the initial frequency list construction.

Table 14: Distribution of error types within each normalization method

Error type	Encoding/Decoding (%)	Lexicon-Based (%)	Frequency-Based (%)
Ambiguous (Context Needed)	24	46	36
Legitimate Spelling Variations	14	13	39
Incomplete Phonological Mappings	51	20	15
Other Factors	11	21	10

- **Ambiguous (Context needed)**: Cases where correct normalization requires sentence-level context (constituting roughly 35% of total errors across methods).
- **Legitimate spelling variations**: Situations where multiple spellings are considered valid or common, making the choice of a single canonical form problematic (around 22% of total errors).
- **Incomplete phonological mappings**: Errors arising because the input orthography lacks sufficient phonetic information (e.g., missing diacritics, very terse spellings) (approximately 28.6% of total errors).
- **Other factors**: Including Out-Of-Vocabulary (OOV) words, unfortunate phonetic collisions between distinct words, length mismatches, etc. (about 14% of total errors).

6. Case study: Moroccan Arabic LLM training using *Sawtone*

To showcase *Sawtone*'s practical utility in a demanding, low-resource, high-variation setting, we applied it to preprocess data for training *Sawalni*, a Large Language Model (LLM) specifically designed for Moroccan Arabic (MA).

6.1. Challenges in processing Moroccan Arabic data

MA presents a unique set of challenges for NLP (Habash 2010):

- **Diglossia and script variation**: MA exists alongside Modern Standard Arabic (MSA) and is frequently written informally using both the Arabic script and various Latin-based Arabeezi systems, often mixed within the same text. Neither script has a standardized orthography for MA (Bouamor 2018).
- **Orthographic inconsistency**: The same MA word can be spelled in numerous ways, leading to extensive variation (Darwish 2014).
- **Phonological complexity**: MA possesses distinct phonetic features (like emphatic consonants and vowel reduction) that are often inconsistently represented in writing (Watson 2002).
- **Data scarcity**: Large, clean, and standardized corpora for MA are scarce (Bird 2020; Kamali & Abchir 2024).

Standard text preprocessing techniques often fall short in this context. An approach capable of handling cross-script variation and normalizing inconsistency based on underlying linguistic principles is highly desirable.

6.2. Data collection and initial state

We compiled a diverse text corpus for MA by gathering data from various online sources, including social media platforms, forums, and blogs. This resulted in a heterogeneous dataset containing MA written in both Arabic and Arabeezi scripts, often interspersed with MSA and French, and exhibiting extreme orthographic inconsistency.

6.3. *Sawtone*-powered preprocessing pipeline

We integrated *Sawtone* into our preprocessing pipeline specifically to tackle the challenges inherent in the MA data:

- **Adapter development**: We developed a *Sawtone* adapter tailored for MA, capable of mapping text from both Arabic and Arabeezi scripts into the universal phonological space (Section 3.2). This was achieved using the Crescendo strategy: starting with an initial rule-based adapter, using its output to train a BERT-based adapter, and further fine-tuning this adapter on MA/Arabeezi pairs along with MA-French and MA-MSA homophone pairs via an STS task.

- **Cross-script alignment and data augmentation**:
  - We encoded both Arabic-script and Arabeezi-script MA texts into the common *Sawtone* phonological space using the dedicated MA adapter.
  - We leveraged phonetic similarity (Section 3.4.2) to identify and align parallel text fragments across the two scripts within our corpus.
  - We trained a transliteration model (capable of converting between Arabic MA and Arabeezi) using *Sawtone*'s phonetic mappings as an intermediate representation.
  - We augmented the dataset by generating script-converted versions of monolingual texts, increasing the amount of parallel data available.
- **Phonetic normalization**: To address the pervasive orthographic variation, we applied *Sawtone'*s phonetic clustering capability (Section 4).
  - *Method*: Words were encoded into *Sawtone* vectors. Words whose vectors were closer than an empirically determined threshold were clustered together. A canonical form was selected for each cluster (typically based on frequency). All variants within a cluster were then mapped to this canonical form. We incorporated mechanisms to consider context for particularly challenging cases involving similar-sounding but distinct words.
  - *Example*: This process successfully reduced over 100 different observed spellings for the word *tbarkellah* (تبارك الله, see Table 5) down to a single canonical form. It also helped manage challenging cases like distinguishing *mchina* 'we went' from *machina* 'train', which are sometimes spelled identically in informal contexts.

