The QCN Egyptian Arabic to English Statistical Machine Translation System for NIST OpenMT'2015

Hassan Sajjad, Nadir Durrani, Francisco Guzman, Preslav Nakov,
Ahmed Abdelali, Stephan Vogel,
†Wael Salloum, †Ahmed El Kholy, ‡Nizar Habash
Qatar Computing Research Institute
†Columbia University, ‡New York University Abu Dhabi

Abstract

The paper describes the Egyptian Arabic-to-English statistical machine translation (SMT) system that the QCRI-Columbia-NYUAD (QCN) group submitted to the NIST OpenMT'2015 competition. The competition focused on informal dialectal Arabic, as used in SMS, chat, and speech. Thus, our efforts focused on processing and standardizing Arabic, e.g., using tools such as *3arrib* and *MADAMIRA*. We further trained a phrase-based SMT system using state-of-the-art features and components such as operation sequence model, class-based language model, sparse features, neural network joint model, genre-based hierarchically-interpolated language model, unsupervised transliteration mining, phrase-table merging, and hypothesis combination. Our system ranked second on all three genres.

1 Introduction

We describe the Egyptian Arabic-to-English statistical machine translation (SMT) system of the QCN team for the NIST OpenMT'2015 evaluation campaign. The QCN team included the Qatar Computing Research Institute, Columbia University, and New York University in Abu Dhabi.

The OpenMT 2015 translation task asked participants to build systems that can translate Egyptian Arabic from three different genres (SMS, chat, and speech) into English. The challenges presented by this multigenre task were many, ranging from scarcity of parallel data for training, to noisy source text and heterogeneity of the references. For example, large portions of the provided data consisted of romanized Egyptian Arabic (aka *Arabizi*) rather than using the Arabic script.

Therefore, several preprocessing steps were needed in order to clean the data before building our SMT system. First, we converted all Arabizi to Arabic script; we then normalized it, trying to make it more like MSA. We also morphologically segmented long Arabic words using segmentation schemes that are standard for MSA but harder to do for dialectal Arabic. We used a statistical phrase-based MT system – Moses (Koehn et al., 2007). We experimented with different data processing schemes and different SMT system settings to achieve better translation quality. Here are the major settings: Egyptian Arabic segmentation (ATB, S2, D3), Egyptian Arabic to MSA conversion, sparse features, class-based models, neural network joint language model, hierarchically-interpolated language model, unsupervised transliteration mining, domain adaptation, and data selection for tuning.

It is worth mentioning that given the above-mentioned challenges, preprocessing by itself yielded the largest gains. The Egyptian Arabic segmentation gave us an improvement of up to 3 BLEU points. The hierarchically-interpolated language model added 1 extra BLEU point. The sparse features, class-based models, and neural network joint language models further improved translation quality by 0.66, 0.70 and 0.43 BLEU points absolute, respectively. In the next sections, we discuss in detail the different settings and decisions we made when preparing our submission.

The remainder of the paper is organized as follows: Section **??** explains the data preprocessing techniques and tools we tried, Section **??** explains in detail the non-standard features and components we tried in our SMT experiments, and Section **??** describes the actual system we submitted for the competition. Finally, Section **??** concludes with a summary of the main points.

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
No segmentation	21.02	21.64	20.27	22.34	20.60	23.36
D3	23.68	23.41	23.22	25.97	21.72	24.89
S2	23.62	23.66	22.82	25.41	21.61	24.67
ATB	23.57	23.50	22.82	26.01	21.68	24.83

Table 1: Comparing different Arabic word segmentation schemes.

2 Data Preprocessing

The NIST dataset contained text from three different genres: short text messages (SMS), chat (CHT), and transcribed conversational speech (CTS). We tackled each of the three genres separately, i.e., we built genre-specific systems. Furthermore, we split the provided training data for each genre into separate training and development sets by reserving approximately 3,000 sentences (the number of sentences is approximate as we did the splitting at the document level) from each set for development, and we used the rest for training.

For evaluation, the organizers provided two additional datasets: (*i*) an official devtest dataset (*Test*), and (*ii*) a small gold dataset (*TestG*), which is a subset of *Test*. The genres of the datasets were specified, and thus there was no need to train a system for automatic genre identification.

