Kabyle ASR Phonological Error and Network Analysis

Christopher Haberland and Ni Lao

Abstract Training on graphemes alone without phonemes simplifies the speech-to-text pipeline. However, models respond differently to training on graphemes of different writing systems. We investigate the impact of differences between Latin and Tifinagh orthographies on automatic speech recognition quality on a Kabyle Berber speech corpus. We train on a corpus represented in a Latin orthography marked for vowels and gemination and subsequently transliterate model output to a consonantal Tifinagh orthography not marked for these features, which results in 10% absolute improvement in word error rate over a model trained on the unmarked orthography. We find that this performance gain is primarily due to a reduced error rate for graphemes marked for vocalic and voiced consonantal phonemes. However, this overall improvement is tempered by a reduction in recognition quality for other phonemes, especially allophonic spirantized consonants that are replete in the Kabyle language and many Berber dialects more widely. We also introduce new methods to characterize the disparity in performance between ASR models by analyzing outputs in terms of phonological networks. To our knowledge, this is the first work analyzing phonological networks of artificial neural network speech model outputs. Our results suggest that inputs written in defective orthographies lead to worse recognition quality for modern speech-to-text architectures compared to those fully marked for vowels and gemination.

Ni Lao contributed to this chapter while he was working at SayMosaic Inc.

C. Haberland (✉)
USAA, San Antonio, TX, USA
e-mail: crh2ke@virginia.edu

N. Lao
Google, Mountain View, CA, USA

M. Abbas (ed.), *Analysis and Application of Natural Language and Speech Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-031-11035-1_3

1 Introduction

Graphemic modeling units and their correspondence with the spoken word can vary between different language communities [1], and even a single language community may have multiple orthographic conventions for application in different contexts [2] (diglossia). Minority languages in particular have often undergone less standardization [3], contributing to a greater tendency to be written in multiple orthographies. Improving speech technologies to support minority and "low-resource" languages and orthographies is crucial to ensuring their vitality and their users' access to information in the digital era [4]. Poor quality of low-resource language systems can compel users to interact with ASR systems in languages of which they are non-native, diminishing use of their native language. Furthermore, high error rates for a low-resource language ASR systems disadvantage monolingual speakers of the low-resource language that have a limited ability to switch to systems in more prevalent languages with better recognition quality.

Modern speech-to-text (S2T) models are trained on audio data paired with sequences of modeling units [5], which may be graphemes, phonemes, or other representations [6] that represent the linguistic constituents. Training models on phonemes constitutes a general paradigm in the creation of S2T systems [7] especially in the context of low-resource languages [8]. Training on phonemes can be advantageous for decoding out-of-vocabulary words or words from an external language [9], but manual annotation of speech data can be prohibitively expensive for low-resource languages [4].

ASR pipelines often include a component to automatically generate phoneme-based training data through grapheme to phoneme (G2P) conversions [10, 11] by training supervised models [12–14] or constructing rule-based systems [15]. There is an emerging trend toward G2P conversion with minimal intervention and preparation to streamline the end-to-end learning process. Several systems intend to streamline the G2P process using different methods, including self-training [16], ensembles of varying degrees of supervision [17], and leveraging open dictionaries of high-resource languages [18]. For low-resource languages, training S2T systems with graphemes alone obviates the G2P step in the S2T pipeline and the need for language-specific expert annotations [19]. There is also evidence that neural speech models implicitly learn phonemes at intermediate hidden layers when training on graphemes [20]. A number of different methods, such as diagnostic probing techniques and Representational Similarity Analysis, have been applied to show phonological learning by peering into neural model internals [21]. However, further research is required to show how S2T model quality responds to training on graphemes of various writing systems and orthographies in practice.

In this chapter, we study the impact of using a fully featured orthography instead of a consonantal orthography on S2T performance for Kabyle, a Berber language of northern Algeria. We chose to experiment with this language to augment the discussion surrounding orthographic choice on S2T quality that has been conducted primarily on Semitic languages that are comparatively more resourced, such as

Arabic. While several previous studies [22–24] have demonstrated the effect of training and decoding using defective (*i.e., omitting vocalic information*) and non-defective orthographies separately, our study is the first to compare a neural speech model's performance between (a) training and decoding in a defective orthography and (b) training on a non-defective orthography and decoding into its defective representation. Our study is also the first to analyze the nature of phonemic errors made by neural ASR models trained on a corpus in a defective and non-defective orthography to understand any systematic difference of types of errors made by models trained on these orthographies. The results demonstrate the importance of including vocalic graphemic inputs for improved S2T recognition of vowels and voiced consonants. To our knowledge, this result represents the first S2T system trained on a Tifinagh-encoded corpus of a Berber language. Our study is also unique in being the first to apply phonological network methods to characterize the differences between phonological networks between neural ASR model outputs to compare them against those of their respective gold vocabularies. We find that phonological networks of learned ASR vocabularies are significantly denser and less modular than gold vocabularies and publish our network data to encourage further investigations of phonological networks of ASR models.

2 Background

2.1 ASR Modeling Units

The investigation of orthographic choices on S2T system performance parallels psycholinguistic and cognitive science research on humans' linguistic and conceptual comprehension from auditory and visual information. A significant body of research aims to uncover how different G2P correspondences across writing systems may predict reading level achievement and interactions with dyslexia [25, 26]. For example, the work [27] assesses the reading abilities of children diagnosed with dyslexia when taught a novel orthography consisting of new G2P mappings. The work [28] studies the effect of diacritization and non-diacritization of dyslexic and non-dyslexic readers' processing of the Arabic script and found spelling knowledge of study participants to be the most significant predictor of processing speed.

S2T learning solely with graphemes has a long history [29]. Recently, studies have focused on identifying the differences between training on phonemic and graphemic inputs. The authors in [30] report that the phonemic–graphemic performance gap closes when model architecture and hyperparameters are attuned to the specific data input. Rao and Sak [31] found improved performance of graphemically trained models in multi-accented corpora and in trials of increased input data scale. Other work has tested derivatives of graphemes, such as bytes [32], wordpieces [31], and context-dependent graphemes (i.e., chenones) [19, 33]. Wang et al. [33] achieved state-of-the-art error rates on English data with graphemically derived modeling units for English.

2.2 Diacritization

Imputation of diacritics to augment defective model inputs has been, and continues to be, another active area of research, especially in the context of Arabic speech-to-text system design [34–38]. Diacritic imputation systems are designed to help computational models resolve heterophonic homographs or congruent graphemic sequences that have multiple phonemic interpretations, in orthographies that do not mark certain features. Sequences of this type are prevalent in consonantal writing systems, such as that used for Arabic, in which roughly one-third of tokens may be pronounced differently when not diacritized [39].

There has been work investigating diacritization's effect on speech modeling in languages that are written in defective orthographies or those not marked for certain phonemes. Afify et al. [40] used HMMs to demonstrate that training on voweled graphemes could increase performance over training on unvowelled graphemes on Arabic broadcast transcripts, even when decoding into unvowelled text. However, to the authors' knowledge, this has not been demonstrated in modern neural speech models with CTC decoding. More recently, [22] showed that training neural acoustic models upon voweled graphemes generally improved WER over unvowelled graphemes *when decoding into the same orthography*. The authors of [41] pre-annotate training transcripts with phonetic information deduced from graphemic context with rules to improve system performance. Alshayeji et al. [23] and Al-Anzi and AbuZeina [24] compare diacritized and non-diacritized input with various S2T model architectures and hyperparameters and observe higher WER for diacritized trials, though they do not train on diacritized data and decode on non-diacritized data.

