Preserving Comorian Linguistic Heritage: Bidirectional Transliteration Between the Latin Alphabet and the Kamar-Eddine System

Naira Abdou Mohamed[1,2], **Abdessalam Bahafid**[1,2], **Zakarya Erraji**[1], **Anass Allak**[1], **Naoufal Mohamed Soibira**[3], **Imade Benelallam**[1,2]
[1] INSEA, Rabat, Morocco [2] ToumAI Analytics, Rabat, Morocco
[3] Sciences Po Grenoble, Grenoble, France
{nabdoumohamed,i.benelallam,a.bahafid,zerraji,aallak}@insea.ac.ma
{naira,imade,abahafid}@toum.ai

Abstract

The Comoros Islands, rich in linguistic diversity, are home to dialects derived from Swahili and influenced by Arabic. Historically, the Kamar-Eddine system, based on the Arabic alphabet, was one of the first writing systems used for Comorian. However, it has gradually been replaced by the Latin alphabet, even though numerous archival texts are written in this system, and older speakers continue to use it, highlighting its cultural and historical significance. In this article, we present Shialifube, a bidirectional transliteration tool between Latin and Arabic scripts, designed in accordance with the rules of the Kamar-Eddine system. To evaluate its performance, we applied a round-trip transliteration technique, achieving a word error rate of 14.84% and a character error rate of 9.56%. These results demonstrate the reliability of our system for complex tasks. Furthermore, Shialifube was tested in a practical case related to speech recognition, showcasing its potential in Natural Language Processing. This project serves as a bridge between tradition and modernity, contributing to the preservation of Comorian linguistic heritage while paving the way for better integration of local dialects into advanced technologies.

1 Introduction

At the crossroads of Africa, Europe, the Middle East, and Southeast Asia (Abeid et al., 2024; Allibert, 2015), the Comoros stand out for their rich cultural heritage, a diversity particularly evident in local dialects that share remarkable similarities with several foreign languages. While these dialects belong to the Bantu family due to their closer affinity with Swahili (Ahmed Chamanga, 2022) and the Sabaki language group (Serva and Pasquini, 2021), they also exhibit similarities with Arabic. This is partly why, like Swahili with the Ajami script (Mugane, 2017), one of the earliest writing systems for Comorian dialects is based on the Arabic script (Lafon, 2007).

Known as the Kamar-Eddine system, this writing system was introduced in the 1960s by linguist Sheikh Ahmed Kamar-Eddine. Although the Latin alphabet is now predominantly used to write Comorian, a minority, primarily older individuals, are only proficient in the Arabic script. Furthermore, many manuscripts are written in this script, emphasizing its historical and cultural significance.

Having a solution to process this system could serve three major purposes: (a) Democratizing access to Natural Language Processing (NLP) technologies, making Comorian dialects accessible to a broader audience, especially those without access to modern digital tools; (b) Preserving and promoting the multicultural richness of the archipelago, highlighting the Kamar-Eddine system as a fundamental element of Comorian linguistic and cultural heritage; (c) Making Comorian national archives accessible to all, facilitating their digitization and long-term preservation while paving the way for new research and applications in NLP.

This work aims to initiate NLP research for this writing system, with the hope of contributing to the preservation of Comorian intangible heritage. More concretely, our main contributions can be summarized as follows:

- **Complementary Study**: This work builds on Michel Lafon's article (Lafon, 2007), which, to the best of our knowledge, is the only study conducted on the Kamar-Eddine system.
- **Foundational Exploration**: We contribute to the introduction of NLP not only for this writing system but also for the processing of Comorian, a language still underrepresented in this field.
- **Shared Innovation**: We make the results of this work accessible by sharing the developed code and models, enabling the community to benefit from our progress.

Proceedings of LaTeCH-CLfL 2025, pages 11–18

2 About ShiKomori

Comorian, or ShiKomori, consists of four dialects, each spoken on a specific island: ShiNgazidja, ShiMwali, ShiNdzuani, and ShiMaore. While ideally, each dialect would be treated individually, this work addresses Comorian as a whole, without distinguishing between its dialectal variations. Two main reasons justify this choice:

- **High Similarities Between Dialects**: The dialects are very closely related (Ahmed Chamanga, 2022). Consequently, a speaker from one island can understand a dialect spoken on another island with little difficulty due to the largely shared lexicon across these variants. This strong similarity facilitates the development of NLP solutions that can generalize across all dialects.

