An Arabizi-English Social Media Statistical Machine Translation System

Jonathan May[*] jonmay@isi.edu
USC Information Sciences Institute, Marina del Rey, CA 90292

Yassine Benjira ybenjira@sdl.com
Abdessamad Echihabi aechihabi@sdl.com
SDL Language Weaver, Los Angeles, CA 90045

Abstract

We present a machine translation engine that can translate romanized Arabic, often known as Arabizi, into English. With such a system we can, for the first time, translate the massive amounts of Arabizi that are generated every day in the social media sphere but until now have been uninterpretable by automated means. We accomplish our task by leveraging a machine translation system trained on non-Arabizi social media data and a weighted finite-state transducer-based Arabizi-to-Arabic conversion module, equipped with an Arabic character-based n-gram language model. The resulting system allows high capacity on-the-fly translation from Arabizi to English. We demonstrate via several experiments that our performance is quite close to the theoretical maximum attained by perfect deromanization of Arabizi input. This constitutes the first presentation of a high capacity end-to-end social media Arabizi-to-English translation system.

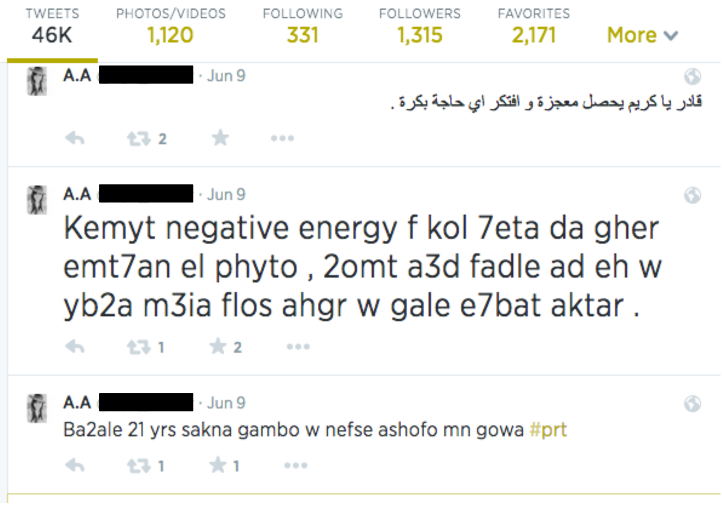
1 Introduction

Arabic-English machine translation systems generally expect Arabic input to be rendered as Arabic characters. However, a substantial amount of Arabic in the wild is rendered in Latin characters, using an informal mapping known as Romanized Arabic, Arabish, or Arabizi. Arabizi mainly differs from strict transliteration or romanization schemes such as that of Buckwalter or ALA-LC[1] in that it is not standardized. Usage is inconsistent and varies between different dialect groups and even individuals. Despite these drawbacks, Arabizi is widely used in social media contexts such as Twitter. As can be seen in Figure 1, it is not uncommon for users to use a mix of Arabic script, Arabizi, and even foreign languages such as English in their daily stream of communication.

Arabizi can be viewed as a romanization of Arabic consisting of both *transliteration* and *transcription* mappings. Transliteration is the act of converting between orthographies in a way that preserves the character sequence of the original orthography. An example of transliteration in Arabizi is the mapping of the character ع to '3' due to the similarity of the glyphs. Transcription (specifically, phonetic transcription) between orthographies is the act of converting in a way that preserves the spoken form of the original orthography as interpreted by a reader of the new orthography's presumed underlying language. An example of transcription in Arabizi is the mapping of the character ج to any of 'g', 'j', or "dj." This reflects the fact that in various

[*] This work was done while the first author was employed by SDL Language Weaver

[1] http://www.loc.gov/catdir/cpso/romanization/arabic.pdf



A.A [redacted] · Jun 9

قادر يا كريم يحصل معجزة و افتكر اي حاجة بكرة .

A.A [redacted] · Jun 9

Kemyt negative energy f kol 7eta da gher emt7an el phyto , 2omt a3d fadle ad eh w yb2a m3ia flos ahgr w gale e7bat aktar .

A.A [redacted] · Jun 9

Ba2ale 21 yrs sakna gambo w nefse ashofo mn gowa #prt

Figure 1: Examples of Arabizi mixed with Arabic and English in Twitter

dialects ج may be pronounced as [ɡ] (as in **g**od), [ʒ] (as in vi**sio**n), or [d͡ʒ] (as in **j**uice), and that the digraph “dj” is used in French for [d͡ʒ].

