On Cross-script Information Retrieval

Nada Naji[1*], James Allan[2]

[1] College of Computer and Information Science
Northeastern University, USA
najin@ccs.neu.edu

[2] Center for Intelligent Information Retrieval, College of Information and Computer Sciences
University of Massachusetts Amherst, USA
allan@cs.umass.edu

Abstract. We address the problem of cross-script retrieval in the context of a microblog system such as Twitter. Specifically, we explore methods for using native Arabic script queries to retrieve Arabic tweets written in a Roman script known as Arabizi. For example, a query for "كتاب" would not match *"kitab"* even though an Arabic reader would see them as the same word. Moreover, because of the lack of Arabic script, automatic language identification methods fail to recognize the Arabizi text as Arabic and label it as English, Polish, or the like. We propose a cross-script retrieval system using automatic rule-based mapping and statistical selection of transliteration keywords. We show that our system can achieve effective cross-script retrieval with minimal knowledge of the target language and without the need to rely on external translation or transliteration tools or lexica. With minimal human annotation, our technique can be applied to other languages such as Hindi and Greek, which are commonly converted to a Roman character set similarly.

Keywords: Cross-script IR, CSIR, social media retrieval, Arabic, Arabizi, cross-language IR, CLIR, mixed-script IR, MSIR, transliteration.

1 Introduction

The Web contains huge amounts of user-generated text in different writing systems and languages but most popular platforms lack the mechanism of implicitly cross-matching Romanized versus native script texts. Twitter's language identifiers seem to only attempt to detect a language when written in its native/official character set. While it succeeds at identifying Arabic most of the time, Twitter does not detect nor identify Arabizi tweets as Arabic ones nor does it count Arabizi as a stand-alone language. Therefore, potentially novel and pertinent content is unreachable by simple search. Our proposed method for identifying Arabizi is intended to help with that challenge. The contributions of this paper are the following: 1) We describe an Arabic to Arabizi transliteration that works in the absence of lexica and parallel corpora. 2) We develop an approach to evaluate the quality of such a transliterator. 3) We demonstrate that our transliterator is superior to reasonable automatic baselines for identifying valid Arabizi transliterations. 4) We make the annotated data publicly available for future research[1].

* This work was done while the author was at the University of Massachusetts Amherst, supported by the Swiss National Science Foundation Early Postdoc.Mobility fellowship project P2NEP2_151940

[1] https://ciir.cs.umass.edu/downloads/

2 Related Work

The problem of spelling variation in Romanized Arabic has been studied closely to perform Named Entity Recognition such as Machine Translation (MT) of Arabic names [8] and conversion of English to Arabic [9]. However, and to the best of our knowledge, no work has been done so far on cross-script Information Retrieval (CSIR) for the Arabic language. Some studies addressed dialect identification in Arabic or Arabizi [1, 3, 4, 5] and statistical MT from Arabizi to English via de-romanization to Arabic [11]. Arabic to Arabizi conversion has only been done as one-to-one mapping such as Qalam[2] and Buckwalter[3] resulting in Romanized vowel-less text. Darwish [2] uses a Conditional Random Field (CRF) to identify Arabizi from a corpus of mixed English and Arabic tweets with accuracy of 98.5%. We are typically transcribing single words or short phrases, where the CRF rules do not work well. Gupta et al.'s work on mixed-script IR (MSIR) [6, 7] proposes a query expansion method to retrieve mixed text in English and Hindi using deep learning and achieving a 12% increase in MRR over other baselines. In contrast to their work, we are using a transliteration-based technique that does not rely on lexica or datasets. Also, we are faced with very short documents lacking the redundancy that can be used to grasp language features. Bies et al. [12] released a parallel Arabic-Arabizi SMS and chat corpus of 45,246 case-sensitive tokens. Although it is a valuable resource, it only covers Egyptian Arabic and Arabizi.

