Code-Switching and Back-Transliteration Using a Bilingual Model

Daniel Weisberg Mitelman
Efi Arazi School
of Computer Science
Reichman University
dwmitelman@gmail.com

Nachum Dershowitz
Blavatnik School
of Computer Science
Tel Aviv University
nachum@tau.ac.il

Kfir Bar
Efi Arazi School
of Computer Science
Reichman University
kfir.bar@runi.ac.il

Abstract

The challenges of automated transliteration and code-switching–detection in Judeo-Arabic texts are addressed. We introduce two novel machine-learning models, one focused on transliterating Judeo-Arabic into Arabic, and another aimed at identifying non-Arabic words, predominantly Hebrew and Aramaic. Unlike prior work, our models are based on a bilingual Arabic-Hebrew language model, providing a unique advantage in capturing shared linguistic nuances. Evaluation results show that our models outperform prior solutions for the same tasks. As a practical contribution, we present a comprehensive pipeline capable of taking Judeo-Arabic text, identifying non-Arabic words, and then transliterating the Arabic portions into Arabic script. This work not only advances the state of the art but also offers a valuable toolset for making Judeo-Arabic texts more accessible to a broader Arabic-speaking audience and more amenable to modern language tools.

1 Introduction

Judeo-Arabic is a family of ethnolects spoken and written by various Jewish communities living in Arabic-speaking countries, from geonic times (9th century) down until the late 20th century. The language is typically written in Hebrew letters, enriched with diacritic marks that relate to the underlying Arabic. However, inconsistencies in rendering Arabic words in the Hebrew alphabet increase the level of ambiguity of a given written word. Furthermore, Judeo-Arabic texts usually include non-Arabic words and phrases, such as quotations or borrowed words from Hebrew and Aramaic. On Judeo-Arabic, see, for instance, (Hary, 2018). Figure 1 is an example of an original text written in Judeo-Arabic in the eleventh century.

A wealth of Judeo-Arabic works (philosophy, Bible translation, biblical commentary, and much more) is already available on the internet. However, most speakers of Arabic are unfamiliar with the Hebrew script, let alone the way it is used to render Judeo-Arabic. Thus, our primary goal in this endeavor is to allow Arabic readers, who are unfamiliar with Hebrew, to nevertheless read and understand these texts.

A very large quantity of ancient texts written in Judeo-Arabic was found in the Cairo Geniza. This treasure trove of handwritten documents, treatises, and books—mostly fragmentary—was discovered in the late 19th century in the attic of old Cairo's Ben-Ezra Synagogue, and has profoundly impacted the fields of Jewish studies, Mediterranean and Indian history, and Semitic linguistics. This unique collection spans over a millennium, from the 9th to 19th century ce, offering invaluable insights into the daily lives, religious practices, commerce, and intellectual pursuits of the Jewish communities and their neighbors in Egypt and the Mediterranean world. Comprising letters, legal documents, religious texts, and fragments of various languages, including Hebrew, Aramaic, Arabic, and Judeo-Arabic, the Geniza illuminates the dynamic intercultural exchanges and adaptations within this diverse Jewish diaspora. Its discovery significantly expanded understanding of medieval Mediterranean society and continues to be a rich source for scholarly research, shedding light on a fascinating and variegated tapestry of human history and culture (Hoffman and Cole, 2011). Images of virtually all this material are viewable on the internet as part of the Friedberg Genizah Project.[1]

Other digital projects and libraries have made additional Judeo-Arabic texts readily accessible. The Ktiv project of the National Library of Israel links to scans of thousands of pages of medieval codices.[2] The Princeton Geniza Project provides

[1] https://fjms.genizah.org/

[2] https://www.nli.org.il/en/discover/manuscripts/hebrew-manuscripts

Figure 1: Beginning of a letter in Judeo-Arabic, found in the Cairo Geniza, from Ṭoviya ben Moshe in Jerusalem to his daughter in Cairo, 1040–1. (Cambridge University Library Or.1080 J21; courtesy the Syndics of Cambridge University Library.)

access to images and transcriptions of thousands of documents.[3] The Friedberg Judeo-Arabic Project provides digital texts for more than 100 important works.[4] Plus there are several additional resources for Judeo-Arabic available.[5]

We focus on two main tasks: (1) automatic identification of the language of morphemes (not just words) in the text, Judeo-Arabic or not (in which case it is virtually always either Hebrew or Aramaic); and (2) automatic transliteration of Judeo-Arabic into Arabic letters (of the Arabic parts only).

Code switching is the act of changing language while speaking or writing, as often done by bilinguals (Winford, 2003). In our case, with cross-language inflections (e.g. when a Hebrew word is inflected following Arabic morphological rules) in addition to the rich morphology of Arabic, code switching turns out to be nontrivial. We use a language model of both Arabic and Hebrew, written in Hebrew script (we elaborate on the model below), fine-tuned on the code-switching task.

Transliteration is the process of converting a text from one (input) script into another (target script). Transliteration differs from translation and is considerably easier, since semantics play only a small role in decipherment.