For a machine translation system to properly handle all textual language that can be called “Arabic,” it is essential to handle Arabizi as well as Arabic script. However, currently available machine translation systems either do not handle Arabizi, or at least do not handle it in any but the most limited of ways. In order to use any of the widely available open-source engines such as Moses (Koehn et al., 2007), cdec (Dyer et al., 2010), or Joshua (Post et al., 2013), one would need to train on a substantial corpus of parallel Arabizi-English, which is not known to exist. Microsoft’s Bing Translator does not appear to handle Arabizi at all. Google Translate only attempts to handle Arabizi when characters are manually typed, letter by letter, into a translation box (i.e. not pasted), and thus cannot be used to translate Arabizi web pages or documents, or even more than a few paragraphs at once.[2]

Because much communication is done in Arabizi, particularly in social media contexts, there is a great need to translate such communication, both for those wanting to take part in the conversations, and those wanting to monitor them. However, the straightforward approach to building an Arabizi-English machine translation system is not possible due to the lack of Arabizi-English parallel data.

In this paper we address the challenge of building such an end-to-end system, focusing on coverage of informal Egyptian communication. We find that we are able to obtain satisfactory performance by enhancing a conventionally built Arabic-to-English system with an initial Arabizi-to-Arabic deromanization module. We experiment with manually built, automatically built, and hybrid approaches. We evaluate our approaches qualitatively and quantitatively, with intrinsic and extrinsic methodologies. To our knowledge, this is the first end-to-end Arabizi-English social media translation system built.

[2] There are other online tools for rendering real-time typed Arabizi into Arabic script for use in search engines, such as Yamli (`www.yamli.com`).

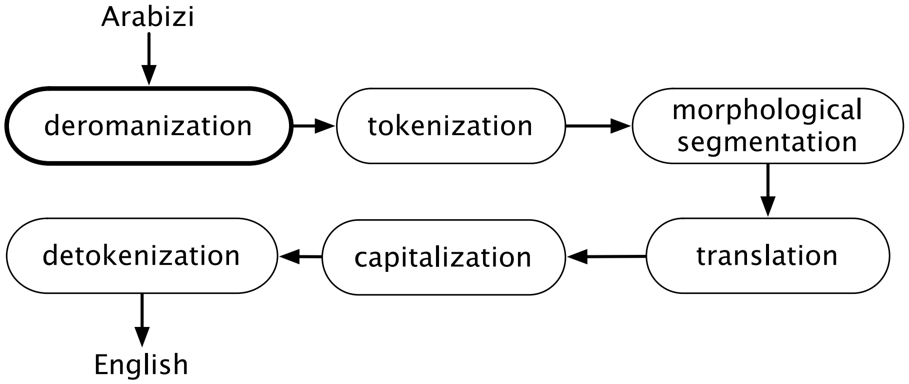


Figure 2: Schematic of our modular wFST-based machine translation system structure. The focus of this work is on the deromanization module.

2 Building an Arabizi-to-Arabic Converter

The design of our phrase-based machine translation system is modular and uses weighted finite-state transducers (wFSTs) (Mohri, 1997) to propagate information from module to module. It can thus accept a weighted lattice of possible inputs and can generate a weighted lattice of possible outputs. Our Arabizi-to-Arabic converter is one module in a pipeline that tokenizes, analyzes, translates, and re-composes data in the process of generating a translation. A schematic overview of the modules in our translation system is shown in Figure 2. An advantage of this framework is that it allows us the opportunity to propagate ambiguity through the processing pipeline so that difficult decisions may be deferred to modules with better discriminative abilities. As an example, consider the sequence "men" which could represent either the English word "men" or an Arabizi rendering of من (from). Without contextual translation of surrounding words, it is difficult to know whether the author intended to code switch to English or not. In the context of translations of surrounding words, this may be clearer, but it is inconvenient to build deromanization directly into an already complicated machine translation decoder. We find an effective solution is to persist both alternatives in the translation pipeline and ultimately let the translation module decide which input path to take. Thus the phrase "the monuments **men** film 7elw awii" (the monuments **men** very nice film) may be handled alongside the sentence "Howa nas kteer **men** el skool ray7een?" (Are there many people **from** school going?). In this work we do not consider attempts to translate code switches into languages other than the source – thus, switches into French or English, for example, would be passed to the output untranslated.