- **Data Scarcity**: It is challenging to find dialect-specific corpora due to the limited research conducted in this field. Furthermore, speakers often prefer writing in French rather than using their local dialects, further restricting access to data.

The high similarities among these dialects, combined with the significant lack of data, make it more practical to treat them as a single language. Attempting to develop solutions for each dialect individually would require working with small, separate corpora, which might not suffice for training effective models. Instead, this approach leverages data-rich dialects to improve performance on those with fewer resources.

This strategy aligns with the findings of Lin et al. (Lin et al., 2019), which explored multilingual transfer learning as a means to improve low-resource language representation by leveraging a well-resourced language with significant similarities. Additionally, the system introduced by Kamar-Eddine considers Comorian as a unified language, with no specific rules tailored to individual dialects.

3 Related Work

Comorian is a language that has been very little studied in the field of NLP. While some previous works have provided solutions addressing it for various use cases (Abdourahamane et al., 2016; Naira et al., 2024), to the best of our knowledge, there is no computational linguistics research that deals with the language in its Arabic script.

Beyond our desire to preserve this intangible heritage, there is a motivation arising from observations made in previous works, such as those found in (Micallef et al., 2023). The latter describes experiments conducted on Maltese in which a curious observation was made: in several tasks (named entity recognition, sentiment analysis, etc.), transliteration into Arabic characters significantly improved the performance of models. The reason for this is that although Maltese is written in Latin characters and contains Italian loanwords, it remains a Semitic language closely related to Arabic. The proximity of Comorian to Arabic thus justifies the exploration of whether existing NLP solutions could be enhanced by adopting a similar approach.

In the absence of work specifically addressing Comorian written in Arabic script, we present in Table 1 a few notable studies that have dealt with the topic of transliteration in general, and particularly for African languages.

4 The Kamar-Eddine System

The standardization of Comorian writing became a priority in the years following the independence of the Comoros archipelago (Chamanga and Gueunier, 1977). While the idea of establishing specific rules for each dialect was quickly abandoned, the debate over whether to use the Latin or Arabic alphabet sparked intense discussions. On one hand, only a small minority of the population, educated in French, the colonial language, knew how to read the Latin alphabet and thus advocated for its use. On the other hand, the majority, having received an education primarily in Quranic schools, were proficient in reading the Arabic alphabet. With public opinion in favor of the latter, Arabic was quickly adopted for the translation of official documents.

However, it is important to note that, despite the widespread use of this alphabet, there were no fixed rules governing its application. It was precisely in this context that Ahmed Kamar-Eddine conceived the idea of standardizing this writing system. He began this project by publishing chronicles in his journal Mwando (see the manuscript of the first edition in Figure 1).

Title	Year	Description
Moroccan Arabizi-to-Arabic conversion using rule-based transliteration and weighted Levenshtein algorithm (Hajbi et al., 2024)	2024	It is a system of transliteration from Arabizi (Moroccan dialectal Arabic written in Latin characters) to Arabic characters. The method used uses the Levenshtein distance.
Exploring the Impact of Transliteration on NLP Performance: Treating Maltese as an Arabic Dialect (Micallef et al., 2023)	2023	Improving the state of the art TAL on several tasks by processing Maltese written in Arabic characters.
A Unified Model for Arabizi Detection and Transliteration using Sequence-to-Sequence Models (Shazal et al., 2020)	2020	Pipeline for detecting Arabizi in a text with code switches (Arabic mixed with other languages, all written in Latin characters) and transliteration into Arabic characters.
Arabizi Chat Alphabet Transliteration to Algerian Dialect (Klouche and Benslimane, 2020)	2020	Transliteration into Arabic characters of comments on the Algerian telephone operator Ooredoo in order to train a sentiment analysis model.

Table 1: Previous work on transliteration into Arabic scripts.

Figure 1: The Mwando Chronicles Manuscript: A historical document showcasing the first application of the Kamar-Eddine system, marking its inaugural use for formalizing the transcription of the Comorian language in Arabic script. The manuscript also describes the writing rules of the system, notably the introduction of long vowels.

Vowels	Transcription	Meaning
na	نَجم (najm)	star
ni	نِظام (nizām)	system
nu	نُور (nūr)	light

Table 2: Diacritics in Arabic writing.

4.1 The first adaptations in Arabic scripts

The Arabic alphabet has the particularity of being an abjad[1]. There are three vowels in Arabic, /a/, /i/, and /u/, represented respectively by the diacritics *fatha*, *kasra*, and *dhamma* (see examples in Table 2). The absence of a vowel is represented by a *sukun*, as in the word بِنْت (bint), which means "girl".