Our primary objective in this study is to develop tools that enable the automatic conversion of Judeo-Arabic texts into Arabic, thus rendering many books and texts readily accessible to Arabic readers. It could also facilitate intertextual studies like (Phillips, 2020), as well as enabling computational processing of Judeo-Arabic texts once they are converted into the Arabic script, for which numerous tools already exist. For instance, Tirosh-Becker et al. (2022) could benefit from using Arabic part-of-speech taggers upon transliterating the texts into Arabic.

2 Related Work

There have been several prior attempts to transliterate texts written in Judeo-Arabic into Arabic script. For other languages and some of the difficulties involved, see Karimi et al. (2011). Modern studies focused on transliteration include (Shazal et al., 2020) for Romanized Arabic (Arabizi) to Arabic, (Jaf and Kayhan, 2021) for Ottoman to the modern Latin Turkish script, and (Shahariar Shibli et al., 2023) for Romanized Bengali (Banglish) to Bengali.

The first attempt at automated transliteration of Judeo-Arabic texts (Kehat and Dershowitz, 2013) employed a method inspired by statistical machine translation, which had been state of the art until deep neural networks took over. This was followed by Bar et al. (2015) who took a similar approach combined with a recurrent neural network (RNN) that was applied to the transliterated Arabic text to handle specific errors, notably those associated with *ta-marbuta*, *hamza*, and *shadda*. In both of those studies, the transliteration procedure is based on a log-linear model, where the main component is a phrase table that captures the number of occurrences of each character in the training data. They used relatively short parallel texts for training the model, which they evaluated on a small test set of 500 words.

[3] https://geniza.princeton.edu/en

[4] http://fjms.genizah.org

[5] Examples include: Passover Haggadoth at https://www.jewishlanguages.org/images-of-haggadot and https://yahad.net/collection; a few manuscripts from the Library of Congress's collection at https://www.loc.gov/collections/hebraic-manuscripts/?q=arabic; some modern texts at https://minds.wisconsin.edu/bitstream/handle/1793/8064/myintro.html; and late 19th and first half of the 20th century newspapers at https://www.nli.org.il/en/newspapers/?lang=Judeo-Arabic.

In a more recent work (Terner et al., 2020), the authors trained a model to automatically transliterate Judeo-Arabic texts into Arabic using an RNN, combined with the connectionist temporal classification (CTC) loss to deal with unequal input and output lengths. They increased the size of the training set by generating some parallel texts synthetically. That brought some improvement over the baseline.

To the best of our knowledge, no previous work has proposed using a pre-trained language model for transliteration, as we introduce here.

3 Methodology

To transliterate a Judeo-Arabic text into Arabic, we employ a two-step approach. The first step involves code switching, where we identify non-Arabic words that are not required for transliteration in the subsequent step. In the second step, we convert each Arabic word from the Judeo-Arabic Hebrew script to the Arabic script. Before delving into the details of each step, we provide a summary of the data sources utilized in both processes.

3.1 Sources

We utilize the following sources to train both the code switching and transliteration models:[6]

Friedberg. We downloaded 110 sources from the Friedberg Judeo-Arabic Project,[7] comprising a total of 3.9 million words. Notably, in all these sources, non-Arabic borrowings have been manually annotated.

Kuzari. The *Kuzari*, originally titled in Arabic, *Kitâb al-ḥujja wa'l-dalîl fi naṣr al-dîn al-dhalîl*, is a medieval philosophical treatise written by Judah Halevi in Andalusia (circa 1140). It was recently published in Arabic by Nabih Bashir (Ha-Levi, 2012).

Mishnah. Maimonides' introduction to his *Commentary on the Mishnah* (1168) was recast in Arabic by Nabih Bashir.

Beliefs. The *Book of Beliefs and Opinions* (*Kitāb al-Amānāt wa l-I'tiqādāt*) by Saadia Gaon (933) was also recast in Arabic by Nabih Bashir.

Al-Falasifa. *The Incoherence of the Philosophers* (*Tahafut Al-Falasifa*) by Al-Ghazali (1095) was composed in Arabic (Nigst et al., 2023).

Al-Tahafut. *The Incoherence of the Incoherence* (*Tahafut Al-Tahafut*) by Averroes (1180) was written in Arabic (Nigst et al., 2023).

Writers of Judeo-Arabic do not adhere to one uniform set of orthographic rules. Not only writers, but modern printers may be inconsistent too. Specifically, an apostrophe or dot might signify or differentiate letters (e.g. *hamza*, *ein*), and in other corpora may be partially or entirely omitted. In light of these inconsistencies, we chose to remove all apostrophes and diacritics from the Judeo-Arabic text as a preprocessing step. Furthermore, we removed all punctuation marks because their usage in Judeo-Arabic does not necessarily correspond to standard modern Arabic conventions.