Augmenting inputs via transliteration has been shown to improve S2T systems or machine translation performance. The authors of [42] transliterate model output as a post-process to improve the recognition of code-switched speech. Le and Sadat [43] and Cho et al. [44] model the G2P task as a neural sequence-to-sequence model and record improvements in named entity recognition and code-switched speech for Vietnamese and mixed Korean–Chinese scripts, respectively. While these studies use neural G2P models, rule-based systems have been developed for under-resourced languages [45, 46].

2.3 Berber Language Tools

To date, there have been limited efforts that apply neural speech models to Berber languages. OCR techniques have been applied to Tifinagh recently [47, 48], and [49] produced a pronunciation dictionary for speech modeling of phonemes. However, to the best of our knowledge, the ASR research community has not documented the training of Berber S2T models aside from those produced from the CommonVoice initiative [50] trained with a Latin-script corpus. We cite [51] as a limited exception, who describe a speech recognition system for Tarifit to recognize spoken numbers in noisy environments.

2.4 Phonological Networks

Phonological network analysis stems from investigations of how humans organize, internalize, recall, and reproduce words during linguistic processing. The authors of [52] were among the first to computationally model and understand the network effects of word similarity on auditory word recognition. Vitevitch [53] and later authors apply the methodology to understand the properties of phonological networks for specific language vocabularies and also to compare them. The authors in [54] use phonological networks to attempt to discover the differences between lexicon structures of different languages, identifying a robustness of connectivity of the networks in response to node removal and small giant components compared to complex networks in other domains. These authors also reveal general trends and commonalities between various languages across phonological network statistics.

Shoemark et al. [55] further improves the methodology and argues the need to control for network size (vocabulary size), to establish baselines for random networks governed by similar properties, and account for morphological processes by casting vocabularies as sets of lemmata. This resulted in more robust measures for phonological network comparison across languages while controlling for expected network variability, word length, and phonological inventory size. They found that most languages exhibit very similar network statistics trends as a result of the way phonological networks are defined but note certain cross-linguistic differences in the average shortest path length (ASPL) and small-world property between languages at different sampled vocabulary sizes. However, the authors did not find conclusive evidence that phonological networks represented "deeper organization within language" as [54] stated.

Beyond comparative linguistics, studies have also attempted to study monolingual phonological networks in the context of language acquisition. Siew [56] uses Louvain optimization to find communities among the phonological network analyzed by Vitevitch [53] and finds that larger communities are more likely to contain short, frequent, and highly connected words and low average age of acquisition ratings and a clustering effect of similar phonological segments in each community. The authors in [57] find an inverse preferential attachment effect as new words are acquired in language learners' networks. Siew and Vitevitch [58] extends the phonological network to cover orthographic differences to uncover joint effects on visual word recognition and spoken word recognition. Neergaard et al. [59] studies monolingual and bilingual speakers of Mandarin and English to understand the differences in structure, cohesion, and interconnectedness of elicited phonological networks. Turnbull [60] describes the graph-theoretic properties of the most common type of phonological networks applied in this literature. Despite considerable efforts to apply connectivist-theoretical methods in the realms of psycholinguistics and comparative linguistics, phonological networks have not heretofore been applied to the analysis of computational speech recognition models to compare lexical network structures with individual and language-wide vocabularies. To our knowledge, no prior work has described phonological networks of artificial neural network speech model outputs.

3 The Kabyle Language and Berber Writing Systems

Kabyle is a Berber language spoken in northern Algeria that has historically been written in Latin, Arabic, and Tifinagh scripts. Contemporary Kabyle is most widely written in a Latin orthography popularized by the linguist Mouloud Mammeri in a 1976 grammar of the language, though the Arabic and Tifinagh scripts are still promoted among certain groups within Algeria society [61]. Souag [61] contends that the Latin script predominates over the others in modern usage.

The alphabetic Neo-Tifinagh orthographies came into use after language planning initiatives for the Berber languages in the mid-twentieth century carried out by organizations such as Morocco's IRCAM (Amazigh), the Nigerien APT (Tuareg) [62], and the Académie berbère (Kabyle) [61]. The consonantal Tifinagh orthographies are not commonly used to write Kabyle. However, we transliterate Kabyle into a consonantal Tifinagh orthography to expand the incomplete literature on decoding into defective orthographies, which has primarily focused on Semitic languages. To our knowledge, no prior study has trained or decoded a speech model for a Berber language using Tifinagh inputs or a consonantal Tifinagh orthography.

We outline the fundamental differences between the Latin Kabyle orthography and the consonantal Tifinagh orthography: the first is that the Latin marks for gemination via digraphs, unlike the traditional Tifinagh. In some dialects, singletons are often spirantized as opposed to their geminated counterparts (e.g., "tt" from "t"). In the Latin orthography, these doubled consonants are phonemically "tense" and correlate with increased pronunciation length [63], register a fortis-lenis contrast that includes devoicing, and can form minimal pairs [64]. A consonantal Tifinagh orthography introduces additional heterophonic homography by graphically equating tense sounds with their non-tense counterparts.

The second fundamental difference is of vowel denotation. Although vowels are written in all contexts in Neo-Tifinagh orthographies, they are not marked save for word-final positions in the traditional Tifinagh orthographies [65, 66]. From the set of Tifinagh characters that may represent vowels, only "ⴰ" exclusively represents non-glide vowels (for "a," "ə"[1]) while "ⵓ" ("u") and "ⵉ" ("i") also represent semi-vowels ("w" and "j," respectively). These latter two graphemes are analogous to the *matres lectionis* of Semitic language scripts [67].

A final difference is that certain Tifinagh orthographies make use of ligatures that elide certain sequences of adjacent graphemes. The number of attested ligatures across the many varieties of traditional Tifinagh is vast [66], and most are not supported by Unicode.[2] We test the effect of ligatures by encoding those used in the Ahaggar orthography [65] as distinct characters in trial (1c) described in Section 5.

[1] We do not find attestations of "ⴻ" in the traditional Tifinagh orthographies described in [65]. We transliterate word-final "e" (primarily in loan-words) as "ⴰ,".

[2] https://www.unicode.org/charts/PDF/U2D30.pdf.

4 Approach

4.1 Mozilla CommonVoice

We use the original CommonVoice Kabyle corpus for all experiments.[3] The audio-transcript pairs come from Mozilla's CommonVoice crowdsourced initiative [50], which has collected data for over 54 languages at the time of writing. All corpora are released with train/dev/test subsets, and a unique speaker may appear in only a single set among each split. Most utterances are derived from Wikipedia, but some have been added by annotators through the language community's Pontoon page.[4] We removed special symbols and normalized Unicode characters of similar graphical appearance to ensure that characters intended to represent a single grapheme were treated as such.[5]

4.2 Mozilla DeepSpeech

For S2T model training, we use Mozilla's DeepSpeech pipeline, which is based on the DeepSpeech framework [68] and is maintained by a large community. After parameter tuning, we found that the default hyperparameters worked well. For all experiments, we used models of 1024 hidden units and trained for 50 epochs, with a learning rate of 0.0001 and a dropout of 0.3. We used batch sizes of 32, 16, and 16 for train, dev, and test sets, respectively. We used the default trigram settings for training the LM with KenLM [69] in our experiments.