3 Cross-Script Retrieval Task Description

Let q be a query in language l written in script s_1. A CSIR system retrieves documents from a corpus C in language l in response to q, where the documents are written in script s_1 or an alternative script s_2 or both s_1 and s_2, and where s_2 is an alternative writing system for l. The underlying corpus C may consist of documents in n languages and m scripts such that $n \geq 1$ and $m \geq 2$. Our definition of the CSIR problem is analogous to Gupta et al.'s definition of MSIR [6], but in their experimental setup, Gupta et al. focus on bilingual MSIR ($n=2$ and $m=2$). We address the problem of a both multi-lingual and multi-scripted corpus ($n \geq 2$, and $m \geq 2$) which is a complex task since vocabulary overlap between different languages is more likely to happen as more languages and more scripts co-exist in the searchable space. We describe our transliteration and statistical selection algorithms below:

AR → ARZ Exhaustive Transliterator: We implement our word modeling algorithm to generate Arabizi forms for a given word in Arabic as described below:

1- Perform AR to ARZ mapping for stable consonants (Table 1). For example, ("كتاب") is mapped to "*ktb*". If the mapping is non-unique, enumerate all possible instances and apply the remaining steps to each candidate.

2- Map and handle long vowels, diphthongs and *hamza*: ('و'), ('ي'), ('ا', 'ى'') or ('ء', 'أ', 'آ', 'أ', 'إ'), with an option to introduce'2' for hamza either alone or combined with a long vowel. Since ("كتاب") contains the long vowel ('ا') *'a'* is inserted accordingly "*ktab*".

3- Generate possible *tashdeed* (emphasis) instance(s) for the second and subsequent consonants or ('و') or ('ي'), then apply the remaining steps on all enumerated instances. "*kttab*", "*kttabb*", "*kttabb*".

4- Pad consecutive non-emphasis consonants or ('و') or ('ي') with an optional short vowel *(v)* (one of *'a', 'e', 'i' 'o',* or *'u'*). "*k(v)tab*", "*k(v)ttab*", "*k(v)tabb*", "*k(v)ttabb*" → *kitab*, *kuttab*, *ktabb*, *kattabb*.

[2] Webpage accessed January 3rd 2016, 19:17 http://langs.eserver.org/qalam

[3] Webpage accessed January 3rd 2016, 19:18 http://languagelog.ldc.upenn.edu/myl/ldc/morph/buckwalter.html

Steps 3 and 4 allow accounting for the dropped diacritics in Arabic. For example, "مصر" can be found as "مِصْر" (*"misr"*) (Egypt) and can also be written as *"masr"*,*"m9r"*, etc.

Arabizi Keyword Selection: To determine the potential Arabizi forms we need to quantify the adequacy of the elaborately produced transliterations. We propose *K score* which measures the "*Arabiziness"* of the resulting transliterations based on their occurrences and association with certain linguistic features across the corpus based on our hypothesis that if a word is Arabizi, it will frequently occur in the presence of other Arabizi words. In particular, it will occur in the presence of common function words such as stopwords. On the other hand, Arabizi candidates that rarely or never occur with other Arabizi words are likely to be words in other languages rather than Arabizi tokens. In operation, K score is systematically provided with the transliterations generated by our word modeling module then measures the Arabiziness of each input form according to the following algorithm:

1- Term Projection: Given the exhaustive set of Arabizi transliterations (***W****ord* ***T****ransliteration*): $WT_{ARZ} = \{WT1_{ARZ}, .., WTn_{ARZ}\}$. For a given single-term Arabic W_{AR} intersect WT_{ARZ} with the set of actually occurring terms using the inverted index: $W_{ARZ} = Ix \cap WT_{ARZ} = \{W1_{ARZ}, W2_{ARZ, ..., } Wn_{ARZ}\}$ where $Ix = \{Ix_1, Ix_2, ..., Ix_N\}$

2- For each transliteration Wi_{ARZ} in W_{ARZ}, find the subset of tweets T_{Wi} that contain Wi_{ARZ} at least once: $T_{Wi} = \{ t1_{Wi}, t2_{Wi}, .., tS_{Wi} \}$

3- For each tweet set T_{Wi}, find the union of all the tokens appearing in the tweets' set $T_{WiUnion}$

4- Given a predefined set of Arabizi stopwords *SW*, find the number of stopwords appearing in $T_{WiUnion}$: $K = | T_{WiUnion} \cap SW |$