As described in subsequent sections, we develop models for both code-switching and transliteration by fine-tuning a language model for each task. Given that Judeo-Arabic consists of Arabic words written in Hebrew script, enriched with borrowings from Hebrew and Aramaic, we opt not to use a standard Arabic language model. Instead, we utilize the recently published, openly available BERT-style language model HeArBERT (Rom, 2024), which was trained on a large corpus containing both Hebrew and Arabic texts, in which Arabic was converted into corresponding Hebrew letters.

3.2 Code Switching Detection

We approach code switching as a token classification task. Each token is assigned one of two labels: "Arabic" or "non-Arabic". To achieve this, we fine-tune HeArBERT specifically for token classification using the entirety of the Friedberg dataset. In this dataset, non-Arabic words are distinctly marked. Given that HeArBERT utilizes a WordPiece tokenizer, we ensure alignment between the original span annotations from the dataset and the tokens. Consequently, every token falling within a non-Arabic span receives the "non-Arabic" label.

Overall, the dataset comprises approximately 3.9 million tokens. Of these, 34% are labeled as "non-Arabic". We allocate 10% of the data for testing, using the remainder for training purposes.

Morphologically code-switched words. In Judeo-Arabic, some Hebrew words carry Arabic prefixes. For example, the word אלמשכילים (*al-maskilim*), which translates to "the philosophers". In this word, the definite article אל (*al*) originates from Arabic, but the stem משכילים (*maskilim*) is borrowed from Hebrew. In the original Friedberg

[6] These sources can be found at https://github.com/dwmitelman/ja_transliteration_tool/tree/main/resources/scrapes.

[7] http://fjms.genizah.org

dataset, words that are a fusion of Arabic and Hebrew components are mostly tagged as Arabic. In our code-switching procedure, we aim to reflect the linguistic complexity of such words more accurately. We do this by labeling the Arabic prefix as "Arabic" and the stem (typically of Hebrew origin) as "non-Arabic".

To do this, we analyze every word having any of the following prefixes: *al* (ال), *lil* (لل), and *bil* (بال). We estimate the frequency of the stem (the word stripped of its prefix) in both Arabic and Hebrew, using some available lexicons.[8] A word is labeled "non-Arabic" (with an Arabic prefix) if it demonstrates low frequency in Arabic, both with and without the prefix, and concurrently shows a high frequency in Hebrew without the prefix.

Broadly speaking, we use the code-switching model to identify non-Arabic words that we avoid transliterating into the Arabic script in the subsequently-applied transliteration model.

3.3 Transliteration

We define the task of transliterating from Hebrew script to Arabic script as a character classification challenge. For each Hebrew (Judeo-Arabic) character input, we produce either a corresponding Arabic character or an epsilon (ε) to signify the absence of a character. The first step toward training such a model involves preparing parallel texts to serve as the training dataset.

Three digitally-available works provided us with parallel texts: Halevi's *Kuzari*, Maimonides' *Mishnah*, and Saadia's *Beliefs*. However, the texts are not perfectly aligned at the word level. This misalignment occurs because some Judeo-Arabic words lack an Arabic equivalent. Additionally, sometimes the paired Arabic word serves as a semantic equivalent, chosen by the translator, especially when the original word is no longer in use in Modern Standard Arabic (MSA). Therefore, a naïve algorithm that pairs words from the two texts in order would be unreliable. To address these challenges, we developed a new alignment algorithm, which comprises the following steps:

(1) Construct a table to document the frequency of each Arabic word in the text.

(2) Compute the average word length for words that appear only once.

(3) For each word that occurs once and has an at least average length, transliterate it into the Hebrew script and search for its occurrence in the Judeo-Arabic text. The transliteration is done deterministically using a lookup table (Table 7a in the appendix). Note that some letters might be entirely omitted from the transliteration. In the table, these letters are signified by allowing their transliteration to be ε. A word is only considered an anchor if we find it within a range of five words before or after the exact location (based on word index) of the original word in the corresponding Arabic text.

(4) Divide the two parallel texts into segments, using the anchor words as delineation points.

(5) For each segment, compare every pair of parallel words as follows: Transliterate the word from Arabic script into all its Hebrew script variations, then match each variation with the original Judeo-Arabic word. Perform this process in the opposite direction as well: Transliterate the Hebrew script word into all its Arabic variations (using Table 7b), Then, match words in the reading direction. To determine a match between an Arabic word and its Judeo-Arabic counterpart, we start by considering all the Hebrew-script transliteration variations of the original Arabic word, comparing them to the original Judeo-Arabic word. Should multiple transliteration variations align perfectly, we select the one generated with the fewest epsilons. In the absence of a match, we reverse the process: We examine the Arabic transliteration variations of the original Judeo-Arabic word and compare them to the original Arabic word, adhering to the same epsilon minimization approach.

(6) Store training instances as a pair of character-level sequences.