This particularity of Arabic, having only three vowels, poses a challenge when adapting certain languages to this script. This is precisely the case for Wolof, which contains nine vowels (Currah, 2015), Swahili (Raia, 2021), and Comorian (Lafon, 2007). For the latter, there are also additional consonants that do not exist in the Arabic alphabet. To address these specificities, certain adaptations were introduced in the early attempts. Among them were:

- **Introduction of additional characters**: Bor-

[1]A writing system in which characters represent consonants, and vowels are either implied or marked with optional diacritics. Scripts like Arabic and Hebrew are examples of abjads. Unlike full alphabets, abjads do not assign separate letters to vowel sounds.

rowings were made from Persian for representing sounds such as /v/ (ڤ), /g/ (غ), and /p/ (پ). However, ambiguities persisted, as the sound /pv/ was sometimes transcribed as ڤ (like /v/) or ف (like /f/).

- **Representation of vowels**: Comorian, with its five vowels /a/, /e/, /i/, /o/, and /u/, required measures to address the absence of /o/ and /e/ in Arabic. These vowels were marked by either using diacritics or resorting to long vowels, و for /o/ and ي for /e/. Yet, this also led to ambiguities in some cases, as terms like "mezi" (month) and "mizi" (roots) were written the same way (مِزِ or ميزي when using long vowels).

4.2 Kamar-Eddine's Original Innovations

To address the ambiguities observed in previous adaptation attempts, one of the solutions proposed by Kamar-Eddine was to abandon diacritics in favor of long vowels. The vowels /a/, /i/, and /u/ retain their original forms, while /e/ and /o/ are represented respectively by ۑ and ه. This categorically resolves certain cases of confusion, such as the last example discussed in the previous subsection. With this correction, the term mezi becomes مـهـزي, and mizi becomes ميزي.

Until then, there had been no clear representation of affricates, which are nonetheless frequent in Comorian. Kamar-Eddine proposed using the *shadda* to accentuate these consonants (see Table 3). Finally, we summarize all the identified rules in Table 4.

5 Methodology

Today, unless it has escaped our notice, there is no Comorian database written in Arabic script. To evaluate the effectiveness of our system, we are therefore compelled to rely solely on Latin-script texts[2] as references. Comprising 17,000 entries (sentences, words, and expressions), the dataset is first used to transliterate into Arabic by applying the rules based on the constructed dictionary. We then perform reverse transliteration to recover the original text. To assess the quality of our system, we use Word Error Rate (WER) and Character Error Rate (CER) as metrics.

The Figure 2 summarizes the pipeline through which an input text passes during the inference of our tool. First, we use computation rules to detect the type of script used, whether it is Arabic or Latin. This determines which dictionary to load (arabic_latin or latin_arabic). Then, once the script type and the corresponding dictionary are identified, we perform the transliteration followed by a reverse transliteration to attempt to regenerate the original text. This allows us to calculate round-trip transliteration scores to measure the confidence of the transliteration. Thus, two elements are returned as output: the transliteration and its confidence score.

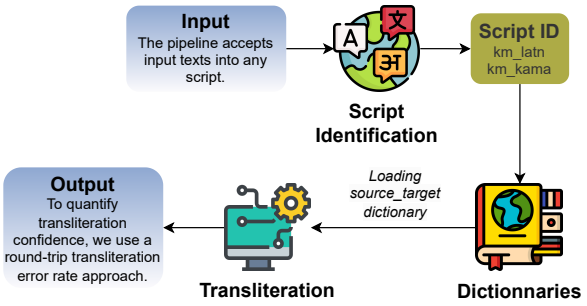

Figure 2: Global Pipeline: the system takes as input a raw text with the possibility to specify the source and target scripts. When no source is specified, a script identification model is used, and then, depending on the detected source, a dictionary is loaded. We use a round-trip transliteration error rate to measure the reliability of the transliteration.

5.1 From Latin to Arabic

The first step of this approach involves identifying the Latin digraphs present in the string and replacing them with their equivalents in Arabic script using a pre-established correspondence dictionary. This step effectively transforms specific sounds represented by two characters into a single appropriate Arabic symbol, such as the digraphs "sh" or "pv" To understand why this is important, imagine we want to transliterate the term shama (association). Failing to identify digraphs at the outset would result in treating sh as two separate letters (interpreting s as س and h as ح), which is a critical error. Instead of this reasoning, we transliterate sh into ش and then process the rest, where each remaining Latin character is converted into its Arabic equivalent according to a second correspondence dictionary for isolated characters, thereby ensuring coverage of sounds not represented by digraphs.

