A Simple but Effective Approach to Improve Arabizi-to-English Statistical Machine Translation

Marlies van der Wees **Arianna Bisazza** **Christof Monz**
Informatics Institute, University of Amsterdam
{m.e.vanderwees,a.bisazza,c.monz}@uva.nl

Abstract

A major challenge for statistical machine translation (SMT) of Arabic-to-English user-generated text is the prevalence of text written in *Arabizi*, or Romanized Arabic. When facing such texts, a translation system trained on conventional Arabic-English data will suffer from extremely low model coverage. In addition, Arabizi is not regulated by any official standardization and therefore highly ambiguous, which prevents rule-based approaches from achieving good translation results. In this paper, we improve Arabizi-to-English machine translation by presenting a simple but effective Arabizi-to-Arabic transliteration pipeline that does not require knowledge by experts or native Arabic speakers. We incorporate this pipeline into a phrase-based SMT system, and show that translation quality after automatically transliterating Arabizi to Arabic yields results that are comparable to those achieved after human transliteration.

1 Introduction

Almost all current state-of-the-art statistical machine translation (SMT) systems for Arabic-to-English translation are trained on data comprising Modern Standard Arabic (MSA). MSA is widely used by professional publishers, such as news agencies, government agencies, and non-governmental organisations (NGOs). On the other hand, in user-generated content, such as weblogs, Internet forums, and short text messages, MSA is substantially less prevalent. Here one can often encounter dialectal variations resulting in a slightly different vocabulary and morphological constructions.

In addition to dialectal variations, user-generated content often also contains Arabic that is not written in the Arabic script, but in Romanized form, typically referred to as *Arabizi*. This is not to be confused with standardized research transliteration schemes such as the Buckwalter encoding for Arabic, or an official phonetic system such as Pinyin for Chinese. Instead, Arabizi encountered in user-generated text emerged from practical limitations such as the lack of an Arabic keyboard.

While Arabizi is not regulated by any standardization and many-to-many Arabizi-Arabic character mappings are ubiquitous, certain conventions have emerged. These conventions mostly rely on reflecting phonetic approximations by using a combination of numbers and Latin letters. Since Arabizi is mostly guided by pronunciation, it is also very sensitive to dialectal variations, which are more noticeable in spoken than in written Arabic. Note that some Arabizi representations are also based on orthographic similarities, such as '3' for the Arabic letter ع. Table 1 shows an example sentence in Arabizi, along with its MSA transliteration, Buckwalter transliteration and English translation, and illustrates the difference between Arabizi and formal transliteration.

Since Arabizi is highly ambiguous and difficult to transliterate with rule-based approaches, there is an extreme scarcity of gold standard transliterated Arabizi, making it a challenging task to develop data-driven statistical approaches involving Arabizi, such as Arabizi-to-English machine translation. Moreover, the standard data sets used by the MT research community do not contain Arabizi, meaning that any attempt to translate Arabic in Arabizi writing will suffer from extremely high (close to 100%) out-of-vocabulary (OOV) rates.

In this paper, we use a handful of small resources to build a simple but effective Arabizi-to-Arabic

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 43–50, Osaka, Japan, December 11 2016.

Arabizi (lowercased)	la2 laa m7adsh by7eb el ka7k ela pappi :(w howa mesh henaaa
Arabic (human)	لا لا محدش بيحب ال كحك الا بابا : (و هو مش هنا
Arabic (Buckwalter)	lA lA mHd$ byHb Al kHk AlA bAbA :(w hw m$ hnA
English (human)	No, no one likes cookies except my father :(and he's not here

Table 1: Example sentence in Arabizi, along with its human Arabic transliteration, the corresponding Buckwalter transliteration, and its human English translation.

transliteration pipeline, which we incorporate into a state-of-the-art phrase-based SMT system. Concretely, our contributions are as follows:

(i) We present and release an Arabizi-to-Arabic transliteration pipeline that combines character-level mapping with contextual disambiguation of Arabic candidate words. We improve transliteration candidate selection by incorporating common Arabizi-Arabic word pairs. We evaluate our end-to-end Arabizi-English translation system using two test sets, and show that translation quality after automatically transliterating Arabizi to Arabic yields results that are comparable to those achieved after human transliteration.