The rationale behind setting a minimum length for anchor words is to avoid selecting common words. Accurately aligning individual occurrences of words that are frequent in the texts would be challenging. Note that this algorithm is not accurate. It may reject aligned words and in rare cases, it may

[8] https://github.com/hermitdave/FrequencyWords. The Arabic lexicon contains approximately 1.2M words, while the Hebrew one has around 0.9M.

accept wrong pairs. Yet, since this is used only for training data, it doesn't have to be accurate.[9]

Dataset expansion. To boost the number of training instances for the model, we utilize texts from pertinent Arabic sources. The Jewish philosophers of that era were influenced by their Muslim counterparts. Consequently, we have selected texts from *Al-Falasifa* and *Al-Tahafut*. However, these sources exist solely in Arabic, lacking a parallel Judeo-Arabic rendering. To address this gap, we artificially generate a Judeo-Arabic version using a straightforward algorithm: We use two out of the three Judeo-Arabic books, *Mishnah* and *Beliefs*, which were previously aligned with their Arabic counterparts, to generate Judeo-Arabic mappings for each Arabic letter and letter bigram. It bears stressing that a monomer (single letter) or dimer can correspond to several mappings. We maintain a record of the frequency for each of these mappings. These records are compiled into what we call a *mapping collection*. This collection consolidates all the mappings for a specific monomer or dimer, along with their frequencies as documented in the three Judeo-Arabic books. To create a Judeo-Arabic version of each Arabic book, we proceed letter by letter in reading order. Our primary attempt is to find a mapping collection for the dimer comprising the current and preceding letters. If successful, we sample a single mapping from its collection, using the frequencies as weights. In the absence of a dimer match, we resort to the mapping collection of the individual letter, employing the same frequency-weighted sampling approach. A complete list of all resulting sources and their corresponding number of words is provided in Table 1. We evaluate the performance of the transliteration model trained with and without the synthetically generated sources. Across all our transliteration experiments, we exclude the *Kuzari* test set (used in (Terner et al., 2020)) from the training set, using only the rest (about 80%).

Transliteration model. We approach the transliteration task from the Hebrew script to the Arabic script as a token classification task, where the tokens are constrained to characters. Each Hebrew letter can be transliterated into one of 34 tags: 33 Arabic letters[10] and the "epsilon" tag. The epsilon tag is used to denote Judeo-Arabic letters that are entirely omitted in the Arabic version. Just as with code switching, we base our transliteration model on HeArBERT by fine-tuning it on the token classification task. However, in contrast to code switching, to restrict tokens to letters only, we modify the model's tokenizer vocabulary by eliminating all tokens that do not represent individual Hebrew or Arabic letters. Given that the original HeArBERT WordPiece tokenizer was trained on complete tokens, we posit that the representation of single-letter tokens in the model might be somewhat diminished. To address the potentially weakened representation of single-letter tokens, we suggest an additional step before fine-tuning the model for the transliteration task. We continue in pre-training the language model using the original masked-language-modeling (MLM) task with 15% masked tokens (now, only single letters). We utilize the entire Friedberg dataset, which contains 3.9M words, for training the model. This training spans ten epochs with a learning rate set to 2×10^{-5}. We evaluate the performance of the transliteration model with and without this continuous pre-training step. It is important to highlight that we utilize the epsilon tag to manage Judeo-Arabic letters that are omitted in the Arabic transliteration. However, we consciously omit handling letters that are introduced in the Arabic version, like the *hamza* in the word مساء *masā'a*, which is conventionally written as מסא in Judeo-Arabic. While this could be perceived as a limitation of our methodology, it is rooted in historical context: documentary middle Arabic seldom employed the *hamza*. Studies of manuscripts from the initial 300 years indicate that Classical Arabic was largely a construct of grammarians, diverging from the way most individuals—including scribes of the Quran—actually penned Arabic (van Putten, 2022).

4 Results

4.1 Language Tagging

As mentioned above, for the code-switching task we split the 3.9M-word dataset with 90% for training and 10% for testing, and train the model for the standard token classification task for the duration of ten epochs, using a learning rate value of 2×10^{-5}

[9] The aligned datasets are at https://github.com/dwmitelman/ja_transliteration_tool/tree/main/resources/align.

[10] A full list of the Arabic letters we use can be found in Table 7b of the appendix. Note that we ignore different *alif* forms (*hamza* above or below, *madda*, *wasla*), *shadda*, and all vocalization marks. The transliterated text is still intelligible.

	Total Words	non-Ar Words	Ar Words	Align Rate	Aligned Words	Aligned Letters
Kuzari (JA)	47,334	5,392	41,942	95.8%	40,194	174,077
Beliefs (JA)	67,898	11,648	56,250	92.2%	51,876	214,704
Mishnah (JA)	15,638	3,798	11,840	74.1%	8,779	36,157
Al-Falasifa (Ar)	Synthetic (Ar only)				48,988	206,794
Al-Tahafut (Ar)	Synthetic (Ar only)				106,074	438,890

Table 1: Number of words and letters of the Judeo-Arabic (JA) and Arabic (Ar) sources, with division into the type of words and alignment success rate between Judeo-Arabic and Arabic.