4.3 Transliterator

To convert the Latin-script CommonVoice corpus to the Tifinagh orthographies in our experiments, we use the `Graph Transliterator` Python package[70]. This constructs a directed tree of ranked transition rules (e.g., **mm** -> ⵎ (not ⵎⵎ) because **mm** -> ⵎ ranks before **m** -> ⵎ) to convert between Latin and Berber orthographies. We write rules for two distinct defective orthographies modeled after [65]'s description of the Ahaggar variant of Tiginagh—one with ligatures and one without. In cases where multiple Unicode graphemes represent the same phonemes across Berber languages and orthographies (e.g., ⴿ, ⵗ), we opted to use the symbol closest to that described in [65]. Heterophonic homographs in the Latin corpus

[3] Accessed April 2020, 4th ed.

[4] https://pontoon.mozilla.org/projects/common-voice/.

[5] For example, ε and € were converted to ɛ (U+025B).

Table 1 Kabyle commonvoice data statistics

Split	Downloaded	Processed	Length
Train	37,056	35,715	35 hrs, 24 min
Dev	11,482	11,100	10 hrs, 52 min
Test	11,483	11,125	11 hrs, 42 min

Table 2 Normalization and transliteration examples

Original	Normalized	Tifinagh Transliteration
D tasnareft taserdasit i yettreṣṣin deg Lezzayer.	**d tasnareft taserdasit i yettreṣṣin deg lezzayer**	ⴷ ⵜⵙⵏⵔⴼⵜ ⵜⵙⵔⴷⵙⵜ ⵉ ⵉⵜⵔⵙⵏ ⴷⴳ ⵍⵣⵉⵔ
Teĉĉiḍ iles-ik waqila?	**teččiḍ iles ik waqila**	ⵜⵞⴹ ⵍⵙ ⴾ ⵓⵈⵍⴰ
Σerḍeγ-t-id ad yekkes lxiq, yeẓẓel iḍarren.	**ɛerḍeɣ t id ad yekkes lxiq yeẓẓel iḍarren**	ⵄⵔⴹⵗ ⵜ ⴷ ⴷ ⵉⴾⵙ ⵍⵆⵈ ⵉⵥⵍ ⴹⵔⵏ
Tawaγit d lmehna d-yeγdel ṭrad γef tmurt.	**taɣaɣit d lmeḥna d yeɣḍel ṭrad ɣef tmurt**	ⵜⵓⵗⵜ ⴷ ⵍⵎⵃⵏⴰ ⴷ ⵉⵗⴹⵍ ⵟⵔⴷ ⵗⴼ ⵜⵎⵔⵜ

remain as such in the transliterated Tifinagh (e.g. ‘d’ represents both ‘d’ and ‘ð’, and is transliterated as “V” and not the IRCAM “V.” All Kabyle phonemes that do not have distinct graphemes in the orthography described in [65] are represented with a corresponding Neo-Tifinagh symbol (e.g. -> ⵞ, -> ⵕ) (Tables 1 and 2).

4.4 Sequence Alignment

We sought to investigate which, and to what degree, phonemic classes are affected by different training orthographies. To facilitate this analysis, we required a tool to align the graphemic output sequences from the ASR systems, such that the aligned character pairs represented the audio data at the same time periods in the input data. We considered multiple techniques for matching the output sequences between the gold input and the inferences of the two models. One potential approach was to use an acoustic alignment model (e.g., the Montreal Forced Aligner [71] or DSAlign [72]), though this method risked substantial error propagation for our analysis. We also considered extracting time-aligned CTC model internals to understand the exact timesteps at which outputs were predicted with respect to the gold data. However, we felt that we could achieve the same results with Sound-Class-Based Phonetic Alignment (SCA) [73] with substantially reduced effort. Sound-Class-Based Phonetic Alignment (SCA) [73] was possible due to the relatively high degree of transparency or unambiguous correspondence between graphemes and phonemes [74] of the Kabyle Latin script. To implement SCA, we use the *prog_align* function contained in the `LingPy` package [75], which constructs a similarity matrix and applies a Neighbor-Joining algorithm (see [76]) to construct a guide tree to successively align phonemic sequences. A dynamic programming routine finds a least-cost path through the matrix to align the multiple sequences according to similar sound classes. We alter the default SCA sound class matrix values to ensure that Tifinagh *matres lectionis* graphemes ((‘j’ | ‘ⵉ’) => ‘I’, (‘w’ | ‘ⵓ’) => ‘Y’) could align with both vowels and semi-glides from the Latin gold transcripts. We find that this approach gives accurate alignment for phonemic sequences. We found no apparent errors after manually inspecting a thousand aligned phoneme pairs.[6]

5 Experimentation and Results

We present our result comparing S2T performance when training on orthographies of varying degrees of phonemic informativeness and analyzing phonemic confusion using sequence alignment techniques.

5.1 Experiments

First, we test the hypothesis that training and testing upon an orthography unmarked for vowels, as opposed to marked, yields lower ASR word error rates. Because the Tifinagh input only registers *matres lectionis* at the end of words, we expect that most intra-word vocalic signals are lost during the training process on the Tifinagh orthography compared to training on the Kabyle Latin script. Experiment 1 compares the effect of training and testing upon the Latin-based orthography and transliterated Tifinagh orthography in a set of trials listed in Table 4 (1a–c). In 1a, the Latin corpus is used for training and testing. The outputs were evaluated against Latin gold utterances in the test split. In 1b, we train in the same manner but test by applying a transliterator to convert the Latin test set into the consonantal Tifinagh orthography without ligatures. The corpus used to train the language model (LM) is composed of the transliterated utterances of the original corpus. In the third setup (1c), we repeat experiment 1b using a transliterator that models the ligatures described in Section 3. Examples of the ligatured Tifinagh are shown in Table 3.

Secondly, we test the hypothesis that learning from an orthography marked for vowels and decoding on an orthography unmarked for vowels result in lower word error rates compared to training and testing on either of the marked or unmarked orthographies alone. In experiment 2, we test the hypothesis that training on the

Table 3 Modelling unit experiment (1c) input example. Note: ⧖ and ∃ are stand-in single-character substitutions for ligatures that are not represented in Unicode and are not graphically representative of the traditional graphemes for these ligatures

Non-ligatured	ⵏⴳⵃⵓⵔ	ⵉⵣⴳⵏ	ⵣⵟⵓ	ⵙⴷⵜ	ⵛⵏⴳⴰ	ⵗⵜ	ⵜⵘⵏ	ⴷⵉ	ⵜⵎⵏⵜⵍⵜ
Ligatured	!ⵃⵓⵔ	ⵉⵣⴳⵏ	ⵣⵟⵓ	ⵙⴷⵜ	ⵛ!ⴰ	ⵗⵜ	ⵜⵘⵏ	ⴷⵉ	ⵜⵎ⧖∃

[6] https://github.com/berbertranslit/berbertranslit.