(ii) We collect and release a web-crawled Arabizi-English parallel corpus of approximately 10,000 sentence pairs. Despite being too small to train a fully data-driven translation system, this corpus is useful for words that cannot be transliterated successfully by our transliteration pipeline.

2 Data sets and resources

For our Arabizi-to-English translation approach we use a number of data sources: First, in order to transliterate Arabizi to Arabic *words*, we start with an Arabizi to Arabic *character* mapping. Several such mappings can be found online, but the most comprehensive one we could find was the one from Wikipedia, see Table 2. We use this resource in Section 3.1 to generate transliteration candidates.

Arabic	Arabizi	Arabic	Arabizi	Arabic	Arabizi
آ, ئ, إ, ؤ, أ, ء	2	ز	z	ك	k , g
ا	a , e	س	s	ل	l
ب	b , p	ش	sh , ch	م	m
ت	t	ص	s , 9	ن	n
ث	s , th	ض	d , 9’	ه	h , a , e , ah , eh
ج	g , j , dj	ط	t , 6	ة	a , e , ah , eh
ح	7	ظ	z , dh , t’ , 6’	و	w , o , u , ou , oo
خ	kh , 7’ , 5	ع	3	ى , ي	y , i , ee , ei , ai , a
د	d	غ	gh , 3’	پ	p
ذ	z , dh , th	ف	f , v	چ	j , tsh , ch , tch
ر	r	ق	2 , g , q , 8 , 9		

Table 2: Mapping of Arabic letters to Arabizi character sequences. Source: Wikipedia. (`http://en.wikipedia.org/wiki/Arabic_chat_alphabet`).

Next, we use a large Arabic-English parallel corpus containing text in several genres. Since Arabizi is characteristic to user-generated text, we have included as much informal, user-generated parallel data as

available. The resulting bitext contains 1.75M sentence pairs and 52.9M Arabic tokens, and comprises approximately 70% news data, mostly LDC-distributed, and 30% data in various other genres (weblogs, comments, editorials, speech transcripts, and small amounts of chat data), mostly harvested from the web. We use this corpus to (i) generate an Arabic vocabulary that guides transliteration candidate selection (Section 3.2), and (ii) to build our main Arabic-English SMT system in Section 4.

Next, we use a small data set released for the most recent NIST OpenMT evaluation campaign[1], containing approximately 10,000 triplets of manually transliterated and translated Arabizi/Arabic/English sentences belonging to the SMS and chat genres. Despite its small size, this 'tritext' contains information of very high quality that we exploit to improve our Arabizi-to-Arabic transliteration approach in Section 3.4. In addition, we extract from this corpus 1,788 Arabizi-Arabic triplets which we split into two test sets to test our pipeline in Sections 3.3 and 3.4, and to evaluate downstream Arabizi-to-English SMT performance in Section 4.

Finally, we use an Arabizi-English bitext crawled from a variety of web pages, containing user-generated comments to news articles which where originally written in Arabizi and translated into English by professional translators. This corpus contains approximately 10K sentence pairs and 180K Arabizi tokens. We use this bitext in Section 4 as part of our end-to-end SMT pipeline. While being too small to train an end-to-end SMT system, we believe that this corpus is a useful resource for researchers working with Arabizi, and we make the bitext available for download.[2]

3 Arabizi-to-Arabic Transliteration

In this section we describe our efforts to convert Arabizi words to Arabic words, which can then be translated into English using our MSA-to-English phrase-based SMT system. The complete transliteration pipeline, which we make available for download[2], is illustrated in Figure 1. The pipeline's methodological components (white boxes) are described in detail in Sections 3.1–3.4.