2.1 Modern Standard Arabic Preprocessing

For training purposes, some datasets for Modern Standard Arabic (MSA) were provided in addition to the Egyptian Arabic data (SMS, CHT, CTS) described above. These datasets consisted mostly of newswire text (i.e., a different genre), and thus we processed them using a standard MSA tool: MADAMIRA in MSA mode (Pasha et al., 2014). The MSA data was used to build a second phrase table to be then combined with some of the genre-specific phrase tables. We planned to match the same morphological tokenization schemes used for Egyptian Arabic, but at the end, we only used ATB segmentation for MSA. For more information on Arabic morphology challenges and tokenization schemes, see **?**).

2.2 Genre-Specific Preprocessing

The CTS data contained speech tags, markers for spelling corrections, etc., which we removed before training. The Arabic side of the SMS and CHT data had elongations and spelling mistakes. Therefore, we normalized the elongations and we standardized the spelling variations.

Parts of the SMS and CHT data contained romanized Arabic text (aka *Arabizi*), which is Arabic written using the Roman alphabet. In order to homogenize the input data, we converted the Arabizi into the standard Arabic script (utf8).

Given the conversational nature of SMS and chat, a further complication with converting romanized text into the Arabic script was that the text was usually affected by code-switching into English; therefore, it was important for us to identify the English words and not to try to convert them to Arabic. We solved these issues using the 3ARRIB tool (Al-Badrashiny et al., 2014).

On the English side of the data, there were multiple translation options, where typically the first option was the *intended meaning* and the second one the *literal meaning* of the Arabic text. We chose to use the *intended meaning* only.

Finally, in all processing, for both Arabic and English, we made sure the emoticons were not affected by the tokenization or the translation.

2.3 Egyptian Arabic Segmentation

A major issue when training an SMT system for the present edition of the task is the small size of the provided SMS, CHT and CTS datasets, which means that data sparseness is a severe problem. One common way to reduce it, at least on the Arabic side, where it is more severe, is to segment the Arabic words into multiple tokens, e.g., by separating the main word from the attached conjunctions, pronouns, articles, etc. Since these are separate words in English, such a segmentation not only reduces sparseness, but also yields improved word mapping to English, thus ultimately helping word alignments and translation model estimation for SMT. The value of Arabic tokenization for SMT, especially under low resource conditions, has been demonstrated by a number of researchers in the past, e.g., (Badr et al., 2008; **?**; Habash et al., 2013; Al-Mannai et al., 2014).

We experimented with common segmentation schemes such as D3, S2 and ATB (Badr et al.,

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Original Egyptian, no segmentation	21.02	21.64	20.27	22.34	20.60	23.36
Egyptian adapted to MSA, no segmentation	21.54	21.82	20.70	22.77	21.30	23.81
Egyptian adapted to MSA, then ATB-segmented	21.32	21.06	21.55	23.70	21.73	24.30

Table 2: Experiments in adapting Egyptian to look like MSA.

2008; **?**). For tokenization, we used MADAMIRA (Pasha et al., 2014), a fast and efficient implementation of MADA for MSA (Habash and Rambow, 2005; **?**), and MADA-ARZ, a version of MADA for Egyptian Arabic (Habash et al., 2013). Table 1 compares the results of using different segmentation schemes including no segmentation. We see gains of up to 3 BLEU points absolute on *Test* when using segmentation compared to no segmentation. However, the differences between the various schemes (ATB, S2 and D3) are small.

2.4 Egyptian to MSA Conversion

Another way to reduce data sparseness is by using additional out-of-domain data, e.g., newswire; this means a double domain shift: (*i*) from an informal text genre to newswire, and also (*ii*) from dialectal Arabic to MSA. While it is hard to do anything about the domain shift, the dialectal shift is somewhat easier to address. Changes between dialects are often systematic and many of the differences are at the level of individual words.

Previous work has shown that converting Egyptian to MSA makes it easier to use MSA resources for translating dialectal Arabic (**?**; Salloum and Habash, 2011; Zbib et al., 2012; Salloum and Habash, 2013; Sajjad et al., 2013a; Durrani et al., 2014a). So, we experimented with converting Egyptian to MSA using an in-house tool (Sajjad et al., 2013a), which performs character-level transformations for each Egyptian word in isolation to generate an MSA version thereof. We then trained an SMT system on this converted data.

The results are shown in Table 2. We can see that converting Egyptian to MSA yields improvements that are systematic across the three genres and also across the two test datasets. However, this improvement is not very large and ranges from 0.18 to 0.70 BLEU points absolute.

Further segmenting the MSA-like Egyptian using MADA ATB yielded very mixed results: in some cases, it added 0.93 BLEU points absolute, but in other there was a drop of 0.76. This could be because of the highly dialectal nature of the NIST data, which differs from MSA in lexical choice, which our tool cannot handle. Our Egyptian to MSA tool works at the character level and converts to MSA only those Egyptian words that are different at the character level.