Table 4 The impact of orthography and language modeling. Group 1: trained and tested on the same orthography types. Group 2: Latin to Tifinagh transliteration at test time given a Latin model. Group 3: the same as Group 1 but without language modeling

Exp.	Train orthography	Transliteration	LM	Test orthography	CER (%)	WER (%)
1a	Latin	No	Yes	Latin	29.9	49.9
1b	Tifinagh	No	Yes	Tifinagh	35.8	57.9
1c	Tifinagh (ligatured)	No	Yes	Tifinagh (ligatured)	33.7	57.4
2	Latin	Yes	Yes	Tifinagh	29.7	47.4
3a	Latin	No	No	Latin	34.9	78.3
3b	Tifinagh	No	No	Tifinagh	38.8	77.9
3c	Latin	Yes	No	Tifinagh	35.6	72.1

plene (fully marked) Latin orthography and subsequently decoding into and testing against the defective Tifinagh orthography yield lower error rates compared to both training and testing on the Tifinagh orthography. We train all components on the Latin script and obtain Latin-script output for test utterances as in 1a. However, we then transliterate the output and test against gold utterances transliterated into Tifinagh, as in 1b. Because our main goal is to study the acoustic model and we do not want a small LM training corpus to negatively affect the experimental result, we build the LM in DeepSpeech on all train, dev, and test utterances of the normalized CommonVoice Kabyle Latin-script data for experiments 1 and 2.

Finally, we train the S2T model without an LM as a post-process to specifically understand the sensitivity of the neural speech component. Trials 3a–c replicate 1a–c but do not apply LM post-processing to help understand the effect of our interventions on the neural ASR component.

5.2 *Results*

We report the results of all three sets of trials in Table 4. 1a and 1b show that the original Kabyle input encoded in the plene Latin orthography yields lower error rates than when training and testing on the transliterated Tifinagh alone (CER: −5.9% and WER: −8%). However, this reduction is less pronounced when the ligatured Tifinagh orthography is used (1c) (CER: −3.8% and WER: −7.5%).

Trial 2 exhibits improved recognition when training on the Latin orthography and subsequently transliterating to and testing against Tifinagh. This arrangement reduces CER by 0.2% and WER by 2.5% with respect to trial 1a in which the plene orthography was used for both training and testing. Compared to training and testing in the defective orthography (1b), Trial 2 shows a 10.5% absolute decrease in WER and 6.1% absolute decrease in CER.

Trial 3 shows that, without the language model, the WER for training upon and testing against Latin orthography (3a) is greater than when using the Tifinagh orthography (3b) by 0.4%. However, the CER for the former procedure with respect to the latter is less by 3.9%, likely due to the increased difficulty of predicting more characters. Applying a Tifinagh tansliterator to the Latin trained model (3c) resulted in a WER reduction of 6.2 and 5.8% with respect to 3a and 3b. 3c exhibits an improved CER compared to the Tifinagh-only trial (3b) (−3.2%), although it is 0.7% higher when compared to the Latin-only trial (3a).

5.3 *Phonemic Confusion Analysis*

To understand the orthographies' effects on the speech model, we conduct an analysis by alignment between the gold utterances and the predictions from experiments 3b and 3c. This analysis is inspired by recent studies by Kong et al. [77], Alishahi et al. [78], and Belinkov et al. [6], to explore the nature of neural learning of phonemic information. More specifically, we use the `LingPy` [75] package to determine phone error rates as described in Sect. 4.4. We translate all graphemes of the gold utterances and their predicted counterparts into sequences of G2P IPA representations and tabulate phoneme class confusions using PHOIBLE's sound classes [79]. To understand the models' differential abilities in detecting spirantized consonants, we establish a "spirantized" feature that is attributed to the consonants "t," "d," "k," "g," and "b" that do not present in the contexts where non-continuant stops are the norm. We follow Chaker's description [80] of predictable Kabyle spirantized contexts to estimate this number across the corpus, as spirantized and non-spirantized consonants are commonly homographic in the Latin script. We modify the SCA model to ensure that *matris lectionis* characters are more easily aligned to their respective vowels in the gold Latin-text transcripts. Table 5 shows example aligned sentences produced by this procedure. By analyzing the aligned utterances, we tabulate estimated confusions between the gold and predicted alignments.

We count phonemic disagreements between the models as a proportion of gold target contexts of the aligned matching phoneme. To understand which model achieves better performance for word-final vowel recognition that is denoted in the Tifinagh orthography, we analyze the counts of all gold contexts in which vowels or semi-vowels appear (always word-finally) against the counts of aligned model inferences at these contexts. Table 6 shows that the model trained on the Latin orthography and subsequently transliterated (3c) achieves higher recognition of the pure vowel grapheme compared to the model trained on the unvowelled traditional Tifinagh (3b).

Table 7 compares the errors across several different phonemic classes. We do not consider the "continuant" and "delayedRelease" features, as the distinction between allophonic and phonemic fricativity is difficult to determine for Kabyle from graphemes alone. Although the PHOIBLE database includes these features as "syllabic," we tally counts for the "approximate," "sonorant," and "dorsal," and "periodic glottal source" features without "syllabic" phonemes so as to better analyze the contribution of non-syllabic features. McNemar's asymptotic test with continuity correction [81] affirms the significance of the difference between 3b and 3c ($P < 0.025$ for all features except the "geminate" feature).

Table 5 Alignment of the same sentence produced by different models in Table 4. * indicates a missing space in the alignment. $^+$ indicates Tifinagh-transliterated gold.

Group - train/decode	Raw		Alignment (in IPA representation)																	
3a - Latin/Latin	Gold	yuweḍ ɣer lebɣi s	j	u	w	ə	ḍ	ʁ	ə	r		l	ə	b	*	ʁ	*	i		s
	Pred	yuweḍ ɣaleb ɣ is	j	u	w	ə	ḍ	ʁ	a	-	*	l	ə	b		ʁ		i	*	s
3b - Tifinagh/Tifinagh	Gold$^+$	ⵢⵓⴹ ⵖⵔ ⵍⴱⵖⵢ ⵙ	j		w		ḍ	ʁ		r		l		b		ʁ		j		s
	Pred	ⵢⵓⴹ ⵖⵍⴱⵖ ⵙ	j		w		ḍ	ʁ		-	*	l		b		ʁ		-		s
3c - Latin/Tifinagh	Gold$^+$	ⵢⵓⴹ ⵖⵔ ⵍⴱⵖⵢ ⵙ	j		w		ḍ	ʁ		r		l		b	*	ʁ		j		s
	Pred	ⵢⵓⴹ ⵖⵍⴱ ⵖ ⵙ	j		w		ḍ	ʁ		-	*	l		b		ʁ		-		s

Table 6 Comparison of model performance for different word-final vowels. The columns represent phoneme pairs (Tifinagh grapheme : Latin IPA). Trial 3c shows considerably higher recognition of vowels.