6.4. Observed impacts on LLM training

Applying *Sawtone*-based preprocessing led to observable changes in the dataset characteristics and suggested potential positive impacts on the LLM training process:

- **Vocabulary reduction**: The number of unique word types in the dataset significantly decreased, from approximately 600,000 in the raw data to around 150,000 after processing. This reduction helps alleviate input layer sparsity for the LLM.
- **Improved data consistency**: Mapping diverse spelling variants to canonical forms based on phonetic equivalence resulted in a more consistent input stream for the LLM. This allows the model to better focus on learning underlying semantic and syntactic patterns rather than grappling with superficial spelling variations (Lourentzou 2019).
- **Enhanced cross-script handling**: The alignment and augmentation steps likely improved the model's ability to understand and generate MA across both Arabic and Arabeezi scripts by exposing it to explicitly linked parallel or phonetically related representations during training.
- **Training observations**: Preliminary comparisons between models trained on the raw dataset versus the *Sawtone*-normalized dataset showed an approximate 8% reduction in perplexity for the model trained on normalized data. This suggests potentially faster convergence and better generalization. We also observed more stable training dynamics, characterized by less variance in the loss function and fewer gradient spikes.

6.5. Discussion: Trade-offs in normalization

While phonetic normalization proved effective in managing orthographic chaos and reducing vocabulary size, it inevitably involves trade-offs. Aggressively normalizing text can lead to the loss of potentially meaningful sociolinguistic or stylistic information conveyed through non-standard spellings (Crystal 2011).

Our decision to normalize heavily was a pragmatic one for this initial LLM development effort, prioritizing model convergence and the capture of core linguistic patterns over the preservation of fine-grained orthographic detail. Future work could explore more nuanced normalization strategies, perhaps preserving particular variation types or training representations that remain sensitive to orthographic choices. Furthermore, phonetic ambiguity where normalization might merge similarly spelled but otherwise distinct words requires deeper integration of contextual information (Elgeish 2019).

This case study demonstrates the significant value *Sawtone* can bring to NLP development for low-resource, high-variation languages like Moroccan Arabic. By leveraging phonetic principles, it provides a principled way to bridge script differences and normalize text variation, offering crucial preprocessing capabilities, although the balance between normalization and information preservation warrants ongoing consideration.

7. Discussion

In this section, we reflect on *Sawtone*'s contributions, acknowledge its limitations, and outline potential directions for future research and development.

7.1. Strengths and contributions

Sawtone's primary contribution is its integrated architecture designed to enable consistent and interoperable phonetic-aware text processing across diverse languages and scripts. Its key strengths include:

1. Interoperability foundation: By decoupling the unified phonological space (Section 3.2) from modular, language-specific adapters (Section 3.4 Modality-specific adapters, Section 4.1), *Sawtone* creates a foundation where adapters developed independently—potentially using different methodologies or data sources—can function cohesively within the framework. This facilitates complex cross-script and cross-language tasks.

2. Consistent phonetic representation: The linguistically grounded, quantitative feature space (Table 1) provides a stable foundation for comparing sounds across languages, promoting consistency regardless of the specific adapters used.

3. Demonstrated applicability: We showcased *Sawtone*'s practical utility in core NLP tasks, including transliteration (Section 5.2, Table 9), cross-script sequence alignment (Section 5.1, Table 3, and text normalization (Section 5.3, Tables 10 & 14). The Moroccan Arabic case study (Section 6) further highlights its value, particularly for low-resource languages exhibiting high orthographic variation, by improving data quality for downstream tasks like LLM training.