We design our converter module as a character-based wFST reweighted with a 5-gram character-based n-gram language model of Arabic. The language model is straightforwardly learned from 5.4m words of Arabic. We use a character-based language model instead of a word-based language model in order to avoid "over-correcting" out-of-vocabulary words, which are typically Arabic names. A portion of the character-based wFST is shown in Figure 3. Next we describe the strategies considered in its construction.

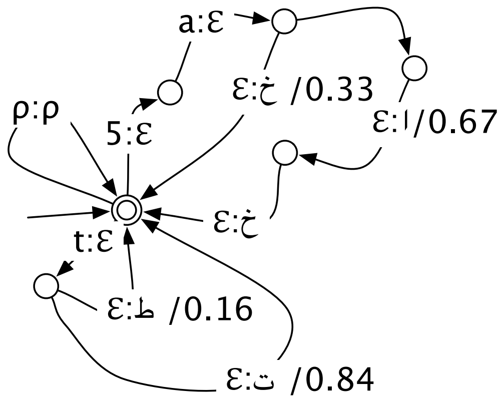


Figure 3: Portion of a wFST used to perform deromanization. This wFST represents the conditional probability of Arabic character sequences given Arabizi character sequences. In the portion shown we see that “5a” can be transformed to اخ with probability 0.67 and to خ with probability 0.33, while ‘t’ can be transformed to ت with probability 0.84 and to ط with probability 0.16. The self-loop labeled ‘*ρ*’ follows the convention of Allauzen et al. (2007) and represents all character sequences not otherwise indicated. The complete wFST has 962 states and 1550 arcs.

	Test 1	Test 2
Segments	7,794	27,901
English word tokens	51,163	168,677
‘Arabizi’ word tokens	35,208	118,857
Percent deromanizable	78.2	97.7

Table 1: Statistics of the test corpora of parallel data used for intrinsic and extrinsic evaluation. The source side of the parallel data is presumed to be Arabizi, but the percentage of deromanizable tokens (those that contain Latin characters) indicates a more heterogeneous mix comprising emoticons, Arabic characters, and other symbols.

sh	ش	1	sh	ش	0.99
			sh	سه	0.01
th	ث	0.5	th	ث	0.58
th	ذ	0.5	th	ذ	0.33
			th	ته	0.08
3	ع	1	3	ع	1
7	ح	1	7	ح	1
n	ن	1	3an	عن	0.92
			3an	عاً	0.08

Figure 4: Portion of (left) manually constructed and (right) automatically induced Arabizi-to-Arabic conditional probability table. The automatically induced table includes wider coverage not in the manual table (e.g. "th" → ته) and multi-character sequences unlikely to be thought of by an annotator (e.g. "3an" → عن).

2.1 Expert construction

As a first attempt at building an Arabizi-to-Arabic wFST, we asked a native Arabic speaker familiar with finite-state machines to generate probabilistic character sequence pairs for encoding as wFST transitions. This effort yielded a set of 83 such pairs, some of which are shown in the left side of the table in Figure 4. While these entries largely match conventional tables of Arabizi-to-Arabic mapping,[3] it is clear that even a human expert might easily construct a less-than-optimal table. For instance, while it is straight-forward for a human to choose to deterministically map the sequence "sh" to the Arabic shin (ش), this would be a bad idea. Such a choice only covers cases where "sh" is intended to convey the voiceless postalveolar fricative [ʃ] (as in **sh**ower). The same character sequence can also be used to convey an alveolar fricative followed by a glottal fricative, [sh] (as in mi**sh**ap) though, as in English, this sequence is relatively uncommon in Arabic.[4] It is hard in general for humans to estimate character sequence frequencies; our human expert gave equal weight to the voiceless and voiced deromanizations of "th," respectively, ث ([θ] as in ba**th**) and ذ ([ð] as in fa**th**er). In fact, ث is more likely in Arabic. It is also difficult and tedious to consider correspondences between sequences of more than two characters, but such context is sometimes necessary. The Arabizi character 'a' has many potential corresponding Arabic characters, and sometimes should not correspond to any character at all. But this is highly context-dependent; in the sequence "3an", for example, the 'a' represents the "short" Arabic vowel "fatha," which is not typically rendered in everyday Arabic script. Creating the correspondences that properly differentiate between long and short vowels in all proper contexts with all appropriate probabilities seems like a task that is too difficult for a human to encode.