	ⴰ : a/ə	ⵉ : i (j)	ⵓ : u/ (w)	All vowels
The number of word-final vowels in gold	7430	6557	1341	15,328
C_w: The portion (%) of all word-final phonemes	11.7%	10.3%	2.1%	13.0%
C_2: The portion (%) of C_w either 3b (x)or 3c is correct	23.7%	28.1%	30.4%	26.2%
C_3: Both 3b and 3c are incorrect	38.2%	46.8%	34.9%	41.6%
C_{3b}: The portion (%) of C_2 for which 3b is correct	18.5%	13.2%	13.7%	15.6%
C_{3c}: The portion (%) of C_2 for which 3c is correct	**81.5%**	**86.8%**	**86.3%**	**84.5%**

We bold the higher percentage between C3b and C3c

Table 7 Comparison of model performance for different phonemic features. C_p represents the portion (%) of G2P mappings the feature comprises the total number of G2P mappings in the corpus. See the definition of C_2, C_3, C_{3b}, and C_{3c} in Table 6. 3c is correct for more disagreements for all features except for the coronal, strident, and trill features. We use McNemar's asymptotic test with continuity correction [81] to test the null hypothesis that there is no difference between the performance of C_{3b} and C_{3c} with respect to different sound classes. χ_1^2 values are particularly high for voiced and syllabic phonemes. We bold the higher between C_{3b} and C_{3c} when $\chi_1^2 > 18.5$ (corresponding to $P = 0.001$)

	C_p	C_2	C_3	C_{3b}	C_{3c}	χ_1^2
Syllabic (vowels) (word-final)	6.1%	26.2%	41.6%	15.6%	**84.4%**	1902.8
Periodic glottal (voiced) (– syllabic)	36.4%	18.5%	29.2%	42.2%	**57.8%**	407.9
Dorsal (– syllabic)	11.8%	17.7%	28.7%	38.2%	**61.8%**	294.6
Sonorant (– syllabic)	24.0%	18.1%	26.2%	44.1%	**55.9%**	151.4
Nasal	11.5%	17.6%	24.1%	42.0%	**58.0%**	130.1
Spirantized stops (+ voiced)	3.3%	20.1%	34.3%	36.0%	**64.0%**	129.8
Continuant (– syllabic)	28.4%	17.1%	27.5%	45.5%	**54.5%**	98.5
Approximate (– syllabic)	12.6%	18.6%	28.2%	45.9%	**54.0%**	38.2
Consonants	53.1%	16.6%	29.7%	47.9%	**52.1%**	38.9
Non-spirantized stops (+ voiced)	0.5%	24.1%	24.1%	34.8%	**65.2%**	27.1
Labial	11.7%	16.1%	49.8%	35.0%	**65.0%**	26.3
Labiodental	1.5%	17.8%	30.3%	40.7%	**59.3%**	23.7
Spread glottis	0.4%	20.3%	47.1%	34.6%	**65.4%**	18.8
Retracted tongue root	2.1%	16.7%	60.0%	45.8%	54.2%	6.0
Lateral	4.3%	18.6%	30.0%	47.4%	52.6%	5.3
Non-spirantized stops (– voiced)	1.2%	22.7%	45.9%	46.4%	53.6%	3.3
Geminate	8.6%	9.0%	56.8%	49.6%	50.4%	0.13
Strident	8.0%	10.5%	33.5%	**53.8%**	46.2%	12.3
Coronal	37.1%	16.4%	29.1%	**51.9%**	48.1%	22.3
Trill	5.0%	16.3%	30.7%	**55.7%**	44.3%	26.0
Spirantized stops (– voiced)	6.7%	16.9%	20.7%	**70.1%**	29.9%	458.1

5.4 Phonological Network Analysis

We sought to understand the differences of the models based on the phonological similarity of their predicted vocabularies. Specifically, we first tokenize the vocabularies of the speech models' unique lexical tokens from their predicted output, as well as the vocabularies of the gold data as encoded in both Latin and the transliterated Tifinagh. We model nodes as surface-form tokens as they appear in their respective texts; we do not lemmatize outputs to study morphological effects on the phonological network as conducted by Shoemark et al. [55] as we are not aware of any available Kabyle lemmatizers. To construct a phonological network, we then assign an edge to any pair of nodes that are one edit away from each other (Fig. 1). That is, for any pair of tokens for which a single change, addition, or subtraction could cause both tokens to be the same token, an edge is formed. For example, for a given vocabulary set, "afət" and "aqət" are linked by an undirected edge, just as "afət" and "fət" are likewise assigned an edge. However, "aqət" and "fət" are not assigned an edge since they differ by an edit distance that is greater than one.

We analyze each gold corpus and speech model's inferred vocabulary as a self-contained phonological network and follow [53] and [55] in reporting common network statistics to characterize the properties of the graph. For each vocabulary network, we report the average degree, degree assortativity coefficient, error assortativity coefficient, and average shortest path length. We control for vocabulary size by computing the average statistics of 200 randomly sampled networks of 6000 nodes. We also obtain size-controlled modularity statistics for each network by (1) obtaining 3 randomly sampled networks of 4000 nodes for each gold and model phonological network, (2) conducting the Clauset–Newman–Moore modularity maximization algorithm to split and bin nodes into communities, and (3) computing the average modularity statistic given these communities. All statistics were obtained using the Python `networkx` package, v.2.6.3 [82] (Table 8).

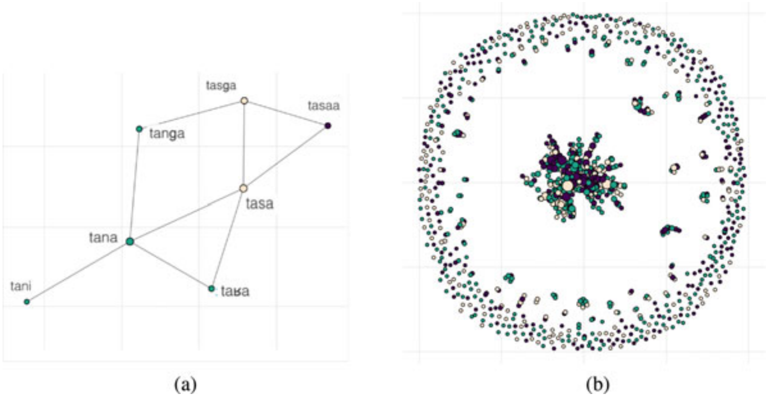


Fig. 1 Visualizations of phonological network structures on combined gold and ASR model vocabularies. Rendered with the Python package `bokeh`, v. 2.4.2 [84]. (**a**) Example module within a phonological network of Latin ASR gold and model output tokens. (**b**) Subgraph of the unionized network of Tifinagh-encoded gold and model vocabularies

Table 8 Descriptive statistics across gold vocabulary and model vocabulary phonological networks. **Bold** statistics are reported from averages across equally sized, randomly sampled subgraphs over multiple trials as reported in Sect. 5.4. **Group** denotes the vocabulary set analyzed, **Orthography** denotes the encoded orthography, **Size** denotes the length of the vocabulary, **% Giant Comp.** denotes the percentage of nodes in the largest component of the graph, **Avg. Degree** is the average degree of all nodes in the graph, **ASPL** is the average shortest path length of the giant component, **DAG** stands for the degree assortativity coefficient, **Err. AC** is the attribute assortativity coefficient of the binary feature of whether the node was outside of the gold vocabulary, and **Mod.** is the modularity of from Clauset–Newman–Moore community groupings. $^+$ indicates a transliterated vocabulary