4. Low-resource and non-standard focus: *Sawtone* is explicitly designed with low-resource languages, non-standard orthographies, and informal digital text (Section 2.4, Section 6.1) in mind. It offers a principled approach to handling variation based on phonetic equivalence, which is crucial for broadening the reach of NLP technologies.

In essence, *Sawtone* provides a structured, interoperable, and phonetically grounded approach that facilitates robust, comparable, and potentially collaborative NLP research and development, especially benefiting work on less-resourced languages and non-standard text varieties.

7.2. Limitations and future work

This initial presentation of *Sawtone* has several limitations that point towards important avenues for future work:

1. Comparative evaluation: The scope of this work did not include extensive benchmarking between the different adapter implementation strategies (rule-based, Text2IPA, BERT-based, see Table 2). Future research should conduct direct comparisons, focusing particularly on aspects like robustness to noise and sensitivity to context.

2. Evaluation rigor: The evaluations presented (Section 5) are preliminary. There is a need for larger, standardized cross-script datasets suitable for benchmarking. More rigorous evaluation protocols, including significance testing, are required. Furthermore, metrics for tasks like alignment and normalization need refinement; for instance, normalization evaluation (Tables 10-13) would benefit greatly from consistent datasets annotated against human-standardized ground truths.

3. Adapter depth and strategy: We did not deeply optimize the specific architectures of the adapters used, nor did we rigorously validate the effectiveness of the Crescendo

strategy (Section 4.3.4) across different scenarios. Further research is needed on optimal adapter implementations and practical testing of interoperability between adapters developed using different methods.

4. Feature space optimization: While the proposed 18-dimensional space (Section 3.2) proved effective for the tasks explored, a deeper analysis is warranted. This includes investigating the optimality of the chosen dimensional mapping, comparing it systematically to alternative feature systems, and conducting sensitivity analyses regarding feature choices and weights.

5. Normalization trade-offs: As discussed (Section 6.5), aggressive normalization can lead to information loss. Developing more nuanced, potentially context-aware normalization strategies (Section 5.3.3) that strike a better balance between achieving consistency and preserving meaningful variation is an important direction.

6. Ambiguity handling: The current mechanisms for handling phonetic ambiguity (Section 3.4.1, Section 3.4.2) are functional but could be improved. Exploring more sophisticated ranking or selection strategies for multiple phonetic interpretations could enhance accuracy.

7. Theoretical extensions: The framework could be extended theoretically, for example, by refining the feature space (incorporating dynamic weights, hierarchical structures, exploring phonological universals (Chomsky 1968), adding suprasegmentals (Lehiste 1970)), investigating advanced neural architectures (end-to-end models (Graves 2014), self-supervised learning (Baevski 2020; Conneau 2020; Devlin 2018), transfer learning (Pan 2020)), integrating audio modalities (for Speech-to-Text (Chan 2015; Wu 2016) or direct speech alignment (Watanabe 2017)), and developing standardized evaluation frameworks and benchmarks specific to cross-script phonetic tasks (Graham 2013; Karimi 2011; Wang 2019).

***. Technical improvements**: Practical enhancements could include optimizing computational performance – parallelization, efficient search algorithms, distributed processing – improving usability through better tooling, such as IDE plugins, web services, mobile app integration, and creating more readily available resources, such as annotated corpora (Kunchukuttan 2018), diverse datasets (Wang 2019), pretrained adapter models (Devlin 2018).

Addressing these limitations will require focused research efforts and likely community collaboration. *Sawtone* aims to provide a solid foundation upon which such interoperable phonetic processing advancements can be built.

8. Conclusion

In this paper, we introduced *Sawtone*, a universal framework designed for cross-script phonetic alignment and normalization, aimed at addressing the inherent challenges in processing text across diverse writing systems. By integrating principles from phonological theory with practical engineering considerations, *Sawtone* offers a flexible and robust system capable of handling arbitrary script pairs through the use of a shared, universal phonological representation space.