	Judeo-Arabic			Non-Arabic		
Acc	Pre	Rec	F1	Pre	Rec	F1
98.46	98.97	98.70	98.83	97.53	98.04	97.78

Table 2: Evaluation of code-switching. The first column is the overall accuracy; the rest of the columns are pre(cision), rec(all) and F1 for the two labels.

and batch size of 32. The evaluation results are summarized in Table 2.

4.2 Transliteration

Table 3 summarizes the transliteration model's evaluation on *Kuzari*, including both the macro average F1 and accuracy. It shows the model's performance at various stages of its development. The best results are obtained in the last row, with both the continuous pre-training step and the inclusion of the artificially generated parallel data in the training set.

The accuracy and macro F1 are quite different; this is due to the fact that the distribution of the labels (Arabic words) is unbalanced. The relatively high accuracy values suggest that some Judeo-Arabic letters are relatively easy to transliterate into Arabic, and some are more difficult. Therefore, in addition to reporting accuracy and F1 on the entire set of letters, we report these metrics on a smaller set of letters, those that are harder to transliterate. The "hard" Arabic letters are those that stem from a Judeo-Arabic origin letter that could be converted into more than one Arabic letter, namely ت (*t*), ث (*th*), ج (*j*), خ (*kh*), د (*d*), ذ (*dh*), ص (*ṣ*), ض (*ḍ*), ط (*ṭ*), ظ (*ẓ*), غ (*gh*), ك (*k*), ء (□), ؤ (*wāw hamzah*), ئ (*yā' hamzah*), ى (*'alif maqṣūrah*).

The per-letter results are summarized in Table 4. Table 5 is a standard confusion matrix for the outcomes. Additionally, Table 8 in the appendix delineates the frequencies with which each Judeo-Arabic letter is converted to its respective Arabic letter.

We compare the performance of our transliteration model with (Terner et al., 2020)—the best prior system—using the label error rate (LER) as defined by those authors, which captures the average wrong labels per word. The formula is $\frac{1}{|S|}\sum_{(x,z)} ED\left(h\left(x\right),z\right)/|z|$, for model h on test data $S \subseteq X \times Z$, where X are the inputs, z is ground truth and $|z|$ is the length of z. The Levenshtein distance, *ED*, is calculated between the predicted characters and the ground truth. It is then normalized by the length of the ground truth. This is a natural measure for a model where the aim is to produce a correct label sequence (Graves et al., 2006). We evaluate our model on exactly the same test set provided by those authors, which was taken originally from the *Kuzari*. Our model achieves 1.40% LER, which is much better than the LER of 2.48% that was reported by Terner et al. (2020); note that by (Terner et al., 2020), simple mapping from Judeo-Arabic to Arabic achieves an LER of 9.51%.

5 Conclusions

We have established a pipeline that integrates the two models we introduced in this work: code-switching detection and transliteration.[11] This pipeline processes Judeo-Arabic text by first identifying non-Arabic words, which do not require transliteration into Arabic, followed by the transliteration of words recognized as Arabic. In Table 6, we provide some sample sentences that were processed with our pipeline. Some notes on the examples (numbers refer to the row in the table): (1) The original text has apostrophes and punctuation. As explained in Section 3.1, we have removed all characters that are not Hebrew letters. The third (والاعتدار) and tenth (اعتدارنا) words have been transliterated mistakenly; still, the rest of the letters were correctly transliterated. (2) The second word is a combination of an Arabic prefix ال ("the") and a Hebrew noun משכילים ("philosophers"). Therefore, this word has been divided, and the Arabic prefix was transliterated into Arabic. (4) Similar to (2),

[11] Our pipeline is available at https://github.com/dwmitelman/ja_transliteration_tool/tree/main.

Continuous MLM	Synthetic Data	Macro Precision		Macro Recall		Macro F1		Accuracy	
		All	Hard	All	Hard	All	Hard	All	Hard
✘	✘	79.7	52.8	76.0	46.1	76.0	46.4	95.3	79.7
✘	✔	83.3	55.2	82.7	54.1	82.9	54.4	96.9	86.1
✔	✘	83.7	55.6	83.1	54.6	83.2	54.8	97.2	87.1
✔	✔	**87.0**	**60.8**	**86.1**	**59.1**	**86.0**	**59.1**	**98.0**	**90.8**

Table 3: Evaluation results of the transliteration model. The first row presents results achieved using the unmodified HeArBERT model, but restricted to single-letter tokens. The second gives results obtained after continuous pre-training of the model using the 3.9M-word Friedberg corpus. The final row shows the impact of adding synthetically generated parallel data to the training set.

there is a word with an Arabic prefix comprising a preposition and the definite article للـ and a Hebrew word רשעים ("the wicked"). (5) The first word ואכתמו represents the word in Arabic واختموا ("you should sign"), and ends with a silent *alif* (ا). Since this letter was not written in the Judeo-Arabic, it has not been transliterated back to Arabic.