[2] https://huggingface.co/datasets/nairaxo/shikomori-texts

Sound	Transcription	Example	Translation
/ny/	نّ	نّاما	meat
/tr/	تّ	تّونكو	grass
/dz/	زّ	مزّو	burden

Table 3: Use of shadda to represent affricates.

Regular Alphabet						Digraphs / Affricates		
Sound	Arabic	Latin	Sound	Arabic	Latin	Sound	Arabic	Latin
/a/	ا	a	/m/	م	m	/ð/	ذ	dh
/ɓ/ or /b/	ب	b or ɓ	/n/	ن	n	/ɖ/	ڕ	dr
/t͡ʃ/	ڜ	c	/o/	ه	o	/d͡z/	ڒ	dz
/ɗ/ or /d/	د	ɗ or d	/p/	پ	p	/ʈ/	ڎ	tr
/e/	ې	e	/r/	ر	r	/ɲ/	ڼ	ny
/f/	ف	f	/s/	س	s	/ʃ/	ش	sh
/g/	غ	g	/t/	ت	t	/β/	ۋ	pv
/h/	ح	h	/u/	و	u	/θ/	ث	th
/i/	ي	i	/v/	ڤ	v	/t͡s/	ڛ	ts
/d͡ʒ/	ج	j	/w/	و	w			
/k/	ك	k	/y/	ي	y			

Table 4: Table of correspondences between sounds, Arabic script, and Latin script.

5.2 From Arabic to Latin

We perform the transliteration of a string from Arabic script to a Latin representation by applying several specific transformations. This process also involves replacing Arabic letters that need to be represented by Latin digraphs with their equivalents. Next, the algorithm handles special Arabic characters such as the symbol ه, replacing them with the appropriate Latin characters and managing specific combinations like یه to ensure phonetically accurate transliteration.

After segmenting the string into individual characters, the algorithm applies a set of specific rules to handle letters used as long vowels, such as و and ي. For instance, if و is used not as a long vowel but as the letter representing the sound /w/, it is replaced by w; otherwise, it is replaced by u. Similarly, for ي, the transliterations y and i are applied to represent the sound /y/ and the long vowel /i/, respectively. Finally, the string is reassembled to produce the final Latin-script version, adhering to the phonetic and graphical conventions of the target language.

5.3 System Evaluation

WER is a common metric used to evaluate the accuracy of an automatic speech recognition or machine translation system. It indicates the rate of errors in the transcription produced compared to a reference transcription. WER accounts for multiple types of errors, including insertions, deletions, and substitutions of words. Lower WER values indicate better performance, meaning the system has fewer errors compared to the reference. WER ranges from 0 to 100%. The formula to compute it is as follows:

$$WER = \frac{S + D + I}{N} \tag{1}$$

where:

- **S**: the number of substituted words (incorrect substitutions),
- **D**: the number of deleted words (omissions),
- **I**: the number of inserted words (incorrect additions),
- **N**: the total number of words in the reference transcription.

The same formula is used to compute the CER, which measures the substitution rate at the character level instead of the word level. While both metrics measure the performance of a system like ours,

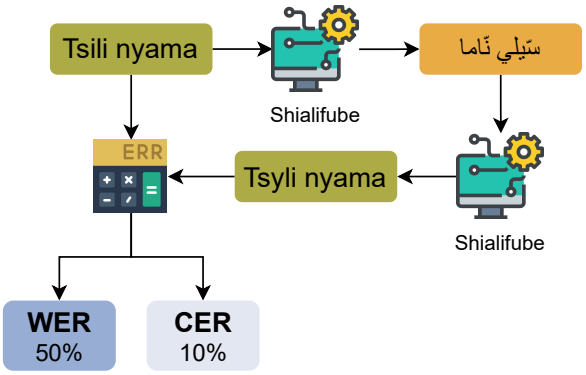

Figure 3: Example of round-trip transliteration and calculation of performance metrics.

they do not necessarily serve the same purpose. For instance, WER tends to measure orthographic divergence between two texts. Let us consider the following example: سّيلي نّاما (I ate meat). It might happen that during transliteration, this phrase is written as سّيل نّام, which is still comprehensible despite the writing error. The WER in this case is 100%, whereas the CER is relatively low at 22.2%.