Its effectiveness has been demonstrated across multiple languages and applications, proving particularly valuable for low-resource languages and systems involving alloglottography. The case study focused on normalizing Moroccan Arabic data for LLM training (Section 6) highlights its potential for real-world impact. Key contributions include the universal feature space itself, the flexible adapter architecture promoting interoperability, the capacity for robust handling of non-standard text based on phonetic principles, and its design for minimal resource requirements.

As digital communication continues to expand and more of the world's languages gain a digital presence (Crystal 2012), frameworks like *Sawtone* become increasingly vital. They offer a path towards more effective cross-script text processing and play a role in preserving linguistic diversity in the digital realm (Bird 2020). We hope this work encourages further research and collaboration to expand *Sawtone*'s capabilities and broaden its impact.

Acknowledgments

This research was conducted by the *Sawalni Team* at *Omneity Labs* (echoing أمنيتي, Arabic for 'my wish'). We extend our gratitude to colleagues who provided valuable insights and to members of the Moroccan Arabic speaking community for their input during adapter development. We also thank the anonymous reviewers for their constructive comments and acknowledge the open-source community for the indispensable tools and libraries utilized in this work.

References

Ameur, M. S. H. & Meziane, F. & Guessoum, A. 2019. ANETAC: Arabic named entity transliteration and classification dataset. (arXiv:1907.03110).

Ansari, Z. 2017. Improving text normalization by optimizing nearest neighbor matching. https://doi.org/10.48550/arXiv.1712.09518.

Archangeli, D. B. & Pulleyblank, D. 1994. *Grounded phonology*. Cambridge, MA: MIT Press.

Baevski, A. & Zhou, H. & Mohamed, A. & Auli, M. 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. https://doi.org/10.48550/arXiv.2006.11477.

Bird, S. 2020. Digital support for threatened languages: Progress and challenges. *Computer* 53(4). 82-85.

Bird, S. & Klein, E. 1994. Phonological analysis in typed feature systems. *Computational Linguistics* 20(3). 455-491.

Bouamor, H. & Hassan, S. & Habash, N. 2018. The MADAR Arabic dialect corpus and lexicon. In Calzolari, Nicoletta & Choukri, Khalid & Cieri, Christopher & Declerck, Thierry (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 3387-3396. Miyazaki: European Language Resources Association.

Chan, W. & Jaitly, N. & Le, Q. & Vinyals, O. 2015. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. (arXiv:1508.01211).

Chomsky, N. & Halle, M. 1968. *The sound pattern of English*. Chicago: The University of Chicago Press.

Clements, George N. 1985. The geometry of phonological features. Phonology 2(1). 225-252. https://doi.org/10.1017/S0952675700000440.

Clements, George N. & Hume, E. V. 1995. The internal organization of speech sounds. In Goldsmith, J. A. (ed.), *The handbook of phonological theory*, 245-306. Cambridge, MA: Blackwell.

Conneau, A. & Khandelwal, K. & Goyal, N. & Chaudhary, V. & Wenzek, G. & Guzmán, F. & Stoyanov, V. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, 8440-8451. Abu Dhabi: Association for Computational Linguistics.

Crystal, D. 2011. *Internet linguistics: A student guide*. London: Routledge.

Crystal, D. 2012. *English as a global language*. Cambridge: Cambridge University Press.

Darwish, K. 2014. Arabizi detection and conversion to Arabic. In Habash, N. & Vogel, S. (eds.), *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. 217-224. Doha: Association for Computational Linguistics.

De Premare, A.-L. 1998. *Dictionnaire arabe-français (dialecte marocain)*. Paris: L'Harmattan.

Devlin, J. & Chang, M.-W. & Lee, K. & Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J. & Doran, C. & Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, vol 1, 4171-4186, Minneapolis: Association for Computational Linguistics.