In summary, our methodology, which utilizes a pre-trained language model, outperforms the best existing model (Terner et al., 2020), evaluated on the same test set. We observe two primary differences between the two. First, while both models are trained for token classification with tokens represented as single letters, our model leverages a pre-trained language model that we further fine-tune using relevant Judeo-Arabic documents. The second distinction lies in the size of the training set; our model utilizes a larger dataset, a consequence of our more advanced robust alignment algorithm.

Dedicated models per genres. Most of our training and test work was performed with a specific, literary genre of data. Classical authors, like Halevi and Saadia whose works we used for training, each follow fixed transcription rules and were consistent in their transliterations from Arabic to Hebrew script. Accordingly, the conversion tool that we created is somewhat crippled when dealing with texts from other genres. Inventory lists, prescriptions, newspapers, and other quotidian documents, written by a large variety of people, may be too diverse in style and too varied in spelling. This leads to the question whether there can be a perfect comprehensive tool that will be able to transliterate every Judeo-Arabic text. Without answering the question, we suggest that, with prior semi-classification, these texts could be transliterated better. One potential enhancement can be done by sampling some specific words, which contain "hard" letters, and determining parameters for the map from Arabic to Hebrew script, consistency in letter mapping, and variety of vocabulary that is used. Armed with this information, we could build downstream post-processors to provide text corrections, or we may even fine-tune individual models for different styles and genres.

Other languages. Judeo-Arabic is not the only language written in a different script than usual for its base language. Other Jewish languages, like Judeo-Persian, Judeo-Yemenite, Ladino, or even Yiddish, are similarly written in Hebrew characters. Various languages of countries in the former USSR and its sphere of influence have undergone Russification. Texts in Polish, Romanian, Serbian, Mongolian, and many other languages have been published in the Cyrillic alphabet, or an extension thereof. In the internet and social-media age, texts in many languages have been shoehorned into using the Latin alphabet, leading to informal written forms like Arabizi and Romanized Hindi. The ideas we developed should help inform efforts to re-express such texts as well.

Limitations

Context awareness. The character-based language model used for transliteration minimizes context information, hindering the accurate transliteration of special cases, like passive verbs, that impact word vowelization and specific *hamza* letters. Selecting between ؤ and ئ proves difficult for the model, which might improve with enhanced context awareness.

Aramaic coverage. We also tried to use Aramaic corpora to aid in the detection of borrowed words with an Arabic prefix, but the quantity of available texts was insufficient.

Diacritics. We ignored non-Hebrew characters due to the inconsistency in writer and publisher conventions, avoiding potential noise and unexpected

Letter	Precision	Recall	F1	Support
ب	1.000	1.000	1.000	5909
ح	1.000	1.000	1.000	2922
ر	1.000	1.000	1.000	6930
ز	1.000	1.000	1.000	834
س	1.000	1.000	1.000	3321
ش	1.000	1.000	1.000	1372
ف	1.000	1.000	1.000	5304
ق	1.000	1.000	1.000	4435
ل	1.000	1.000	1.000	21337
ن	1.000	1.000	1.000	10139
م	1.000	0.999	0.999	11481
ع	0.999	1.000	0.999	5724
ا	0.996	0.998	0.997	30475
و	0.987	1.000	0.993	11284
ت	0.984	0.995	0.990	6175
ه	0.982	0.967	0.974	7773
ك	0.972	0.975	0.973	4590
د	0.962	0.982	0.972	3868
ط	0.972	0.970	0.971	1173
ج	0.954	0.958	0.956	1767
ي	0.918	0.990	0.953	11446
ة	0.932	0.967	0.949	3538
ذ	0.966	0.931	0.948	2137
ص	0.935	0.951	0.943	1779
ظ	0.937	0.940	0.938	550
ث	0.963	0.897	0.929	944
خ	0.916	0.901	0.911	1405
ض	0.920	0.897	0.908	1134
غ	0.895	0.883	0.889	711
ّ	0.942	0.601	0.733	323
ئ	0.796	0.578	0.669	559
ى	0.939	0.442	0.601	1600
ء	0.013	0.118	0.024	17
ؤ	0.000	0.000	0.000	121

Table 4: Results per letter, sorted by F1 score.

behaviors. While this choice omitted some informative Arabic characters, future work will employ various language models that include these marks.

Ethics Statement

We see no potential ethical issues in this work.

Acknowledgments

We thank Nabih Bashir, Yonatan Belinkov, Yoav Phillips, Marina Rustow, and researchers at the Princeton Geniza Project. This research was funded in part by the European Union (ERC, MiDRASH, Project No. 101071829). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Kfir Bar, Nachum Dershowitz, Lior Wolf, Yackov Lubarsky, and Yaacov Choueka. 2015. Processing Judeo-Arabic texts. In *Proceedings of the First International Conference on Arabic Computational Linguistics (ACLing '15, Cairo, Egypt)*, pages 138–144.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376. ACM.

Judah Ha-Levi. 2012. *The Kuzari – The Book of Refutation and Proof on the Despised Faith*. Al-Kamel Verlag, Freiberg. Transliterated and edited by Nabih Bashir with assistance of 'Abed 'l-Salam Muosa.