Generally, to compute these metrics, labeled data is required, which is not the case for our system. To address this, we employ a technique inspired by back-translation (Kementchedjhieva and Søgaard, 2023), where we transliterate our Latin text into Arabic using our system, and then transliterate it back to Latin. We then calculate WER and CER metrics to evaluate the performance of our solution. Figure 3 illustrates an example of a back-transliteration process.

6 Experimental Results

In this section, we first present the results and performance metrics of Shialifube, along with descriptions of the various iterations adopted to improve its performance. Additionally, we conduct an experiment on a real-world use case in speech recognition: the first machine learning model ever designed for Comorian written in Arabic script.

Convinced that open-source contributions are the key to advancing the representation of low-resource languages in the field of NLP, we have made the Shialifube library[3], its code on GitHub[4], and a HuggingFace Space[5] publicly available for everyone.

[3] https://pypi.org/project/shialifube/
[4] https://github.com/nairaxo/shialifube
[5] https://huggingface.co/spaces/nairaxo/swauti

6.1 Round-trip Transliteration

The process of applying our transliteration rules was incremental, with our algorithm gradually adjusting based on the specific cases encountered. The goal was to find the most optimal approach that minimizes the evaluation metrics. Each time we adjusted our algorithm, we recalculated these metrics. Table 5 describes the different scenarios used. In total, we conducted four iterations. The final iteration yielded interesting metrics, indicating a certain reliability of our system, although we propose exploring new improvement avenues in future work.

It is important to note that while we have strived to handle all special cases, limitations may still arise during the system's use. To minimize these limitations, we plan to continue refining and updating the library. The current version is, in fact, a pre-release.

6.2 Use Case: Speech Recognition

In this section, we introduce the first speech recognition model for Comorian using the Arabic script. Our objective is twofold: first, to demonstrate the feasibility of such a model by leveraging our Kamar-Eddine transliteration system and second, to assess the effectiveness of our transliteration framework by measuring its impact on speech recognition performance. In fact, if the conversion of Comorian text into the Arabic script significantly altered the data, it would negatively affect model training, leading to degraded performance.

Regarding the choice of model architecture, we selected Whisper (Radford et al., 2022), one of the most performant speech recognition models in the state of the art. Whisper is pre-trained on a large multilingual dataset that includes Swahili and Arabic. This pre-training phase involves teaching the model to better understand each language by capturing latent parameters within the audio data. We fine-tune the model by updating its parameters for speech recognition tasks, specifying Swahili for the Latin script model and Arabic for the Arabic script model.

The results in Table 6 indicate better performance for the Latin script model compared to the Arabic script model. Two main reasons explain this discrepancy:

- **Untransformed data**: Transforming the data affects its quality. While this approach was necessary to generate data in our case, it

Experiment	Description	WER (%)	CER (%)
1	Initial iteration, without digram handling.	68.56	34.41
2	Digram handling and long vowel processing.	43.09	21.30
3	Corpus sequence standardization and corrections[a].	33.89	16.75
4	Handling additional edge cases and incorporating observations from previous iterations.	14.84	9.56

[a] The corpus used comes from various sources, and given the lack of fixed writing rules for Comorian, a standardization procedure was applied to unify the writing style and correct inconsistencies. This standardized writing facilitates the generalization of our transliteration system.

Table 5: Evaluation metrics for the round-trip transliteration approach.

does compromise performance compared to manual annotation. Manual annotation is a promising avenue for future work, not only to improve speech recognition performance but also for other NLP tasks such as sentiment analysis, named entity recognition, etc.

- **Unknown vocabulary**: The use of a pre-trained model depends on its vocabulary. While Comorian is similar to Arabic, it is not closer than Swahili. Consequently, during tokenization of the Arabic script text for model training, there are more unknown tokens for the pre-trained model compared to training with Latin script text.

Script	WER (%)	CER (%)
Latin	35.48	17.76
Arabic	37.44	21.42

Table 6: WER and CER for speech recognition models trained on Latin and Arabic script corpora. The Latin script model serves as a baseline, while the Arabic script model evaluates the effectiveness of the Kamar-Eddine transliteration system.

Finally, these results demonstrate that training a Comorian speech recognition model using the Arabic script is feasible, thanks to the effectiveness of the Kamar-Eddine transliteration system. While the Latin script model achieves slightly better performance, the Arabic script model remains competitive, highlighting the potential of our approach. Future work will focus on improving data quality through manual annotation and further optimizing the transliteration process to enhance speech recognition accuracy.