2.2 Machine Translation-based construction

For the next attempt to build a wFST we sought inspiration in statistical machine translation system construction, which begins with the unsupervised alignment of words in hand-aligned sentences. We collected a corpus of 863 Arabizi/Arabic word pairs. We treated the word pairs

[3] http://en.wikipedia.org/wiki/Arabic_chat_alphabet

[4] After much thought, we came up with تسهيل," or "tashil" (facilitate).

Arabizi length	Arabic length	automatic count	manual count
1	0	0	**7**
1	1	**55**	51
1	2	**3**	0
2	1	**178**	25
2	2	**341**	0
2	3	**3**	0
3	1	112	0
3	2	736	0
3	3	415	0
3	4	2	0
4	1	10	0
4	2	369	0
4	3	698	0
4	4	216	0

Figure 5: Distribution of Arabizi-to-Arabic character sequence lengths in automatic and manually generated approaches to wFST building. Entries in **boldface** indicate the subsets of the automatic or manual construction that were included in the semi-automatic construction.

as sentence pairs and the characters as words, and estimated Arabizi-to-English character alignments using a standard GIZA implementation (Och and Ney, 2003) with reorderings inhibited. We then extracted character sequence pairs up to four characters in length per side that were also consistent with the character alignments, in accordance with standard practice for building phrase translation correspondence tables (Koehn et al., 2003). This resulted in a set of 3138 unique sequence pairs. We estimated conditional probabilities of Arabic given Arabizi by simple maximum likelihood. A portion of the learned table is shown on the right side of Figure 4. We can see that, in comparison to the manually constructed table on the left side of the figure, the automatically constructed table captures more—perhaps unintuitive—correspondences, and sequence pairs which provide longer context. Figure 5 compares the distribution of the lengths of the sequences learned via manual and automatic means. Note that while this automatic method learns long-context sequences, the manual annotator indicated cases of character deletion (generally of vowels) that are not learnable using this approach. However, the effects of deletion are covered via the automatic method's learning of long-context sequences where the Arabic sequence is shorter than the Arabizi sequence (see the examples for "3an" in Figure 4). Another potentially negative consequence of the automatic approach is that many useless, noisy pairs are introduced, and this can degrade quality and impact performance.

2.3 Semi-automatic construction

We sought to marry the small description length and human intelligence behind the manual approach with the empirically validated probabilities and wide coverage of the automatic approach. Consequently, after inspecting the automatically built wFST, we constructed a reduced version that only contained sequence pairs from the original if the Arabizi side had fewer than three characters (see Figure 5). We then added the vowel-dropping sequence pairs from the manual wFST.[5] This forms a hybrid of the two aforementioned constructions we call the "semi-automatic" method. While this manual intervention was feasible given the relatively small size of the automatically generated table and the availability of a native Arabic speaker, a more prin-

[5] The manual construction also includes a "w"-dropping sequence pair, which we elected not to add.

deromanization approach	BLEU	
	Test 1	Test 2
none	18.2	0.3
manual	20.1	1.7
manual + lm	21.5	2.9
automatic + lm	25.6	7.7
semi-automatic + lm	25.8	8.0

Table 2: Deromanization performance (note: **not** machine translation performance) of manually and automatically constructed modules, measured as word-based BLEU against a reference deromanization.

cipled and still automatic approach such as that taken by Johnson et al. (2007) may accomplish the same goal.

2.4 Intrinsic Evaluation

Even though our wFST-based machine translation system architecture is designed such that we can persist multiple deromanization (and non-deromanization) possibilities, it is helpful to examine the Viterbi deromanization choices of our methods, both qualitatively and quantitatively.

For quantitative evaluation, both intrinsic and extrinsic, we use two test corpora of sentence-aligned Arabizi-English social media data made available to us as part of DARPA-BOLT. Statistics of the corpora are shown in Table 1. The data also includes reference deromanizations of the Arabizi. We evaluate our deromanization approaches using the familiar BLEU metric against these reference deromanizations. The results are shown in Table 2. We see that the inclusion of a language model is helpful, and that the models influenced by corpus-based automatic learning (i.e. “automatic” and “semi-automatic”) outperform the manual model. We note, however, that the semi-automatic model, which is strongly influenced by the manual model, outperforms the automatic model slightly, and with far fewer transducer arcs.