Benjamin Hary. 2018. Judeo-Arabic in the Arabic-speaking world. In B. Hary and S. Benor, editors, *Languages in Jewish Communities, Past and Present*, pages 35–69. de Gruyter, Boston.

Adina Hoffman and Peter Cole. 2011. *Sacred Trash: The Lost and Found World of the Cairo Geniza*. Schocken, New York.

Ashti Afasyaw Jaf and Sema Koç Kayhan. 2021. Machine-based transliterate of Ottoman to Latin-based script. *Scientific Programming*, 2021:1–8.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Comput. Surv.*, 43(3).

Gitit Kehat and Nachum Dershowitz. 2013. Statistical transliteration of Judeo-Arabic text. In *Israeli Seminar on Computational Linguistics (ISCOL)*, Beersheba, Israel. Abstract.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2023. OpenITI: A machine-readable corpus of Islamicate texts. Zenodo.

Yoav Phillips. 2020. Identifying the Islamic sources of *baḥya ibn paqūda's: "kitāb al-hidāya □ilā farā□iḍ al-qulūb"*: Using automatic transliteration and text reuse processes. Master's thesis, Haifa University, July.

Aviad Rom. 2024. Processing dialectal Arabic with Transformer-based language models: Challenges and potential solutions. Thesis, Reichman University, Israel.

G. M. Shahariar Shibli, Md. Tanvir Rouf Shawon, Anik Hassan Nibir, Md. Zabed Miandad, and Nibir Chandra Mandal. 2023. Automatic back transliteration of Romanized Bengali (Banglish) to Bengali. *Iran Journal of Computer Science*, 6(1):69–80.

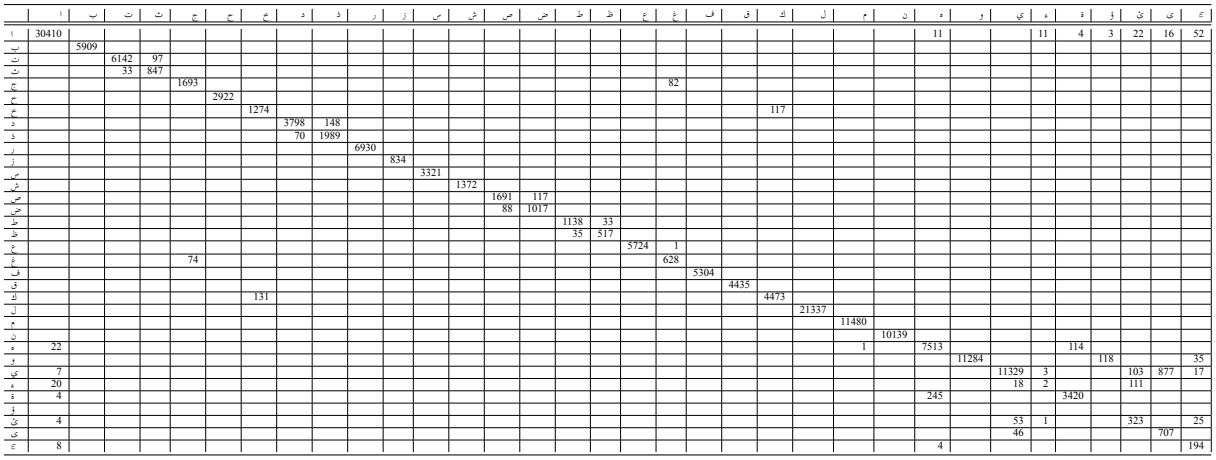

Table 5: Confusion matrix for all Arabic letters. Rows are predicted labels; columns are true labels.

	Transliterated Arabic (output)	Judeo-Arabic (cleaned input)
1	فاحصار النية والاعتدار عن العمل نافع بعض منفعة مثل اعتدارنا في صلاتنا ومفني חטאינו גלינו מארצנו وما اشبه ذلك *fahsar alniya wal'tidar 'ani l'amal nafi' ba'ḍ manfa'a mithl i'tidarna fi ṣalatina* ***umipney khataenu galinu meartsenu*** *wama aushbuhu dhalika*	פאחצאר אלניה ואלאעתדאר ען אלעמל נאפע בעץ מנפעה מתל אעתדארנא פי צלאתנא ומפני חטאינו גלינו מארצנו ומא אשבה דלך
2	فصار המשכילים لانفسهم مثل الرقيع ومستصلحي غيرهم كالشمس والكواكب *faṣara ḍalmaskilim lianfashum mithl alraqi' wamustaṣliḥi ghayrahum kashams walkawakib*	פצאר אלמשכילים לאנפסהם מתל אלרקיע ומסתצלחי גירהם כאלשמס ואלכואכב
3	للרשעים فقط *lil**resha'im** faqaṭ*	ללרשעים פקט
4	واختمو علي بجميع الالفاظ *waikhtimu 'ala bijami' alalfaẓ*	ואכתמו עלי בגמיע אלאלפאט
5	יהודה הזקן הנכבד נע وقال لنا ***yehuda hazaqen hanikhbad nishmato eden*** *waqala lana*	יהודה הזקן הנכבד נע וקאל לנא

Table 6: Examples of sentences processed with our Judeo-Arabic pipeline. Words (or morphemes) in the input that are Hebrew, not Arabic, are not transliterated. Phonetic transliteration is provided here for readability.