7 Conclusion

This work aimed to lay the foundation for NLP applied to the Comorian language, with a focus on transcribing this language into Arabic script using the Kamar-Eddine system. Initially, we compiled the set of writing rules for this system, which served as the basis for Shialifube, a bidirectional transliteration system for Comorian.

In the absence of parallel data to directly evaluate the performance of our solution, we adopted a round-trip transliteration approach. This involved transcribing a corpus from Latin script to Arabic script and then retranscribing it back to Latin script. This method yielded promising metrics after several iterations: a WER of 14.84% and a CER of 9.56%.

To assess the utility of this tool for practical use cases, we also conducted experiments in speech recognition. We observed encouraging performance with a WER of 37.44% for the Arabic script version, although it remained slightly lower than the Latin script model, which achieved a WER of 35.48%.

Finally, it is worth noting that this work represents a preliminary step. We plan to continue refining it as part of future contributions, hoping it will contribute to the preservation and enhancement of Comorian intangible heritage. To encourage other researchers to further this initiative, we are making the entire source code, the Shialifube library, and the trained models publicly available.

References

Moneim Abdourahamane, Christian Boitet, Valérie Bellynck, Lingxiao Wang, and Hervé Blanchon. 2016. Construction d'un corpus parallèle français-comorien en utilisant de la TA français-swahili. In *TALAf (Traitement Automatique des Langues africaines)*, Paris, France.

Said Nassor Abeid, Hamid Farhane, Majida Motrane, Fatima Ezzahra Anaibar, and Nourdin

Harich. 2024. Inference on the biological history of the comoros archipelago using the cd4 alu/str compound system. *Gene Reports*, 34:101865.

Mohamed Ahmed Chamanga. 2022. *ShiKomori, the Bantu Language of the Comoros: Status and Perspectives*, page 79–98. BRILL.

Claude Allibert. 2015. L'archipel des comores et son histoire ancienne. essai de mise en perspective des chroniques, de la tradition orale et des typologies de céramiques locales et d'importation. *Afriques*, 06.

Mohamed Ahmed Chamanga and Noël Jacques Gueunier. 1977. Recherches sur l'instrumentalisation du comorien : problèmes d'adaptation lexicale (d'après la version comorienne de la loi du 23 novembre 1974). *Cahiers d'Études africaines*, 66-67:213–239.

Galien Currah. 2015. Orthographe wolofal.

Soufiane Hajbi, Omayma Amezian, Nawfal El Moukhi, Redouan Korchiyne, and Younes Chihab. 2024. Moroccan arabizi-to-arabic conversion using rule-based transliteration and weighted levenshtein algorithm. *Scientific African*, 23:e02073.

Yova Kementchedjhieva and Anders Søgaard. 2023. Grammatical error correction through round-trip machine translation. In *Findings of the Association for Computational Linguistics: EACL 2023*, page 2208–2215. Association for Computational Linguistics.

B. Klouche and S. M. Benslimane. 2020. *Arabizi Chat Alphabet Transliteration to Algerian Dialect*, page 790–797. Springer International Publishing.

Michel Lafon. 2007. Le système Kamar-Eddine : une tentative originale d'écriture du comorien en graphie arabe. *Ya Mkobe*, 14-15:29–48.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3125–3135. Association for Computational Linguistics.

Kurt Micallef, Fadhl Eryani, Nizar Habash, Houda Bouamor, and Claudia Borg. 2023. Exploring the impact of transliteration on nlp performance: Treating maltese as an arabic dialect. In *Proceedings of the Workshop on Computation and Written Language (CAWL 2023)*, page 22–32. Association for Computational Linguistics.

John Mugane. 2017. The odyssey of ajami and the swahili people. *Islamic Africa*, 8(1–2):193–216.

Abdou Mohamed Naira, Benelallam Imade, Bahafid Abdessalam, and Erraji Zakarya. 2024. Datasets creation and empirical evaluations of cross-lingual learning on extremely low-resource languages: A focus on comorian dialects. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 140–149, St. Julians, Malta. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Annachiara Raia. 2021. One text, many forms: a comparative view of the variability of swahili manuscripts. In R.H. Samsom and C. Vierke, editors, *Manuscript Cultures*, 17, pages 65–86.

Maurizio Serva and Michele Pasquini. 2021. The sabaki languages of comoros.

Ali Shazal, Aiza Usman, and Nizar Habash. 2020. A unified model for Arabizi detection and transliteration using sequence-to-sequence models. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 167–177, Barcelona, Spain (Online). Association for Computational Linguistics.