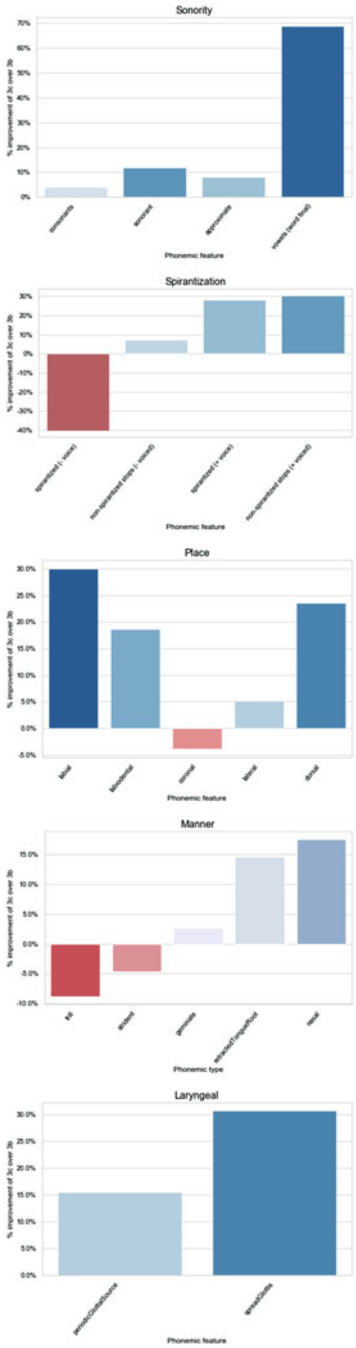
Group	Orthography	Size	% Giant comp.	Avg. deg.	ASPL	DAC	Err. AC	Mod.
Gold	Latin	12,860	48.9%	1.9 (**0.9**)	10.0 (**5.9**)	0.66 (**0.59**)	–	(**0.96**)
Gold	Tifinagh$^+$	9320	84.1%	8.4 (**5.4**)	5.5 (**5.8**)	0.54 (**0.53**)	–	(**0.66**)
3a	Latin	13,985	70.8%	6.4 (**2.7**)	6.5 (**4.3**)	0.56 (**0.54**)	0.11	(**0.79**)
3b	Tifinagh$^+$	8481	97.1%	22.6 (**16.0**)	3.9 (**4.1**)	0.40 (**0.40**)	0.10	(**0.53**)
3c	Tifinagh$^+$	7396	95.3%	19.8 (**16.0**)	4.1 (**4.2**)	0.45 (**0.45**)	0.16	(**0.54**)

6 Discussion

Performance when training on fully featured inputs (3c) to decode word-final vowels improves when compared 3b in which intra-word vowels are hidden from the model. The results suggest that sonorous and vocalized phonemes benefit more from model training on the voweled text. When only one model between 3b and 3c is correct, we see that "approximate," "sonorant," and "period glottal" phonemes exhibit comparatively high disagreement, surpassed only by the phonemes with positive "lateral" and "syllabic" features. The model may share information across these features, and in particular, voicing. All of these features record higher recognition rates in the case of 3c. While the difference in error rates for sonorous and voiced consonants between 3b and 3c does not exactly trend according to the sonority hierarchy [83], the number of disagreements between the models does follow this trend. These findings suggest that the model in 3c is leveraging correlates of sonority for phoneme recognition (Fig. 2).

A surprising finding was an improved ability of the 3b model in classifying non-tense/non-geminated phonemes modeled to be spirantized. This is interesting in that spirantized consonants are often homographic with non-spirantized consonants, so we are able to understand the variability of each model's recognition for a homograph corresponding to multiple sounds. The fact that non-spirantized consonants were better recognized by the Latin-trained 3c suggests that it is spirantization, not occlusivity, that is correlative with 3c's decreased performance in recognition of such phonemes. The reason for this disparity is unclear and may deserve additional investigation. It is notable that all contexts in which stops are modeled to be non-spirantized in our confusion analysis follow consonants. Model 3c, therefore, may better be able to recognize a non-spirantized consonants since its input would otherwise often include a vowel between the characters in question. However, the

Fig. 2 Comparison of the relative error difference between 3b and 3c



magnitude of the advantage of 3b over 3c in recognizing unvoiced spirantized stops is highly significant, especially in light of the fact that both voiced and unvoiced *non*-spirantized stops were more likely to be recognized by 3c when the two models disagreed (Table 7).

The models exhibit different rates of correctly detecting coronal and dorsal consonants. We hypothesize that this difference is a function of heterogeneous distributions in the context of vowels and geminate consonants. Further inspection of the data may also uncover imbalanced distributions between dorsal and coronal consonants with respect to word-internal vowels that are omitted in the consonantal orthography tested in this work. The improvement in the "spread glottis" feature between 3b and 3c is notable, though it is difficult to generalize given the low prevalence of graphemes representing phonemes possessing this feature. The other major orthographic difference of the Latin text compared to Tifinagh is that of marked gemination by means of digraphs. However, our results do not suggest significant differences in the models' abilities to correctly recognize these phonemes. The portion of alignments in which both models failed exhibits a wide range. Graphemes denoting the "retracted tongue root" feature were least likely to be correctly aligned. This feature, however, comprises a relatively low portion of the total number of alignments, and the models might simply not have enough instances to be able to detect the difference of this feature well. The observations we present may not hold for languages that observe some level of intra-word vowel denotation, for example, Arabic and other languages whose consonantal writing systems attest *matres lectionis* characters that present medially. To the authors' knowledge, there are no consonantal writing systems in widespread use that do not employ medial *matres lectionis* in the same way as consonantal Tifinagh. Nevertheless, the results characterize effects that may generalize to non-voweled orthographies as input to a non-Semitic language.

Our phonological network analysis reveals a stark contrast between the average degrees of the Latin and Tifinagh groups. As the phoneme vocabulary size is larger, the hyperlexica [60] of the Latin vocabulary is larger, and this effect outweighs the fact that the size of the Latin networks is larger to contribute to a high average degree. The average degrees of the ASR model output networks is greater by roughly a factor of 3 with respect to the gold networks. We believe this reflects the consolidation of choices elected by the models toward gold tokens, causing dense, closed structures to emanate from the gold signal. This interpretation is supported by a comparatively larger portion of the tokens in the ASR models' networks membership in the giant component of the graph. We note that the ASPL of the speech models' output is all roughly the same, whether encoded in Tifinagh or Latin outputs. However, it is generally shorter than those of the gold vocabularies', which structurally reflects a consolidation and narrowing in the choices to which the models converge as viable output emissions. We observe a higher average modularity statistic of Latin networks compared to that of the Tifinagh networks, reflecting the greater dispersion of highly connected modules in the network with a larger possible emission set. We find that the error assortativity coefficient trends with the models' error rates, which may reflect a tendency of erroneous tokens predicted by the higher performing ASR models to be more similar to each other.

7 Future Work

Our study experiments with the DeepSpeech architecture using a single set of hyperparameters for a single data set and language. Future work can investigate the interactions of model architectures, hyperparameters, data scales, G2P mappings, and statistics of orthographic informativeness on S2T performance. Additionally, future work could study the incidence of particular features of phonological features in modular communities in a phonological network context. An interesting direction would be to explore how other features specific to ASR modeling goals, such as a token's character edit distance from nearest neighbors, classification status as erroneous or licit, and its frequency in the gold corpora, vary with respect to specific network structures. We would also like to understand network statistics across different epoch checkpoints to observe how the network connectivity changes during the training process of the neural model.

8 Conclusion

Our study is the first to document S2T performance on Tifinagh inputs and shows that the choice of orthography may be consequential for S2T systems trained on graphemes. We amplify findings of prior studies focused on Semitic languages by showing that a Berber S2T model intended to output unvowelled graphemes benefits from training on fully featured inputs. Our research suggests that ensuring data inputs are fully featured would improve ASR model quality for languages that conventionally use consonantal orthographies, like Syriac, Hebrew, Persian, and Arabic vernaculars. Using phonological networks, we have also introduced a new way to analyze the similarities between ASR model outputs trained on different orthographies with respect to their respective gold vocabularies.