Doherty, L. n.d. Ipa-dict – monolingual wordlists with pronunciation information in IPA. (https://github.com/open-dict-data/ipa-dict) (Accessed 2025-05-25).

Elgeish, M. 2019. Learning joint acoustic-phonetic word embeddings for speech recognition. (arXiv:1908.00493).

Gessler, L. & Zeldes, A. 2022. MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning. In Ataman, D. etc. (eds.), *Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL)*, 86-99. Abu Dhabi: Association for Computational Linguistics.

Goldsmith, J. A. 1990. *Autosegmental and metrical phonology*. Oxford: Basil Blackwell.

Gordon, M. K. & Ladefoged, P. 2001. Phonation types: A cross-linguistic overview. *Journal of Phonetics* 29(4). 383-406.

Graham, Y. & Baldwin, T. & Moffat, A. & Zobel, J. 2013. Continuous measurement scales in human evaluation of machine translation. In Pareja-Lora, A. & Liakta, M. & Dipper, S. (eds.), *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 33-41. Sofia: Association for Computational Linguistics.

Graves, A. & Jaitly, N. 2014. Towards end-to-end speech recognition with recurrent neural networks. In Xing, E. P. & Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*. 1764--1772. Beijing: PMLR.

Habash, N. Y. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies* 3(1). 1-187. https://doi.org/10.2200/S00277ED1V01Y201008HLT010.

Ibn Ḥammād al-Jawharī, Ismāʿīl. n.d. *Al-Ṣiḥāḥ fī al-lughah*. Bayrūt: Dār al-Fikr.

Ibn Sīda. 2000. *Al-Muḥkam wa-al-muḥīt al-aʿẓam*. Bayrūt: Dār al-Kutub al-ʿIlmiyya.

International Phonetic Association. 1999. *Handbook of the IPA: A guide to the use of the international phonetic alphabet.* Cambridge: Cambridge University Press.

Kamali, Omar. & Abchir, M. 2024. Finding Moroccan Arabic (Darija) in Fineweb 2. (https://huggingface.co/blog/omarkamali/gherbal-multilingual-fineweb-moroccan-arabic) (Accessed 2025-05-25).

Karimi, S. 2011. Machine transliteration survey. ACM Comput. Surv. 43. 17. https://doi.org/10.1145/1922649.1922654.

al-Khalīl, Ibn Aḥmad al-Farāhīdī. 2003. *Kitāb al-ʿAyn*. Bayrūt: Dār al-Kutub al-ʿIlmiyya.

Knight, K. & Graehl, J. 1998. Machine transliteration. *Computational Linguistics* 24(4). 599-612.

Kondrak, G. 2003. Phonetic alignment and similarity. *Computers and the Humanities* 37(3). 273-291. https://doi.org/10.1023/A:1025071200644.

Koskenniemi, K. 1983. Two-level model for morphological analysis. IJCAI'83. *Proceedings of the Eighth international joint conference on Artificial intelligence*, vol. 2, 683-685. Karlsruhe: Morgan Kaufmann Publishers.

Kunchukuttan, A. & Mehta, P. & Bhattacharyya, P. 2018. The IIT Bombay English-Hindi parallel corpus. In Calzolari, N. & Choukri, Kh. & Cieri, C. & Declerck, T. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki: European Language Resources Association.

Ladefoged, P. & Johnson, K. 2011. *A course in phonetics*. Boston, MA: Cengage Learning.

Ladefoged, P. & Maddieson, I. 1996. *The sounds of the world's languages*. Oxford: Blackwell Publishing.

Lehiste, I. 1970. *Suprasegmentals*. Cambridge, MA: MIT Press.

Li, X. & Metze, F. & Mortensen, D. & Watanabe, S. & Black, A. 2022. Zero-shot learning for grapheme to phoneme conversion with language ensemble. In Muresan, S. & Nakov, P. &Villavicencio, A. (eds.),

Findings of the Association for Computational Linguistics: ACL 2022, 2106-2115, Dublin: Association for Computational Linguistics.