In general, full conversion of dialectal Arabic to MSA would require not only word-level transformations but also phrase-level ones (**?**), while taking context into account (**?**), and also modeling morphological phenomena (**?**). There are also potential gains from a smarter character alignment models (**?**), or even from using a specialized decoder (**?**). Ultimately, the real benefit is when combining the adapted version of the smaller dialect with a large dataset in the bigger dialect (**?**; **?**), which we will do below.

3 Translation System Characteristics

We started our experiments from a strong baseline system, which was originally designed for MSA to English translation (Sajjad et al., 2013b). We then extended it with some additional models and features (Durrani et al., 2014b). Most notably, we used minimum Bayes risk decoding (MBR) (Kumar and Byrne, 2004), monotone-at-punctuation reordering, dropping of out-of-vocabulary words, operation sequence model for reordering (OSM) (Durrani et al., 2011; Durrani et al., 2013), a smoothed BLEU+1 version of PRO for parameter tuning (Nakov et al., 2012), etc.

Given this baseline system, we experimented with several further extensions, which we will describe below. Some of them were eventually included in our final submission.

3.1 Genre-Based Hierarchically-Interpolated Language Model

First, we experimented with building a hierarchically-interpolated language model for each genre (i.e., CTS, SMS, and CHT). We tuned each model to minimize the perplexity on a held-out set for that target genre.

We examined the text resources that were available for training an English language model, and

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Baseline using concatenated language model	24.19	24.00	23.34	25.89	22.75	25.09
System using interpolated language model	**25.20**	**25.04**	**23.48**	**26.16**	**23.01**	**25.67**

Table 3: Experiments with interpolated language models of each genre in comparison to the baseline language model built on the concatenation of the English side of SMS, CHT and CTS.

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Baseline	24.58	24.82	23.36	26.11	22.64	24.95
+ sparse features	24.54	25.36	**24.02**	**27.11**	21.61	24.08

Table 4: Experiments with sparse features.

we split them into six groups: (1) Egyptian-source (the target sides of the CTS, CHT and CTS training bi-texts), (2) MSA GALE News (GALE P3 {R1,R2},P4{R1,2,3}), (3) Chinese GALE (GALE P2 {BC,BC,BL,NG}), (4) MSA NEWS (newsetirr, news-par, news-trans, ISI), (5) MSA GALE non-news (GALE P1 {BLOG} P2 {BC1, BC2, WEB}), and (6) Gigaword v5, split into four subgroups by year (Guzmán et al., 2012) (1994-1997, 1998-2001, 2002-2005, 2006-2010).

For each such group, (*i*) we built an individual 5-gram language model with Kneser-Ney smoothing for each member of the group, and (*ii*) we interpolated these language models into a single language model, minimizing the perplexity for the target genre. Then, (*iii*) we performed a second (hierarchical) interpolation, this time combining the resulting six group language models, again minimizing the perplexity for the target genre. We used the SRILM toolkit (Stolcke and others, 2002) to build the language models.

Table **??** shows the results of using interpolated language model for each domain in comparison with using a language model built on the concatenation of the English side of the SMS, CHT and CTS corpora. The interpolated language model consistently improved all genres with a maximum improvement of up to 1 BLEU points on *Test*.

3.2 Translation Model with Sparse Features

The next thing we experimented with were sparse features (Chiang et al., 2009), which are a recent addition to the Moses SMT toolkit. In particular, we used target and source word insertion features: (*i*) top 50, and (*ii*) all. The latter worked better, and thus we only show results for it.

The results for *all* are shown in Table 3, where we can see that sparse features only helped for CHT, while they were harmful for CTS, and yielded mixed results for SMS. Thus, in our final system, we only used them for CHT.

3.3 Class-Based Language Models

Next, we experimented with using automatic word clusters, which we computed on the source and on the target sides of the training bi-text using *mkcls*. We also experimented with OSM models (**?**) over cluster IDs (Durrani et al., 2014c; Bisazza and Monz, 2014). Normally, the lexically-driven OSM model falls back to context sizes of 2-3 operations due to data sparseness, but learning operation sequences over cluster IDs enabled us to learn richer translation and reordering patterns that can generalize better in sparse conditions.