One might expect 0 BLEU for the baseline case, where we use no deromanization method at all. This is not so due to the nature of social media data. As indicated in Table 1, many non-Arabizi tokens, such as emoticons, URLs, Arabic words, and English code switches, occur throughout the data, often mixed into predominantly Arabizi segments. The Test 1 corpus contains a significantly larger percentage of such tokens than the Test 2 corpus.

One might also expect higher overall BLEU scores at the bottom of Table 2, given the general track record of transliteration performance (Darwish, 2013; Al-Onaizan and Knight, 2002). We note that dialectical Arabic is in general not a written language, and as such there are many different spellings for words, even when rendered in Arabic script. Thus the task is closer to machine translation than classic transliteration (in that “correctness” is a squishy notion). Additionally, we did not specifically optimize our deromanizer for this intrinsic experiment, where we must decide whether or not to deromanize a possibly non-Arabizi word. Choosing incorrectly penalizes us here but should not impact extrinsic MT performance (evaluated in Section 4), due to our pipeline architecture’s ability to present both deromanized and non-deromanized options to downstream modules (see discussion in Section 2).

For some qualitative analysis, we consider an example comparison between our various deromanizer approaches in Figure 6. We observe the following:

- The Arabizi sentence starts with the chat acronym “isa,” which is expandable to إن شاء الله, “in sha allah” (God willing). The manual wFST outputs “sa” while the automatic wFST outputs “issa.” Both are expected to be wrong, since acronyms are not handled in the current approach.

derom approach	derom output	machine translation output
none	isa akher elesbo3 ele gay	god akher elesbo3 fear gay
manual	ٮسا اكهر لسبوع ل جاى	asa elesbo akher3 for coming
manual + lm	سا اخر الاسبوع الا جاى	sa at the end of the week, but is coming
[semi-]automatic + lm	يصا اخر الاسبوع الي جاي	god at the end of the week to come.
reference	إن شاء الله آخر الأسبوع اللي جاي	god willing, the end of next week.

Figure 6: Effect of various deromanizers on an Arabizi sentence and the effect of deromanization on translation. The semi-automatic and automatic deromanizers give the same result for this sentence. The reference translation is “god willing by the end of next week.”

- The second word, “akher,” is deromanized correctly by the manual and automatic wFSTs. Since the deromanizer was developed for an MT engine, the maddah diacritic (ˇ) that extends the sound of the alif (ا) is normalized (i.e. removed). We note the same type of discrepancy for the “el” in “elesbo3” where the hamza (ء) over the alif (ا) is normalized in both wFSTs.
- For the fourth word, “ele,” the automatic wFST rightly corrects the manual wFST’s use of alif (ا) into a ya’ (ي), which is equivalent to replacing the vowel ‘a’ by ‘i’ in English. Although, as the reference deromanization shows, both wFSTs miss out on the additional letter lam (ل), which represents the alveolar lateral approximant [l] (as in **l**iquid), somewhat predictably, since “ele” can be seen as a “misspelling.”
- For the final word, “gay,” the automatic wFST again rightly corrects the manual wFST by turning the broken alif (ى) into a ya (ي), which is also the equivalent of replacing the English vowel ‘a’ by ‘i’.

3 System Description

As illustrated in Figure 2, our deromanization module is one component in a pipeline of processing that forms a machine translation system. Aside from the deromanization module, which we vary in the following experiments, our system is constant and built as follows: The preprocessing additionally consists of a regular expression-based tokenization and normalization module to separate punctuation, and a word morphological segmentation module based on the type-based unsupervised approach of Lee et al. (2011). The machine translation module is phrase based, in the style of Koehn et al. (2003), and is trained on informal Arabic-English parallel and monolingual data made available through DARPA BOLT. The post-processing consists of deterministic detokenization based on the output word sequence. The capitalizer is part of our pipeline, as noted in Figure 2, but since we do not evaluate cased translations it was turned off for these experiments.