Llama 3 Team. 2024. The llama 3 herd of models. (arXiv:2407.21783).

Lourentzou, I. & Manghnani, K. & Zhai, C. X. 2019. Adapting sequence to sequence models for text normalization in social media. In Calzolari, N. & Choukri, K. & Cieri, C. & Declerck, T. (eds.), *Proceedings of the Thirteenth International AAAI Conference on Web and Social Media (ICWSM 2019)*, 335-345. Munich: AAII.

Qwen Team. 2024. Qwen2.5 technical report. (arXiv:2412.15115).

Mortensen, D. 2018. Epitran: Precision G2P for many languages. In Calzolari, N. & Choukri, K. & Cieri, C. & Declerck, T. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2710-2714. Miyazaki: European Language Resources Association (ELRA).

Naji, N. & Allan, J. 2016. *On Cross-Script Information Retrieval*. 9626. 10.1007/978-3-319-30671-1_70.

Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48(3). 443-453. DOI:10.1016/0022-2836(70)90057-4.

Ni, J. 2018. Multilingual grapheme-to-phoneme conversion with global character vectors. *Interspeech 2018*. 2823-2827. https://doi.org/10.21437/Interspeech.2018-1626.

Pan, L. 2020. Multilingual BERT post-pretraining alignment. In Toutanova, K. & Rumshisky, A. & Zettlemoyer, L. & Hakkani-Tur, D. & Beltagy, I. & Bethard, S. & Cotterell, R. & Chakraborty, T. & Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 210-219. Abu Dhabi: Association for Computational Linguistics.

Radford, A. & Wu, J. & Child, R. & Luan, D. & Amodei, D. & Sutskever, I. 2019. Language models are unsupervised multitask learners. (https://api.semanticscholar.org/CorpusID:160025533) (Accessed 2025-05-25)

Raffel, C. & Shazeer, N. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. (arXiv:1910.10683).

Rosca, M. & Breuel, T. 2016. Sequence-to-sequence neural network models for transliteration. (arXiv: 1610.09565).

Sälevä, J. & Lignos, C. 2022. ParaNames: A massively multilingual entity name corpus. (arXiv:2202.14035).

Sharma, D. 2021. Learning phonetic word embeddings. (arXiv:2109.14796).

Sonmez, O. 2014. Graph-based text normalization. In Moschitti, A. & Walter, B. & Daelemans, W. (eds.), *Proceedings of the 2014. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 313-324, Doha: Association for Computational Linguistics.

Sproat, R. & Jaitly, N. 2016. RNN approaches to text normalization: A challenge. (arXiv:1611.00068).

Stevens, K. N. 1998. *Acoustic phonetics*. Cambridge, MA: MIT Press.

Unseth, P. 2005. Sociolinguistic parallels between choosing scripts and languages. *Written Language & Literacy* 8(1). 19-42.

Wang, A. & Pruksachatkun, Y. & Nangia, N. & Singh, A. & Michael, J. & Hill, F. & Levy, O. & Bowman, S. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In Wallach, H. & Larochelle, H. & Beygelzimer, A. & d'Alché-Buc, F. & Fox, E. & Garnett, R. (eds.), *Advances in neural information processing systems, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2190-2194. Vancouver: Association for Computational Linguistics.

Watanabe, S. & Hori, T. & Kim, S. & Hershey, J. R. & Hayashi, T. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *Journal of Selected Topics in Signal Processing* 11(8). 1240-1253. https://doi.org/10.1109/JSTSP.2017.2763455.

Watson, J. C. E. 2002. *The phonology and morphology of Arabic*. Oxford: Oxford University Press.

Wu, Y. & Schuster, M. & Chen, Z. & Le, Q. V. & Norouzi, M. & Macherey, W. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. (arXiv:1609.08144).