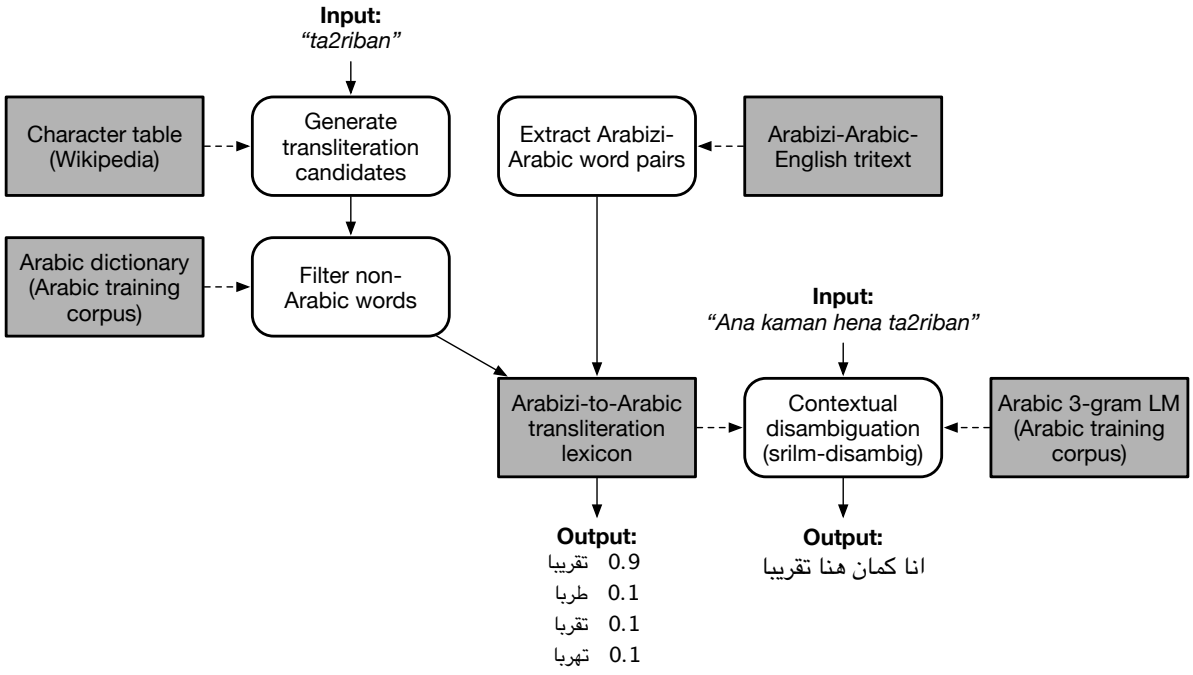

Figure 1: Arabizi-to-Arabic transliteration pipeline. White boxes represent methodological components, and grey boxes represent data components.

[1]LDC catalog number: LDC2013E125

[2]http://ilps.science.uva.nl/resources/arabizi/

3.1 Generating transliteration candidates

Table 2 shows that Arabizi-to-Arabic character mappings are often ambiguous, and in many cases Arabic script letters are represented by a sequence of two characters. This in turn introduces segmentation ambiguity as two individual Arabizi characters can be mapped separately to a sequence of two Arabic letters or the two Arabizi characters can be mapped together to a single Arabic script letter. This type of mapping problem is very reminiscent of the phrase-based SMT task assuming that Arabizi characters correspond to source words and sequences of Arabizi characters correspond to source word phrases.

Given this similarity we first cast the Arabizi to Arabic transliteration problem as a machine translation problem. The phrase table, i.e., translation model, consists of all possible character and character sequence mappings. Standard statistical machine translation systems use a word-based language model over target language sequences to estimate the fluency of translation hypotheses. Here, we use an Arabic-character based language model instead. Decoding is carried out in the same manner as the normal translation setting, except that the distortion limit is set to 0, enforcing monotone decoding. Note that while we opt for using a publicly available character table, it would also be possible to learn a character mapping and its corresponding probabilities from an Arabizi-Arabic bitext. This is done for example in related work by May et al. (2014).

An important problem at this point is that vowel mappings result in Arabic words having too many vowels orthographically present. This is particularly problematic for Arabizi words with repetitive vowels, a common phenomenon in user-generated text, such as observed in the word *henaaa* in Table 1. In order to address this problem, we allow for more flexible character mappings of vowels in which Arabizi vowels can be dropped. As a result, transliteration candidates for *henaaa* also include candidates for *hena*, among which the correct Arabic word هنا.