4 Extrinsic Experiments

In Table 3 we show the results of evaluating our informal Arabic-English MT system on the two aforementioned test sets while equipped with various configurations of the deromanization module. We also evaluate, as an upper bound, performance using a system with no deromanization module, but with a reference deromanization as input. We report detokenized, case-free

deromanization approach	BLEU	
	Test 1	Test 2
none	7.7	3.7
manual	9.6	5.8
manual + lm	12.0	8.9
automatic + lm	15.1	13.2
semi-automatic + lm	15.3	13.4
reference deromanization	18.4	17.9

Table 3: Comparison of end-to-end MT performance using a deromanization module and Arabic-English system to translate Arabizi-English. The automatically learned wFST approach outperforms the manual wFST and makes good progress toward the reference deromanization upper bound. Scores reported are detokenized, lowercased BLEU.

BLEU. The scores in Table 3 track those in Table 2, indicating a strong correlation between deromanizer performance and translation performance.

Turning to the qualitative results in Figure 6, we note the following:

- Although the non-deromanized system mostly passes the input through unchanged, we produce the words "god" and "fear." The latter is likely an error due to a spurious low-count alignment of "ele" to "fear" in training data, but the former is due to a correspondence with "isa," which, as previously noted, is shorthand for "god willing." This is indicative of small amounts of Arabizi appearing in our training data.
- In the manual-based cases, the incorrect deromanization of "isa" leads to an unknown Arabic word being selected and then transliterated back into English, producing "asa" or "sa." In the automatic-based cases the same could have happened, but the decoder instead chose to use the non-deromanized alternative and produced "god" as in the baseline case. Naturally, the reference deromanization, which correctly expands the acronym, leads to the best translation of this token.
- Since our MT engine normalizes away ligatures, the substantial differences between our deromanization approaches and the reference deromanization due to ligature placement results in little tangible effect on translation performance. This accounts for the correct translations of "at the end of the week."
- The deromanizers' inabilities to properly include the additional lam in "ele" accounts for the erroneous translation of "ele" as "to."

5 Related Work

After this work had been substantially completed, we became aware of a similar effort by Al-Badrashiny et al. (2014). That effort, which resulted in the "3arrib" standalone deromanizer for Egyptian Arabic, also uses a wFST-based approach but verifies suitability using a hand-crafted Arabic morphological analyzer. Additionally, an effort was made in 3arrib to handle 32 special cases such as the expansion of "isa." We compare their work to ours in Table 4. It should be noted that the 3arrib system was used in the preparation of the Test 1 and Test 2 data. That is, the initially collected Arabizi data was run through 3arrib, then post-edited by annotators. The intrinsic empirical results in particular should thus be taken with a grain of salt.

			Test 1		Test 2	
deromanization approach	deromanization	translation	Transliteration BLEU	Translation BLEU	Transliteration BLEU	Translation BLEU
none	isa akher elesbo3 ele gay	god akher elesbo3 fear gay	18.2	7.7	0.3	3.7
semi-automatic + lm	يصا اخر الاسبوع الي جاي	god at the end of the week to come.	25.8	15.3	8.0	13.4
3arrib (Al-Badrashiny et al., 2014)	إن شاء الله آخر الاسبوع الجاي	god willing, the end of the coming week.	56.0	15.7	51.0	14.8
reference deromanization	إن شاء الله آخر الأسبوع اللي جاي	god willing, the end of next week.	100	18.4	100	17.9

Table 4: A comparison of qualitative and quantitative, extrinsic and intrinsic results using the deromanization method described in the current work and that of Al-Badrashiny et al. (2014). While our work does not employ deep expert knowledge such as a hand-built morphological analyzer or special case handling such as acronym expansion, we are nonetheless able to build a system with comparable extrinsic performance to the boutique system of Al-Badrashiny et al. (2014). Note too that the intrinsic scores of Al-Badrashiny et al. (2014) reflect the fact that the gold data for this task was constructed by post-editing 3arrib output.

Darwish (2013) addresses many of the problems tackled in this work, though not in the context of machine translation. Like our baseline experiments, that work uses a hand-constructed transliteration table to map between Latin and Arabic sequences. Darwish (2013) places particular emphasis on detecting the difference between Arabizi and non-Arabizi words, and not attempting to deromanize the latter. He trains a conditional random field (CRF) to identify Arabizi words and reports accuracy of 98.5%. Since we have the luxury of downstream modules that can take ambiguous input (see Section 2), we simply allow each word to be transliterated or not and allowed nonsense deromanizations of non-Arabic words to be ignored by the translation engine in lieu of handling the original word. While the CRF approach is an appealing one that we will consider, we note that by not making a firm decision we allow words that are ambiguously Arabizi or English to be discriminated by a system that contains the rich context necessary for translation.