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 (WANLP 2020)*, pages 167–177, Barcelona, Spain (Online). Association for Computational Linguistics.

Ori Terner, Kfir Bar, and Nachum Dershowitz. 2020. Transliteration of Judeo-Arabic texts into Arabic script using recurrent neural networks. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop (WANLP 2020)*, pages 85–96, Barcelona, Spain (Online). Association for Computational Linguistics.

Ofra Tirosh-Becker, Michal Kessler, Oren Becker, and Yonatan Belinkov. 2022. Part-of-speech and morphological tagging of Algerian Judeo-Arabic. *Northern European Journal of Language Technology*, 8(1).

Marijn van Putten. 2022. *Quranic Arabic: From its Hijazi Origins to its Classical Reading Traditions*, volume 106 of *Studies in Semitic Languages and Linguistics*, chapter 6: Classicized Hijazi: Imposition of the *hamzah*. Brill, Leiden, The Netherlands.

Donald Winford. 2003. Code switching: Linguistic aspects. In *An Introduction to Contact Linguistics*, chapter 5, pages 126–167. Blackwell Publishing, Malden, MA.

Appendix: Transliteration Tables

In Table 7a, we present the lookup table used for transliterating Arabic words from Arabic script into Hebrew script. Since each Arabic letter may correspond to multiple Hebrew characters, utilizing this table may result in several potential Hebrew transliteration variations for a given Arabic word. The choice of some forms (medial, final) is determined by the position of the letter in the word.

Table 7b is a similar lookup table for deterministically transliterating Judeo-Arabic words from the Hebrew script into the Arabic. Some Hebrew letters correspond to multiple Arabic characters. Some forms (initial, medial, final) are determined by the position of the letter in the word.

Table 8 contains the frequencies at which each Judeo-Arabic letter is converted to the respective Arabic letter.

Arabic (from)	Hebrew (to)
ا	א, ε
ب	ב
ت	ת
ث	ת', ת
ج	ג', ג
ح	ח
خ	ח, כ, ך', כ', ח'
د	ד
ذ	ד, ד'
ر	ר
ز	ז
س	ס
ش	ש
ص	צ, ץ
ض	ץ', ד, צ'
ط	ט
ظ	ז, ד, ט
ع	ע
غ	ע, ג
ف	ף, פ
ق	ק
ك	ך, כ
ل	ל
م	ם, מ
ن	ן, נ
ه	ה, ε
و	ו, ε
ي	י
ء	א, י, ε
ة	ה', ה
ؤ	ו, ε
ئ	א, י, ε
ى	א, י, ε

(a) Transliteration table from Arabic to Hebrew. (ε means no substitution.)

Hebrew (from)	Arabic (to)
א	ا, ى, ئ, إ, أ, آ, ء, ة, ه, أ, ؤ, ε
ב	ب
ג	غ, **ج**
ד	ذ, ض, ظ, **د**
ה	ا, ة, **ه**
ו	**و**, ؤ, ε
ז	ظ, **ز**
ח	خ, **ح**
ט	ظ, **ط**
י	**ي**, ى, ئ, ا, ε
כ	خ, **ك**
ל	ل
מ	م
נ	ن
ס	س
ע	غ, **ع**
פ	ف
צ	ض, **ص**
ק	ق
ר	ر
ש	ش
ת	ث, **ت**
ך	خ, **ك**
ם	م
ן	ن
ף	ف
ץ	ص, **ض**

(b) Transliteration table from Hebrew to Judeo-Arabic. (ε means no substitution. The Arabic letter in bold is the one most commonly transliterated.)

	א	ב	ג	ד	ה	ו	ז	ח	ט	י	ך	כ	ל	ם	מ	ן	נ	ס	ע	ף	פ	ץ	צ	ק	ר	ש	ת
ا	30528				1																						
ب		5909																									
ت																											6239
ث																											880
ج			1775																								
ح								2922																			
خ											3	1388															
د				3946																							
ذ				2059																							
ر																									6930		
ز							834																				
س																		3321									
ش																										1372	
ص																						95	1713				
ض																						299	806				
ط									1171																		
ظ									552																		
ع																			5725								
غ			702																								
ف																				519	4785						
ق																								4435			
ك											1190	3414															
ل													21337														
م														2871	8609												
ن																4867	5272										
ه					7649									1													
و						11437																					
ي										12336																	
ء	42									109																	
ة					3669																						
ؤ																											
ئ	42									364																	
ى										753																	
ε	206																										

Table 8: Frequencies of conversions of each Judeo-Arabic letter to each Arabic letter (columns: input; rows: prediction).