3.2 Filtering non-Arabic words

Using the described character mapping approach, we exhaustively consider all possible mappings, and therefore deliberately over-generate the number of Arabic word candidates. This can result in dozens and sometimes hundreds of Arabic character sequences, the vast majority of which are character sequences that are possible in Arabic, at least according to the 5-gram Arabic character language model, but are not actual words. To filter out these character sequences, all candidates are compared to a large Arabic vocabulary, and all candidates not occurring in the vocabulary are eliminated from further processing. The vocabulary contains 200K unique words, and is constructed from the Arabic side of all parallel corpora that we use for our SMT experiments in Section 4. Note that using this vocabulary can cause potentially correct Arabic words to be removed from the transliteration candidate list, as the vocabulary only covers the bitext. However, for the task of Arabizi-to-English translation this does not affect the final outcome as our SMT system can only predict English translations for Arabic words occurring in the bitext. For other downstream applications, a different or larger Arabic vocabulary can be used.

Restricting Arabic candidates to words occurring in the vocabulary reduces the number of candidates for a given Arabizi word considerably to approximately 5 to 10, and excludes transliteration candidates for Arabizi words with character repetitions, such as *henaaa* in Table 1.

3.3 Contextual disambiguation

The character mapping and filtering process results in an ambiguous Arabizi-to-Arabic transliteration lexicon. We feed this lexicon to srilm-disambig[3], a publicly available tool that searches for the best transliteration of each Arabizi sentence according to a 3-gram Arabic language model trained on the source side of the available parallel Arabic-English corpora.

Evaluated on the two Arabizi-Arabic test sets described in Section 2, our pipeline up to this point achieves a word-level transliteration error rate of about 50%, see top row in Table 3. Note that word-level error rate is much harsher than character-level error rate, but it gives us a better estimate of how

[3] http://www.speech.sri.com/projects/srilm/manpages/disambig.1.html

Transliteration method	Test set 1	Test set 2
Char.map.+disambig	46.4%	50.8%
Char.map.+disambig+word pairs	25.7%	27.9%

Table 3: Word-level transliteration error rates for two variants of our transliteration pipeline measured on two held-out test sets.

much noise will be propagated into the SMT system. With half of the words being wrongly transliterated, we can expect a very poor SMT quality at the end of the pipeline.

3.4 Improving transliteration using Arabizi-Arabic word pairs

Next, to reduce the error rate we exploit transliterated *word pairs* extracted from the Arabizi and Arabic sides of our tritext. These can be almost perfectly aligned at the word level (about 6% of the sentence pairs had a mismatching number of words and were discarded), yielding high-precision transliteration candidates. We add these word pairs to the transliteration lexicon used by srilm-disambig, giving them a high score (0.9 versus 0.1 for the other transliteration candidates) so that they will be preferred most of the time over the candidates generated by the character-level SMT system (see bottom of Figure 1). Adding this step to our pipeline results in a decrease of the word-level transliteration error rate from 50% to about 25% on the test sets, see bottom row in Table 3. From a manual inspection of a data sample, we find that a large part of the remaining transliteration errors are due to different possible spellings of English words in Arabic (e.g. *baby* / بابي or بيبي) or different spellings of Arabic dialectal forms (e.g. *when* / امته or امتى), which reveals the difficulty of evaluating Arabizi transliteration.

4 Arabizi-to-English machine translation

We examine the success of the proposed transliteration approaches by running Arabic-to-English SMT experiments, which we describe in Section 4.1 and discuss in Section 4.2.

4.1 Experimental setup

We perform our translation experiments using an in-house state-of-the-art phrase-based SMT system similar to Moses (Koehn et al., 2007). The system is trained on the collection of Arabic-English parallel corpora discussed in Section 2, comprising 1.75M lines (52.9M Arabic tokens) of parallel text. In addition, we use a 5-gram English language model that linearly interpolates different English Gigaword subcorpora with the English side of our bitext.

When no valid Arabic transliteration is found for an Arabizi word, our software component leaves it unchanged. To increase the chances of handling such cases, we exploit our in-house Arabizi-English corpus of web-crawled user comments (see Section 2), on which we train a separate Arabizi-English system. Instead of using this system for the actual translation task, which would suffer from very low coverage, we merge the Arabizi-English phrase translation and phrase reordering models to the main Arabic-English models using a fillup technique (Bisazza et al., 2011). In this way, a non-transliterated word that is not matched by the main Arabic-English models has still a chance of being translated directly by the Arabizi-English models.