A higher K value indicates the presence of more Arabizi stopwords in the tweet union when the transliteration form in question appears, hence reflecting more potential Arabiziness. A lower K means that there is less confidence that the word is in Arabizi. For example, let $WT_{ma9r} = \{WT1_{ma9r,} WT2_{ma9r, ..., } WTn_{ma9r}\}$ be the set of Arabizi transliterations of "مصر" generated by our AR → ARZ transliterator such that: WT_{ma9r} = *{"m9r", "ma9r", "masr", "masar", "miser", "misr", "mo9ur", "mu9irr"}*. First, WT_{ma9r} elements are projected against the inverted index's list of words *Ix*. Only*"mo9ur"* doesn't appear in *Ix* and is therefore excluded from the resulting W_{ma9r}. Each transliteration element in W_{ma9r} is then linked to the list of tweets in which it appears and a set of the words appearing in those tweets is formed. Assume that *"masr"* appeared in the following pseudo-tweets: $t1_{masr}$ = *"la fe masr.. ana fe masr delwaty fel beet"*, $t2_{masr}$ = *"salam keef el 2hal f masr"*, $t3_{masr}$ = *"creo que en brasil hay masr argentinos que brasileros"*. Whose term union yields the set: $T_{masrUnion}$ = *{"2hal", "ana", "argentinos", "beet", "brasil", "brasileros", "creo", "delwaty", "el", "en", "f", "fe", "fel", "hay", "keef", "la", "masr", "que", "salam"}*. The last step is to obtain the number of Arabizi stopwords that appear in $T_{masrUnion}$, in this case we have *"el", "f", "fe", "fel", and "la"*. Despite the fact that *"el"* and *"la"* overlap with other languages such as Spanish, the other stopwords do not which makes them distinctive features for Arabizi in this case. Finally, the K score is equal to the number of stopwords in $T_{masrUnion}$, hence $K_{masr} = 5$. The same process is repeated with the other transliterations to obtain their respective K values and the transliterations are then sorted accordingly to reflect their Arabiziness.

4 Evaluation and Discussion

Main corpus: Our dataset comprises around 72M tweets that we automatically collected via an API over the period between mid-June and mid-July 2014 regardless of language. The content of "text:" was extracted to create an inverted index. **Queries:** We manually generated 50 single-term Arabic queries in neutral and dialectal forms. **Projected**

corpus: The set of Arabic single-term queries is provided to our AR → ARZ transliterator, each keyword was then mapped to n transliterations ($n > 1$) which were then *sifted* by term projection against the inverted index. **Relevance judgments and human assessment:** The transliterations are then manually judged by our annotators to determine whether each transliteration is a correct Arabizi transliteration (relevant) or not (non-relevant). Legitimate but non-matching Arabizi words were labeled as *edge* (neither relevant nor non-relevant). To ensure fair and abstract judgment, the annotators had to review the transliterations individually and without seeing the tweets. **Stopwords:** Definite articles, prepositions, and conjunctions are attached to the word in Arabic script. Surprisingly, Arabizi writers tend to separate such articles from words [2]. We expanded the set of stopwords indicated by Darwish [2] to include more forms with dialectal variants (54 in total).

4.1 Evaluation Methodology and Baselines

Given an Arabic word, a system outputs a ranked list of Arabizi transliterations. For an Arabic word A, a system outputs k Arabizi words Z_1 to Z_k in ranked order. Our evaluation corpus has the complete list of correct Arabizi words, $Y = \{Y_1, \ldots, Y_m\}$. We calculate the well-known average precision (AP) measure. We average this value for all words in the test dataset to determine the system's MAP or mean AP score. We also provide standard interpolated recall/ precision graphs and measure the reciprocal rank (RR) of the first valid Arabizi word in the ranked list. If Z_i is the best-ranked Arabizi word that is in Y, then the RR for that Arabic word is $1/i$. We average this score over all queries to determine MRR, the mean RR. We provide the following baselines to demonstrate that the K score-based approach is an improvement on obvious solutions to this task. **AllHuman** where only annotator-selected candidates are included. Since these are by definition correct, these results are perfect. (They are provided primarily for verification). **1stHuman** is a human-generated baseline, wherein we used the single best Arabizi transliteration for each Arabic word as provided by the pool of annotators. The remaining baselines are automatically generated: **allCommon** includes all Arabizi candidates generated as part of the algorithm described earlier. They are ordered by the number of tweets in which they appear. **1stCommon** is the first item from allCommon. We also evaluate a number of approaches: **K score** which is the set of all candidates ranked by the value of K (see Arabizi Keyword Selection) and **+K SW** which is the same as the K score, except that any Arabizi candidate that has fewer than K stopwords is discarded.