Table 4 shows the experimental results when adding a target language model and an OSM model over cluster IDs. We can see that these class-based models yielded consistent improvements in all cases. We also tried using *word2vec* (Mikolov et al., 2013) for clustering, but the results did not improve any further and they were occasionally worse than those with *mkcls*. We tried both with 50 and 500 classes, but using additional classes did not help.

3.4 Unsupervised Transliteration Models

A consequence of data sparseness is that at test time, the SMT system would see many unknown or out-of-vocabulary (OOV) words. One way to cope with them is to just pass them through untranslated. This works somewhat fine with newswire text and for languages with (roughly) the same alphabet, e.g., English and Spanish, as many OOVs are likely to be named entities (persons, locations, organizations), and are thus likely to be

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Baseline	24.22	24.33	23.02	25.60	21.93	24.88
+ class-based models	**24.63**	**25.16**	23.18	**26.30**	22.20	25.04

Table 5: Experiments with class-based language models.

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Baseline	24.58	24.33	24.02	27.11	22.64	24.95
+ NNJM Model	**25.01**	25.72	24.24	27.41	22.68	25.21

Table 6: Experiments with a neural network joint language model.

preserved in translation.

However, for languages with different scripts, such as Arabic and English, passing through is not a good idea, especially when translating into English as words in Arabic script do not naturally appear in English. In that case, it is much safer just to drop the OOVs, which is best done at decoding time; this was indeed our baseline strategy.

A better way is to transliterate OOV words either during decoding or in a post-processing step (**?**). We also experimented with this approach. For this purpose, we built an unsupervised transliteration model (Durrani et al., 2014d) based on EM as proposed in (Sajjad et al., 2011; Sajjad et al., 2012). Unfortunately, it did not help much, probably because in these informal genres the OOVs are rarely named entities; they are real words, which need actual translation, not transliteration.

3.5 Neural Network Joint Language Model

Recently, neural networks have come back from oblivion with the promise to revolutionize NLP. Major performance gains have already been demonstrated for speech recognition, and there have been successful applications to semantics. Most importantly for us, last year, very sizable performance gains were also reported for SMT using a neural joint language model or NNJM (Devlin et al., 2014).

We tried the Moses implementation of JNLM using the settings described in (Birch et al., 2014). While we managed to achieve consistent improvements for all three genres and for both test sets, as Table 5 shows, these gains are modest, 0.04–0.57 BLEU points absolute, which is far from what was reported in (Devlin et al., 2014). It is unclear what the reasons are, but it could have to do with the small size of our training bitexts and the informal genres we are dealing with.

3.6 Domain Adaptation

We experimented with different techniques for domain adaptation, trying to combine bi-texts from different genres, e.g., our Egyptian SMS, CHT, and CTS, but also MSA newswire.

First, we experimented with concatenating our SMS, CHT and CTS bitexts for training, but then using genre-specific tuning sets; this did not work as well as some other alternatives. Next, we experimented with building separate phrase tables, one in-domain and one out-of-domain, and then (a) using phrase table backoff, or (b) merging phrase tables and reordering tables as in (**?**; **?**; Sajjad et al., 2013b).

The results of our domain adaptation experiments when testing on SMS as the target genre are shown in Table 6. Note that the results on *Test* and on *TestG* differ a lot, and thus we focus on *Test* as it is much larger. We can see that the best way to combine SMS, CHT and CTS is simply to concatenate them, which yields +2.5 BLEU points of improvement absolute over training on SMS data only. Further small gains can be achieved by merging the resulting SMS+CHT+CTS phrase table with a phrase table trained on MSA, where the two tables are merged using extra indicator features as described in (**?**).

3.7 Tuning

Our phrase-based SMT system combines different features in a log-linear model. We tune the weights for the individual features of that model by optimizing BLEU (**?**) on a tuning dataset from the same genre as that in the test. We use PRO (**?**), but with smoothed BLEU+1 as proposed in (Nakov et al., 2012). We allowed the optimizer to run for up

Training data	Test	TestG
SMS	21.30	21.99
CAT(SMS, CHT, CTS)	**23.78**	23.20
SMS, Backoff(CHT,CTS)	22.55	23.00
CAT(SMS, CHT), Backoff(CTS)	22.54	23.20
MergePT(CAT(SMS, CHT), CTS)	23.69	**24.40**
CAT(SMS, CHT, CTS), Backoff(MSA)	23.70	**23.64**
MergePT(CAT(SMS, CHT, CTS), MSA)	**23.83**	23.60

Table 7: Experiments with different training data combinations (bitext concatenations, phrase table merging, and backoff), testing on SMS as the target genre.