References

1. Turki, H., Adel, E., Daouda, T., Regragui, N.: A conventional orthography for maghrebi Arabic. In: Proceedings of the International Conference on Language Resources And Evaluation (LREC), Portoroz, Slovenia (2016)
2. Zitouni, I.: Natural Language Processing of Semitic Languages. Springer, Berlin (2014)
3. Jaffe, A.: Introduction: non-standard orthography and non-standard speech. J. Socioling. **4**, 497–513 (2000)

4. Cooper, E.: Text-to-Speech Synthesis Using Found Data for Low-Resource Languages. Columbia University (2019)
5. Davel, M., Barnard, E., Heerden, C., Hartmann, W., Karakos, D., Schwartz, R., Tsakalidis, S.: Exploring minimal pronunciation modeling for low resource languages. In: Sixteenth Annual Conference Of The International Speech Communication Association (2015)
6. Belinkov, Y., Ali, A., Glass, J.: Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition (2019). Preprint ArXiv:1907.04224
7. Yu, X., Vu, N., Kuhn, J.: Ensemble self-training for low-resource languages: grapheme-to-phoneme conversion and morphological inflection. In: Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 70–78 (2020)
8. Besacier, L., Barnard, E., Karpov, A., Schultz, T.: Automatic speech recognition for under-resourced languages: a survey. Speech Commun. **56**, 85–100 (2014)
9. Hu, K., Bruguier, A., Sainath, T., Prabhavalkar, R., Pundak, G.: Phoneme-based contextualization for cross-lingual speech recognition in end-to-end models (2019). Preprint ArXiv:1906.09292
10. Kubo, Y., Bacchiani, M.: Joint phoneme-grapheme model for end-to-end speech recognition. In: ICASSP 2020-2020 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 6119-6123 (2020)
11. Chen, Z., Jain, M., Wang, Y., Seltzer, M., Fuegen, C.: Joint grapheme and phoneme embeddings for contextual end-to-end ASR. In: INTERSPEECH, pp. 3490–3494 (2019)
12. Rao, K., Peng, F., Sak, H., Beaufays, F.: Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4225–4229 (2015)
13. Jyothi, P., Hasegawa-Johnson, M.: Low-resource grapheme-to-phoneme conversion using recurrent neural networks. In: 2017 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 5030–5034 (2017)
14. Arora, A., Gessler, L., Schneider, N.: Supervised Grapheme-to-Phoneme Conversion of Orthographic Schwas in Hindi and Punjabi (2020). Preprint ArXiv:2004.10353
15. Abbas, M., Asif, D.: Punjabi to ISO 15919 and Roman transliteration with phonetic rectification. In: ACM Transactions On Asian And Low-Resource Language Information Processing (TALLIP), vol. 19, pp. 1–20 (2020)
16. Hasegawa-Johnson, M., Goudeseune, C., Levow, G.: Fast transcription of speech in low-resource languages (2019). Preprint ArXiv:1909.07285
17. Yu, X., Vu, N., Kuhn, J.: Ensemble self-training for low-resource languages: Grapheme-to-phoneme conversion and morphological inflection. In: Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 70–78 (2020). https://www.aclweb.org/anthology/2020.sigmorphon-1.5
18. Deri, A., Knight, K.: Grapheme-to-phoneme models for (almost) any language. In: Proceedings of the 54th Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers), pp. 399-408 (2016)
19. Le, D., Zhang, X., Zheng, W., Fügen, C., Zweig, G., Seltzer, M.: From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition. In: 2019 IEEE Automatic Speech Recognition And Understanding Workshop (ASRU), pp. 457–464 (2019)
20. Krug, A., Knaebel, R., Stober, S.: Neuron activation profiles for interpreting convolutional speech recognition models. In: NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language (IRASL) (2018)
21. Chrupała, G., Higy, B., Alishahi, A.: Analyzing analytical methods: The case of phonology in neural models of spoken language (2020). Preprint ArXiv:2004.07070
22. Alhanai, T.: Lexical and language modeling of diacritics and morphemes in Arabic automatic speech recognition. Massachusetts Institute of Technology (2014)
23. Alshayeji, M., Sultan, S., et al., Diacritics effect on arabic speech recognition. Arab. J. Sci. Eng. **44**, 9043–9056 (2019)

24. Al-Anzi, F., AbuZeina, D.: The effect of diacritization on Arabic speech recogntion. In: 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1–5 (2017)
25. Daniels, P., Share, D.: Writing system variation and its consequences for reading and dyslexia. Sci. Stud. Read. **22**, 101–116 (2018)
26. Rafat, Y., Whitford, V., Joanisse, M., Mohaghegh, M., Swiderski, N., Cornwell, S., Valdivia, C., Fakoornia, N., Hafez, R., Nasrollahzadeh, P., et al.: First language orthography influences second language speech during reading: Evidence from highly proficient Korean-English bilinguals. In: Proceedings of the International Symposium on Monolingual and Bilingual Speech, pp. 100–107 (2019)
27. Law, J., De Vos, A., Vanderauwera, J., Wouters, J., Ghesquière, P., Vandermosten, M.: Grapheme-phoneme learning in an unknown orthography: A study in typical reading and dyslexic children. Front. Psychol. **9**, 1393 (2018)
28. Maroun, L., Ibrahim, R., Eviatar, Z.: Visual and orthographic processing in Arabic word recognition among dyslexic and typical readers. Writing Syst. Res., **11**(2), 142–158 (2019)
29. Eyben, F., Wöllmer, M., Schuller, B., Graves, A.: From speech to letters-using a novel neural network architecture for grapheme based ASR. In: 2009 IEEE Workshop On Automatic Speech Recognition & Understanding, pp. 376-380 (2009)
30. Wang, Y., Chen, X., Gales, M., Ragni, A., Wong, J.: Phonetic and graphemic systems for multi-genre broadcast transcription. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5899–5903 (2018)
31. Rao, K., Sak, H.: Multi-accent speech recognition with hierarchical grapheme based models. In: 2017 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 4815–4819 (2017)
32. Li, B., Zhang, Y., Sainath, T., Wu, Y., Chan, W.: Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5621–5625 (2019)
33. Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., Huang, H., Tjandra, A., Zhang, X., Zhang, F., et al.: Others transformer-based acoustic modeling for hybrid speech recognition. In: ICASSP 2020-2020 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP), pp. 6874–6878 (2020)
34. Schone, P.: Low-resource autodiacritization of abjads for speech keyword search. In: Ninth International Conference on Spoken Language Processing (2006)
35. Ananthakrishnan, S., Narayanan, S., Bangalore, S.: Automatic diacritization of Arabic transcripts for automatic speech recognition. In: Proceedings of the 4th International Conference on Natural Language Processing, pp. 47–54 (2005)
36. Alqahtani, S., Diab, M.: Investigating input and output units in diacritic restoration. In: 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 811–817 (2019)
37. Alqahtani, S., Mishra, A., Diab, M.: Efficient convolutional neural networks for diacritic restoration (2019). Preprint ArXiv:1912.06900
38. Darwish, K., Abdelali, A., Mubarak, H., Eldesouki, M.: Arabic diacritic recovery using a feature-rich biLSTM model (2020). Preprint ArXiv:2002.01207
39. Maroun, M., Hanley, J.: Diacritics improve comprehension of the Arabic script by providing access to the meanings of heterophonic homographs. Reading Writing **30**, 319–335 (2017)
40. Afify, M., Nguyen, L., Xiang, B., Abdou, S., Makhoul, J.: Recent progress in Arabic broadcast news transcription at BBN. INTERSPEECH. **5**, 1637–1640 (2005)
41. Alsharhan, E., Ramsay, A.: Improved Arabic speech recognition system through the automatic generation of fine-grained phonetic transcriptions. Inf. Process. Manag. **56**, 343–353 (2019)
42. Emond, J., Ramabhadran, B., Roark, B., Moreno, P., Ma, M.: Transliteration based approaches to improve code-switched speech recognition performance. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 448–455 (2018)
43. Le, N., Sadat, F.: Low-resource machine transliteration using recurrent neural networks of asian languages. In: Proceedings of the seventh Named Entities Workshop, pp. 95–100 (2018)