4.2 Results and Discussion

Our results are shown in Table 2 which reports the MRR and MAP values. As expected, allHuman performs perfectly. The allCommon run is our operational baseline. The K score results shows that ranking by overlap of stopwords improves results: MAP increases from 56.28% to 64.18%, an almost 8% absolute gain and a 14% relative improvement over allCommon. The top-ranked choice improves with MRR increasing by just over 7% absolute, or almost 11% relative. We originally hypothesized that very low stopword overlap may indicate that a word is unlikely to be Arabizi. Dropping all terms with zero overlap (+1SW) causes a large drop in MAP and a modest drop in MRR. Each successful drop of candidates lowers both scores consistently. It seems that a weak (in terms of K score) match is better than no match at all. Both K score and +1SW returned matches for all 50 queries. However, K score clearly outperforms +1SW as it

always returns relevant matches with 58% percent of the time at ranks as early as the first one. The degradation in performance is proportional to the cutoff value *K*. A close examination of the results shows that unanswered queries are experienced starting at +2SW and gradually worsens as *K* increases (Fig. 1). The K score run is the second highest run at low recall and it maintains the highest precision across all levels of recall. As expected, the Buckwalter representation does not constitute a suitable real-life Arabizi transliteration system as can be seen from Table 2.

5 Conclusion and Future Work

Our system can be seen as a module that existing search engines can integrate into their retrieval pipeline to cater for languages that are alternatively Romanized such as Arabic, Hindi, Russian, and the like. By doing so, relevant transliterated documents will be retrieved at an average rank as early as the second or first as opposed to not being retrieved at all. We plan to extend this work to handle multi-term queries, inflectional and morphological variants and attached articles and pronouns. We believe that it is fairly feasible to implement our work on other Romanizable languages given our preliminary work in other languages, in which non-linguist Arabizi users were able to cover about 80% of the mapping and conversion rules within a reasonably short amount of time (less than 30 minutes) as opposed to the creation of parallel corpora – which is far more costly and time-consuming.

Acknowledgements This work is supported by the Swiss National Science Foundation Early Postdoc.Mobility fellowship project P2NEP2_151940 and is supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

6 References

1. Chalabi, A. and Gerges, H. 2012. Romanized Arabic transliteration. In *Proceedings of the Second Workshop on Advances in Text Input Methods*, pages 89–96 (Mumbai, India, 2012). The COLING 2012 Organizing Committee.
2. Darwish, K. 2013. Arabizi detection and conversion to Arabic. arXiv:1306.6755 [cs.CL], arXiv. http://arxiv.org/abs/1306.6755.
3. Al-Badrashiny, M., Eskander, R., Habash, N., Rambow, O. 2014. Automatic Transliteration of Romanized Dialectal Arabic. In *Proceedings of the 18th Conference on Computational Language Learning* (Baltimore, Maryland USA, 2014).
4. Habash, N., Ryan, R., Owen, R., Ramy, E., and Nadt, T. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of Conference of the North American Association for Computational Linguistics (NAACL)* (Atlanta, Georgia, 2013).
5. Arfath, P., Al-Badrashiny, M., T. Diab, M., Habash, N., Pooleery, M., Rambow, O., M. Roth, R., and Altantawy, M. 2013. DIRA: Dialectal Arabic Information Retrieval Assistant. Demo Paper. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)* (Nagoya, Japan, 2013).
6. Gupta, P., Bali, P., E. Banchs, R., Choudhury, M., and Rosso, P. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR'14*. New York, NY, USA, 677-686.
7. Saha Roy, R., Choudhury, M., Majumder, P., and Agarwal, K. 2013. Overview and Datasets of FIRE'13 Track on Transliterated Search. In 5th Forum for Information Retrieval Evaluation, 2013.