Genre	Test	TestG
SMS - unfiltered tune	23.57	23.56
SMS - filtered tune	23.35	24.36
CTS - unfiltered tune	22.07	25.10
CTS - filtered tune	22.70	25.22

Table 8: Results using a length-filtered vs. an unfiltered dataset for tuning.

to 25 iterations, and to extract 1000-best lists on each iteration.

The choice of tuning set has been shown to have a huge impact on the quality of the learned parameters (Nakov et al., 2013a). In particular, PRO is very sensitive to length, which can result in pathological translations in some circumstances (Nakov et al., 2013b).

Given that there were no official development sets for this year, we synthesized datasets that are specific for CTS and SMS, based on sentence-length as a selection criterion.[1] This is crucial to remove potentially noisy data. It can also help to speed up the tuning process.

In order to achieve this, we filtered all sentence pairs for which either the source or the target sentences were shorter than 4 words or longer than 25 words. The cut-offs were determined empirically by analyzing the kernel density estimations (KDE $\alpha = 0.3$).

Table **??** compares tuning on an unfiltered and on a filtered tuning set. For SMS, using a filtered tuning set yields mixed results: we see a drop in BLEU on *Test* and a gain on *TestG*. For CTS, filtering helped both on *Test* and on *TextG*, with a gain on the former of +0.63.

4 Final Submission and Output Combination

We recombined hypotheses produced (a) by our best individual systems and (b) by other systems that are both relatively strong and can contribute diversity, e.g., using a different word segmentation scheme. For this purpose, we used the Multi-Engine MT system, or MEMT, (Heafield et al., 2009), which has been proven effective in such a setup and has contributed to achieving state-of-the-art results in a related competition in the past (Sajjad et al., 2013b).

The results are shown in Table **??**. We can see that using output combination yields notable improvement for SMS and CHT. However, for CTS, BLEU dropped by 0.45 points on *Test*. Thus, we submitted as a primary system the output combination for SMS and CHT, but our best individual system for CTS (which uses D3 segmentation).

5 Conclusion

We presented the Egyptian Arabic-to-English SMT system of the QCN team for the NIST OpenMT'2015 evaluation campaign. The system was ranked second in the competition on all three genres: SMS, chat, and speech.

Given the informal dialectal nature of these genres, we benefited from careful pre-processing, cleaning, and normalization, which yielded an improvement of up to 3 BLEU points over a strong baseline.

We further added a number of extra advanced features, which yielded 2.5 more BLEU points of absolute improvement on top of that due to pre-processing. In particular, sparse features contributed 0.7 BLEU points for CHT, class-based models added 0.7 and 0.6 BLEU points for CHT and SMS, respectively, and NNJM yielded gains of up to 0.4 BLEU points absolute.

References

[Al-Badrashiny et al.2014] 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.

[Al-Mannai et al.2014] Kamla Al-Mannai, Hassan Sajjad, Alaa Khader, Fahad Al Obaidli, Preslav Nakov,

[1] The CHT sentences were generally of reasonable length, and thus we did not apply filtering for them.

	SMS		CHT		CTS	
	Test	TestG	Test	TestG	Test	TestG
Best system with D3 segmentation	25.28	26.05	23.87	27.07	**23.34**	**26.05**
Best system with S2 segmentation	24.93	25.61	24.09	27.01	22.11	24.50
Best system with ATB segmentation	25.13	25.80	24.24	27.41	22.83	25.56
System with ATB segmentation + MSA phrase table as a backoff	25.20	25.04	23.48	26.16	23.01	25.67
Output combination	**26.13**	**26.79**	**24.86**	**27.95**	22.89	25.88

Table 9: Results for system recombination.

and Stephan Vogel. 2014. Unsupervised word segmentation improves dialectal Arabic to English machine translation. In *Proceedings of Workshop on Arabic Natural Language Processing*, Doha, Qatar, October.

[Badr et al.2008] Ibrahim Badr, Rabih Zbib, and James R. Glass. 2008. Segmentation for English-to-Arabic statistical machine translation. In *Proceedings of the Association for Computational Linguistics (ACL'08)*, Columbus, OH, USA.

[Birch et al.2014] Alexandra Birch, Matthias Huck, Nadir Durrani, Nikolay Bogoychev, and Philipp Koehn. 2014. Edinburgh slt and mt system description for the IWSLT 2014 evaluation. In *Proceedings of the 11th International Workshop on Spoken Language Translation*, pages 49–48, Lake Tahoe, US, December.