We tokenize all Arabic data—training data as well as transliterated Arabizi—using the MADA toolkit (Habash and Rambow, 2005).

4.2 Results

Table 4 shows SMT quality measured with case-insensitive BLEU (Papineni et al., 2002) for a number of transliteration scenarios. First, we see that Arabizi-to-English translation without any preprocessing (top row) results in very poor translation quality. There is, however, a large difference in BLEU between the two test sets, with test set 2 achieving a surprisingly high score given that almost the entire source text is

	Test set 1		Test set 2	
Preprocessing scheme	BLEU	% of gold standard	BLEU	% of gold standard
Original Arabizi	1.30	(14.4%)	4.99	(50.5%)
Char.map.+disambig	7.46	(82.6%)	9.42	(95.2%)
Char.map.+disambig+word pairs	8.68	(96.1%)	10.32	(104.3%)
Human transliteration	9.03	(100%)	9.89	(100%)

Table 4: BLEU scores of Arabizi-to-English translation experiments using different preprocessing schemes: no transliteration, two variants of our automatic transliteration pipeline, and human transliteration, respectively. The latter can be considered as a gold standard, so we also present the results relative to what is achieved by human transliteration.

unseen vocabulary. This result can be explained by the large portion of English words and emoticons in test set 2 with respect to test set 1: 25.7% of the source tokens in test set 2 match a target word in their corresponding reference sentence, as opposed to only 16.7% matching tokens in test set 1. English words in the source text are in most cases unknown to the SMT system but will not harm output quality if copied over to the output verbatim. See Table 5 for an example from test set 2 illustrating this observation.

Second, comparison of the two proposed transliteration variants (second and third row) shows that the approach that makes use of Arabizi-Arabic word pairs (third row) achieves better performance than the variant without word-pair information (second row). However, despite the large difference in word-level transliteration error rate between both variants (Table 3), BLEU differences are fairly small.

Finally, we compare our automatic transliteration approaches to human Arabizi-to-Arabic transliteration (bottom row) and notice that BLEU scores of our best automatic method are comparable to those achieved with human transliteration. Interestingly, for test set 2 we achieve higher BLEU scores with the new transliteration approach than with human transliterations. This observation can be explained by the prevalence of English words in the Arabizi source text. We observe numerous sentences where English words in the original Arabizi text were transliterated to Arabic in the human transliteration, leading to phrases that are unknown by the SMT system. In contrast, our transliteration pipeline can in most cases not find a valid Arabic word when transliterating an English word, and hence leaves the English word unchanged, which in turn may lead to correct output in the final translation. See Table 5 for an example that illustrates this scenario.

Preprocessing scheme	SMT input	SMT output
Original Arabizi	mashyyyy okay did you have fun	okay mashyyyy did you have fun
Char.map.+disambig	did you have fun ماشيي كي	in order to did you have fun
Char.map.+disambig+word pairs	did you have fun ماشي اوكي	ok, ok, did you have fun
Human transliteration	ماشي أوكي ضد يا هاف ف ان	ok, ok, against,,

Table 5: SMT input-output example pairs for a sentence containing English words in the original Arabizi text. English reference translation: **fine, okay, did you have fun?**.

5 Related work

In the past years, a few approaches to transliterate Arabizi to Arabic (or other tasks of *deromanizing* text in non-native Romanized script (Irvine et al., 2012)) have been presented, most of which rely to at least some extent on knowledge of experts or native speakers. In contrast, the approach we have presented does not rely on expert knowledge and is constructed using only publicly available data sources.

Chalabi and Gerges (2012) present a transliteration engine that, like our approach, follows the SMT paradigm. However, they complement their method with handcrafted rules.

A similar approach by Darwish (2014) focuses on Arabizi detection as well as conversion to Arabic. The former is important when Arabizi text is alternated with words from other languages, such as French or English, which it regularly is. While our approach leaves unconvertible words unchanged and often yields the right SMT output for English source words, addressing the problem of language detection before transliteration will likely benefit our approach.

Bies et al. (2014) present their work on manually transliterating Arabizi SMS and chat messages to Arabic. Their work is focused on releasing a new resource rather than presenting a transliteration methodology, and naturally yields high-quality transliteration.