Chalabi and Gerges (2012) present an approach to Arabizi transliteration and mention the applicability of this functionality to improving machine translation but do not specify the approach taken in great detail.

Irvine et al. (2012) perform deromanization of Urdu as part of an overall normalization task for cleaning Urdu text messages. Their approach to building a subsequence correspondence table, which is described in Irvine et al. (2010), is similar to ours, though their training data does not include Arabizi.

Al-Onaizan and Knight (2002) use a cascade of wFSTs to attack the converse problem, that is, romanizing names from Arabic script into English.

This work can be considered a special case of handling user-generated content, as opposed to more formal content such as that from news or government sources. Others who have focused on handling user-generated content for machine translation include Jiang et al. (2012); Pennell and Liu (2011) and Carter et al. (2011). We took a comparatively simple approach to special cases such as URLs, emoticons, and hash tags, by using regular expressions to avoid translating untranslatable entities or splitting up special formatting.

6 Conclusion

Translation systems that can cope with the realities of informal communication need to be built with an understanding of the cultural forces that shape the way communication happens. In this work, we explored the consequences of societies wishing to communicate with a language that is not normally written in Latin characters (or, indeed, written at all) but being constrained to the Latin character set for historical, technological, or perhaps arbitrary reasons. These limitations prove no real barrier to infinitely creative humans but can confound computer systems built with regular assumptions in mind. We have shown that adapting our systems to match real-world behavior is not difficult, but requires an awareness of the forces at play.

Acknowledgement

We thank Steve DeNeefe, Dragos Munteanu, Thomas Wieland, Eli Pak, Amos Kariuki, and Markus Dreyer for their contributions to this work. We thank Aliya Deri, Marjan Ghazvininejad, Nima Pourdamghani, Hui Zhang, Kevin Knight, and Daniel Marcu for providing useful feedback. This work was supported by TSWG contract N41756-13-C-3050 and by DARPA contract HR0011-12-C0014.

References

Al-Badrashiny, M., Eskander, R., Habash, N., and Rambow, O. (2014). Automatic transliteration of romanized dialectal Arabic. In *Proceedings of the Eighteenth Conference on Com-*

putational Natural Language Learning, pages 30–38, Ann Arbor, Michigan. Association for Computational Linguistics.

Al-Onaizan, Y. and Knight, K. (2002). Machine transliteration of names in Arabic texts. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. `http://www.openfst.org`.

Carter, S., Tsagkias, M., and Weerkamp, W. (2011). Twitter hashtags: Joint translation and clustering. In *Proceedings of the 3rd International Conference on Web Science*.

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. The COLING 2012 Organizing Committee.

Darwish, K. (2013). Arabizi detection and conversion to Arabic. arXiv:1306.6755 [cs.CL], arXiv. `http://arxiv.org/abs/1306.6755`.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden. Association for Computational Linguistics.

Irvine, A., Callison-Burch, C., and Klementiev, A. (2010). Transliterating from all languages. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, Denver, Colorado, USA.

Irvine, A., Weese, J., and Callison-Burch, C. (2012). Processing informal, romanized Pakistani text messages. In *Proceedings of the Second Workshop on Language in Social Media*, pages 75–78, Montréal, Canada. Association for Computational Linguistics.

Jiang, J., Way, A., and Haque, R. (2012). Translating user-generated content in the social networking space. In *Proceedings of AMTA-2012, the Tenth Biennial Conference of the Association for Machine Translation in the Americas*.

Johnson, H., Martin, J., Foster, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lee, Y. K., Haghighi, A., and Barzilay, R. (2011). Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9, Portland, Oregon, USA. Association for Computational Linguistics.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Pennell, D. and Liu, Y. (2011). A character-level machine translation approach for normalization of sms abbreviations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 974–982, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Post, M., Ganitkevitch, J., Orland, L., Weese, J., Cao, Y., and Callison-Burch, C. (2013). Joshua 5.0: Sparser, better, faster, server. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 206–212, Sofia, Bulgaria. Association for Computational Linguistics.