8. Al-Onaizan, Y. and Knight, K. 2002. Machine Transliteration of Names in Arabic Text. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages* (2002).
9. AbdulJaleel, N. and S. Larkey, L. 2003. Statistical transliteration for English-Arabic cross language information retrieval. In *Proceedings of the 12th international conference on Information and knowledge management (CIKM '03)*. ACM (New York, NY, USA, 2003), 139-146.
10. Zhou, D., Truran, M., Brailsford, T., Wade, V., Ashman, H. Translation techniques in cross-language information retrieval. ACM Comput. Surv., 45(1):1:1–1:44, Dec. 2012.
11. May, J., Benjira, Y., Echihabi, A. An Arabizi-English Social Media Statistical Machine Translation System. In *Proceedings of the Eleventh Biennial Conference of the Association for Machine Translation in the Americas*, Vancouver, Canada (2014).
12. Bies, A., Song, Z., Maamouri, M., Grimes, S., Lee, H., Wright, J., Strassel, S., Habash, N. , Eskander, R. , Rambow, O. Transliteration of Arabizi into Arabic Orthography: Developing a Parallel Annotated Arabizi-Arabic Script SMS/Chat Corpus. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP), pages 93–103* (Doha, Qatar, 2014).

Table 1. Arabic to Arabizi mapping chart. Parenthesized letters are optional. '?' indicates an optional single character depending on the immediate subsequent character

AR	ARZ	AR	ARZ	AR	ARZ	AR	ARZ	AR	ARZ
ا	*a, 2*	خ	*5,(')7('), kh, x*	ش	*sh*	غ	*(')3('), g, gh*	ن	*n*
ب	*b*	د	*d*	ص	*9, s*	ف	*f*	ه	*h*
ت	*t*	ذ	*th, d, z*	ض	*(')9('), d*	ق	*q, g, k, a, 2, 8*	و	*w, o, u*
ث	*th, s, t*	ر	*r*	ط	*6, t*	ك	*k*	ي	*y, i, e*
ج	*j, g, ch*	ز	*z*	ظ	*(')6('), th, z*	ل	*l*	ء	*2 ([aeiou])?*
ح	*7, h*	س	*s*	ع	*3, (')?([aeiou])?*	م	*m*	ة	*t, h, a*

Table 2. K score and two baselines evaluation. * and † denote statistically significant difference with respect to allCommon and K score runs (two-tailed t-test, α=5%)

	System	MAP	MRR
Baseline	1stComm	0.1051	0.5600
	allComm	0.5628	0.6757
K Score		0.6418*	0.7487
Human	1stHuman	0.2137	1.0000
	allHuman	1.0000	1.0000
Buckwalter		0.0424*†	0.3000*†

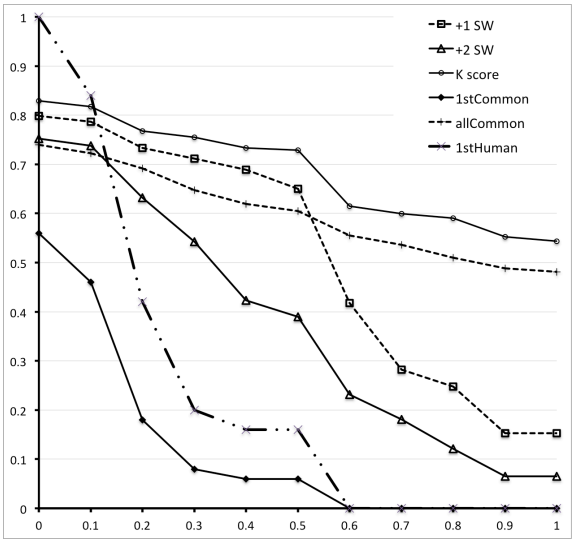

Fig. 1 Interpolated Precision-Recall curves for the K score and CSIR baselines.