44. Cho, W., Kim, S., Kim, N.: Towards an efficient code-mixed grapheme-to-phoneme conversion in an agglutinative language: A case study on to-Korean Transliteration. In: Proceedings of the The 4th Workshop on Computational Approaches to Code Switching, pp. 65–70 (2020)
45. Ahmadi, S.: A rule-based Kurdish text transliteration system. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **18**, 1–8 (2019)
46. Abbas, M., Asif, D.: Punjabi to ISO 15919 and Roman transliteration with phonetic rectification. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **19** (2020). https://doi.org/10.1145/3359991
47. Sadouk, L., Gadi, T., Essoufi, E.: Handwritten tifinagh character recognition using deep learning architectures. In: Proceedings of the 1st International Conference on Internet of Things and Machine Learning, pp. 1–11 (2017)
48. Benaddy, M., El Meslouhi, O., Es-saady, Y., Kardouchi, M.: Handwritten tifinagh characters recognition using deep convolutional neural networks. Sensing Imaging **20**, 9 (2019)
49. Lyes, D., Leila, F., Hocine, T.: Building a pronunciation dictionary for the Kabyle language. In: International Conference on Speech and Computer, pp. 309–316 (2019)
50. Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F., Weber, G.: Common voice: A massively-multilingual speech corpus (2019). Preprint ArXiv:1912.06670
51. Zealouk, O., Hamidi, M., Satori, H., Satori, K.: Amazigh digits speech recognition system under noise car environment. In: Embedded Systems And Artificial Intelligence, pp. 421–428 (2020)
52. Luce, P., Pisoni, D.: Recognizing spoken words: the neighborhood activation model. Ear Hearing **19**, 1 (1998)
53. Vitevitch, M.S: What can graph theory tell us about word learning and lexical retrieval? J. Speech Lang. Hear. Res. **51**(2), 408–422 (2008)
54. Arbesman, S., Strogatz, S., Vitevitch, M.: The structure of phonological networks across multiple languages. Int. J. Bifurcat. Chaos **20**, 679–685 (2010)
55. Shoemark, P., Goldwater, S., Kirby, J., Sarkar, R.: Towards robust cross-linguistic comparisons of phonological networks. In: Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 110–120 (2016)
56. Siew, C.: Community structure in the phonological network. Front. Psychol. **4**, 553 (2013)
57. Siew, C., Vitevitch, M.: An investigation of network growth principles in the phonological language network. J. Exper. Psychol. General **149**, 2376 (2020)
58. Siew, C., Vitevitch, M.: The phonographic language network: using network science to investigate the phonological and orthographic similarity structure of language. J. Exper. Psychol. General. **148**, 475 (2019)
59. Neergaard, K., Luo, J., Huang, C.: Phonological network fluency identifies phonological restructuring through mental search. Sci. Rep. **9**, 1–12 (2019)
60. Turnbull, R.: Graph-theoretic properties of the class of phonological neighbourhood networks. In: Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pp. 233–240 (2021)
61. Souag, L.: Kabyle in Arabic script: A history without standardisation. In: Creating Standards, pp. 273. De Gruyter, Boston (2019)
62. Blanco, J.: Tifinagh & the IRCAM: Explorations in cursiveness and bicameralism in the tifinagh script. Unpublished Dissertation, University of Reading (2014)
63. Louali, N., Maddieson, I.: Phonological contrast and phonetic realization: The case of Berber stops. In: Proceedings of the 14th International Congress Of Phonetic Sciences, pp. 603–606 (1999)
64. Elias, A.: Kabyle "Double" Consonants: Long or Strong? UC Berkeley (2020). Retrieved from https://escholarship.org/uc/item/176203d
65. Elghamis, R.: Le tifinagh au Niger contemporain: Étude sur lécriture indigène des Touaregs. Unpublished PhD Thesis, Leiden: Universiteit Leiden (2011)
66. Savage, A.: Writing Tuareg–the three script options. Int. J. Sociol. Lang. **2008**, 5–13 (2008)

67. Posegay, N.: Connecting the dots: The shared phonological tradition in Syriac, Arabic, and Hebrew Vocalisation. In: Studies In Semitic Vocalisation And Reading Traditions, p. 191–226 (2020)
68. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: Scaling up end-to-end speech recognition (2014). Preprint ArXiv:1412.5567
69. Heafield, K., Pouzyrevsky, I., Clark, J., Koehn, P.: Scalable modified Kneser-Ney language model estimation. In: Proceedings of the 51st Annual Meeting Of The Association For Computational Linguistics (Volume 2: Short Papers), pp. 690-696 (2013). https://www.aclweb.org/anthology/P13-2121
70. Pue, A.: Graph transliterator: a graph-based transliteration tool. In: J. Open Source Softw. **4**(44), 1717 (2019). https://doi.org/10.21105/joss.01717
71. McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M.: Montreal forced aligner: trainable text-speech alignment using Kaldi. Interspeech **2017**, 498–502 (2017)
72. Tilmankamp, L.: DSAlign. GitHub Repository (2019). https://github.com/mozilla/DSAlign
73. List, J.: Sequence comparison in historical linguistics. Düsseldorf University Press (2014)
74. Marjou, X.: OTEANN: Estimating the transparency of orthographies with an artificial neural network. In: Proceedings of the Third Workshop On Computational Typology And Multilingual NLP, pp. 1–9 (2021). https://aclanthology.org/2021.sigtyp-1.1
75. List, J., Greenhill, S., Tresoldi, T., Forkel, R.: LingPy. A Python library for quantitative tasks in historical linguistics. Max Planck Institute for the Science of Human History (2019). http://lingpy.org
76. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biol. Evolut. **4**, 406–425 (1987)
77. Kong, X., Choi, J., Shattuck-Hufnagel, S.: Evaluating automatic speech recognition systems in comparison with human perception results using distinctive feature measures. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5810–5814 (2017)
78. Alishahi, A., Barking, M., Chrupała, G.: Encoding of phonology in a recurrent neural model of grounded speech (2017). Preprint ArXiv:1706.03815
79. Moran, S., McCloy, D. (Eds.): PHOIBLE 2.0. Max Planck Institute for the Science of Human History (2019). https://phoible.org/
80. Chaker, S.: Propositions pour la notation usuelle a base latine du Berbère. In: INALCO-CRB, p. e0245263 (1996)
81. Edwards, A.: Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. Psychometrika **13**, 185–187 (1948)
82. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using NetworkX. Los Alamos National Lab. (LANL), Los Alamos, NM (2008). https://github.com/networkx/networkx/releases/tag/networkx-2.6.3
83. Ladefoged, P., Johnson, K.: A Course in Phonetics. Nelson Education, Toronto (2014)
84. Bokeh Development Team: Bokeh: Python library for interactive visualization. (2022) https://bokeh.org/