Al-Badrashiny et al. (2014) use a weighted finite-state transducer (wFST) approach to converting Arabizi to Arabic in their system "3arrib". They incorporate linguistic information by using CODA, a conventional orthography for Dialectal Arabic (Habash et al., 2012) and morphological analysis, and thus heavily rely on expert knowledge.

All of the above focus on Arabizi-to-Arabic conversion outside the context of SMT from Arabizi. The work by May et al. (2014) is the only that presents an Arabizi-to-English SMT system, in which the authors not only focus on transliterating Arabizi to Arabic, but also evaluate performance in end-to-end Arabizi-to-English SMT experiments. Their transliteration approach uses wFSTs which are constructed by (i) experts, (ii) machine translation, or (iii) semi-automatically. In downstream SMT experiments, the semi-automatic construction performs best but depends partially on expert knowledge. The version of their approach in which wFSTs are constructed fully automatically is very similar to our approach, with a few main differences: While we start using a character mapping with uniform weights, they learn weights from an Arabizi-Arabic bitext. Next, they select the most probable transliteration candidates using Viterbi paths while we use srilm-disambig. Finally, we use Arabizi-Arabic parallel data to guide candidate selection, and Arabizi-English parallel data to enhance our SMT system. Unfortunately, the system of May et al. (2014) is not publicly available, making it impossible to compare performances.

Besides the described work, a few commercial systems for Arabizi conversion exist: Google Ta3reeb, Microsoft Maren, and Yamli. These are, however, not suitable for batch translation as is common in SMT research. Moreover, their approaches have not been published in the research community.

6 Conclusions

A major challenge for SMT of Arabic-to-English user-generated text is the prevalence of text written in *Arabizi*, or Romanized Arabic, which is typically not covered in the SMT models. In this paper we have presented our work on translating Arabizi into English by first transliterating Arabizi into Arabic using an approach that does not require knowledge of experts or native Arabic speakers.

Our transliteration pipeline uses character mapping following the phrase-based SMT paradigm, supplemented with vocabulary-based filtering and contextual disambiguation of candidate Arabic words. In addition, the availability of a small Arabizi-Arabic-English tritext allows us to (i) further improve the transliteration pipeline by prioritizing transliteration options that are supported by Arabizi-Arabic word pairs in the tritext, and (ii) evaluate our method in terms of transliteration error rates and in SMT experiments.

The transliteration pipeline exploiting Arabizi-Arabic word pairs yields considerably lower word-level transliteration error rates, dropping from approximately 50% for the simpler variant without word-pair information to 25% for the extended approach. When evaluating our approach in SMT experiments with two held-out test sets, we see that BLEU scores of the two variants reflect this large difference in error rate only to a limited extent. Furthermore, we have shown that translation quality after automatically transliterating Arabizi to Arabic yields results that are comparable to those achieved after human transliteration.

Finally, we make available for download both the transliteration pipeline software and a web-crawled Arabizi-English bitext of approximately 10,000 sentences.

Acknowledgements

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project number 639.022.213. We thank the anonymous reviewers for their thoughtful comments.

References

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic transliteration of romanized dialectal Arabic. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rambow. 2014. 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.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proceedings of the 8th International Workshop on Spoken Language Translation*, pages 136–143, San Fransisco, California, December.

Achraf Chalabi and Hany Gerges. 2012. Romanized Arabic transliteration. In *Proceedings of the Second Workshop on Advances in Text Input Methods (WTIM 2)*, pages 89–96.

Kareem Darwish. 2014. Arabizi detection and conversion to Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 573–580.

Nizar Habash, Mona T Diab, and Owen Rambow. 2012. Conventional orthography for dialectal Arabic. In *LREC*, pages 711–718.

Ann Irvine, Jonathan Weese, and Chris Callison-Burch. 2012. Processing informal, romanized Pakistani text messages. In *Proceedings of the Second Workshop on Language in Social Media*, pages 75–78.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Jonathan May, Yassine Benjira, and Abdessamad Echihabi. 2014. An Arabizi-English social media statistical machine translation system. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pages 329–341, Vancouver, Canada, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.