[Bisazza and Monz2014] Arianna Bisazza and Christof Monz. 2014. Class-Based Language Modeling for Translating into Morphologically Rich Languages. In *Proceedings of the 25th Annual Conference on Computational Linguistics (COLING)*, Dublin, Ireland, August.

[Chiang et al.2009] David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

[Devlin et al.2014] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, June.

[Durrani et al.2011] Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A Joint Sequence Translation Model with Integrated Reordering. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, Portland, OR, USA.

[Durrani et al.2013] Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov models over minimal translation units help phrase-based SMT? In *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria.

[Durrani et al.2014a] Nadir Durrani, Yaser Al-Onaizan, and Abraham Ittycheriah. 2014a. Improving Egyptian-to-English SMT by mapping Egyptian into MSA. In *Computational Linguistics and Intelligent Text Processing*, pages 271–282, Khatmandu, Nepal. Springer Berlin Heidelberg.

[Durrani et al.2014b] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014b. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, MD, USA, June.

[Durrani et al.2014c] Nadir Durrani, Philipp Koehn, Helmut Schmid, and Alexander Fraser. 2014c. Investigating the Usefulness of Generalized Word Representations in SMT. In *Proceedings of the 25th Annual Conference on Computational Linguistics (COLING)*, Dublin, Ireland, August.

[Durrani et al.2014d] Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014d. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2014)*, Gothenburg, Sweden, April. Association for Computational Linguistics.

[Eskander et al.2014] Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. 2014. Foreign words and the automatic processing of arabic social media text written in roman script. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 1–12, Doha, Qatar, October. Association for Computational Linguistics.

[Guzmán et al.2012] Francisco Guzmán, Preslav Nakov, Ahmed Thabet, and Stephan Vogel. 2012. QCRI at WMT12: Experiments in spanish-english and german-english machine translation of news text. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, pages 298–303, Montréal, Canada, June. Association for Computational Linguistics.

[Habash and Rambow2005] 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*, ACL '05, pages 573–580.

[Habash and Sadat2006] Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'06)*, New York, NY, USA.

[Habash et al.2013] Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of the 2013 Conference of NAACL-HLT*, Atlanta, GA.

[Heafield et al.2009] Kenneth Heafield, Greg Hanneman, and Alon Lavie. 2009. Machine translation system combination with flexible word ordering. In *Proceedings of the Workshop on Statistical Machine Translation (WMT'09)*, Athens, Greece.

[Hopkins and May2011] Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11.

[Koehn et al.2007] 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 Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.

[Kumar and Byrne2004] Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*, Boston, MA, USA.

[Mikolov et al.2013] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

[Nakov and Ng2009] Preslav Nakov and Hwee Tou Ng. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*, Singapore.

[Nakov et al.2012] Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *Proceedings of the International Conference on Computational Linguistics (COLING'12)*, Mumbai, India.

[Nakov et al.2013a] Preslav Nakov, Fahad Al Obaidli, Francisco Guzmán, and Stephan Vogel. 2013a. Parameter optimization for statistical machine translation: It pays to learn from hard examples. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP'13)*, September.

[Nakov et al.2013b] Preslav Nakov, Francisco Guzman, and Stephan Vogel. 2013b. A tale about PRO and monsters. In *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria.

[Pasha et al.2014] Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland.*

[Sajjad et al.2011] Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2011. An algorithm for unsupervised transliteration mining with an application to word alignment. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, Portland, OR, USA.

[Sajjad et al.2012] Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the Association for Computational Linguistics (ACL'12)*, Jeju, Korea.

[Sajjad et al.2013a] Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013a. Translating Dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, August.

[Sajjad et al.2013b] Hassan Sajjad, Francisco Guzmn, Preslav Nakov, Ahmed Abdelali, Kenton Murray, Fahad Al Obaidli, and Stephan Vogel. 2013b. QCRI at IWSLT 2013: Experiments in Arabic-English and English-Arabic spoken language translation. In *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, December.

[Salloum and Habash2011] Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21. Association for Computational Linguistics.

[Salloum and Habash2013] Wael Salloum and Nizar Habash. 2013. Dialectal arabic to english machine translation: Pivoting through modern standard arabic. In *HLT-NAACL*, pages 348–358.

[Stolcke and others2002] Andreas Stolcke et al. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Speech Communication Association (INTERSPEECH'02)*, Denver, CO, USA.

[Zbib et al.2012] Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of arabic dialects. In *The 2012 Conference of the North American Chapter of the Association for Computational Linguistics*, Montreal, June. Association for Computational Linguistics.