QCRI at IWSLT 2013:
Experiments in Arabic-English and English-Arabic Spoken Language Translation

Hassan Sajjad, Francisco Guzmán, Preslav Nakov, Ahmed Abdelali,
Kenton Murray, Fahad Al Obaidli, Stephan Vogel

Qatar Computing Research Institute
Qatar Foundation
{hsajjad,fguzman,pnakov,aabdelali,kwmurray,faalobaidly,svogel}@qf.org.qa

Abstract

We describe the Arabic-English and English-Arabic statistical machine translation systems developed by the Qatar Computing Research Institute for the IWSLT'2013 evaluation campaign on spoken language translation. We used one phrase-based and two hierarchical decoders, exploring various settings thereof. We further experimented with three domain adaptation methods, and with various Arabic word segmentation schemes. Combining the output of several systems yielded a gain of up to 3.4 BLEU points over the baseline. Here we also describe a specialized normalization scheme for evaluating Arabic output, which was adopted for the IWSLT'2013 evaluation campaign.

1. Introduction

We describe the Arabic-English and English-Arabic statistical machine translation (SMT) systems developed by the Qatar Computing Research Institute (QCRI) for the 2013 open evaluation campaign on spoken language translation organized in conjunction with the International Workshop on Spoken Language Translation (IWSLT). Below we give an overview of the settings we experimented with:

- **Decoders:** We used a phrase-based SMT (PBSMT), as implemented in Moses [1], and two hierarchical decoders: Jane [2] and cdec [3]. See Section 6 for details.
- **Decoder settings:** There are a variety of settings available for the above decoders. We explored a number of them, most notably, operation sequence model, minimum Bayes risk decoding, monotone-at-punctuation, dropping out-of-vocabulary words, etc. We selected to retain those settings that improved the overall translation quality as measured on the dev-test set. See Section 4 for further details.
- **Arabic segmentation:** To reduce data sparseness, Arabic words are typically segmented into multiple tokens, e.g., by segmenting out conjunctions, pronouns, articles, etc. We experimented with standard segmentation schemes such as D0, D1, D2, D3, S2 and ATB, as defined in MADA [4, 5]. See Section 5 for details.
- **Domain adaptation:** We experimented with three domain adaptation methods to make better use of the huge UN data, which is out-of-domain: (i) Modified Moore-Lewis filtering, (ii) phrase table merging, and (iii) phrase table backoff. See Section 7 for details.

For our final submission, we synthesized a translation by combining the output of our best individual system with the output of other systems that are both relatively strong and can contribute to having more diversity, e.g., using a different decoder or a different segmentation scheme.

We achieved the most notable improvements in terms of BLEU when translating from Arabic-to-English using an operation sequence model (+0.6 BLEU on tst2010), phrase table merging and phrase table backoff (+0.6 BLEU), interpolated language model (+1.5 BLEU), and system combination using different decoders and different segmentation schemes (+0.6 BLEU). For the English-to-Arabic direction, we observed smaller improvements compared to the reverse direction, but there the absolute baseline was also much lower.

Finally, we proposed normalization for Arabic output evaluation, which was adopted as official for IWSLT'2013.[1]

2. Data

For the Arabic-English language pair, the IWSLT'2013 training data consisted of a small in-domain bitext, i.e. the TED talks[2] (IWSLT), and a large out-of-domain bitext, i.e. the multiUN corpus (UN). There were also tuning and development bitexts: dev2010 and tst2010. Conversely, for language modeling, a larger number of monolingual corpora were permissible. They are all listed in Table 1, together with their corresponding word count statistics.

3. Baseline

Data. We built a baseline system using the Moses toolkit and the IWSLT training data only, i.e., the TED talks. At development time, we tuned and tested on the provided dev2010 and tst2010 datasets.

[1] The normalizer is freely available at http://alt.qcri.org/tools/.

[2] https://wit3.fbk.eu/mt.php?release=2013-01

Monolingual corpora	**# Words**
English	
IWSLT mono	2.7M
10^9 English-French	575M
SETimes	4.2M
UN (Es-En + En-Fr)	597M
UN (Ar-En)	115M
News Crawl 2007-2009	643M
News Crawl 2009-2012	745M
Common Crawl	185M
Wiki Headlines	1.1M
Europarl v.7	54M
News Commentary v.8	5.3M
Gigaword v.5	4,032M
Arabic	
IWSLT mono	2.7M
UN	134M
News Commentary Arabic v.8	4.8M
Gigaword Arabic v.5	1,373M

Table 1: Admissible training data for language modeling. Here English is tokenized, and Arabic is ATB-segmented.

Preprocessing. We segmented the Arabic side of the bitext following the ATB scheme and using the Stanford word segmenter [6]. For the English side, we used the standard tokenizer of Moses, and we further applied truecasing/lowercasing when English was the target/source language.

Training. We built separate directed word alignments for English-to-Arabic and for Arabic-to-English using IBM model 4 [7], and we symmetrized them using the *grow-diag-final-and* heuristics [8]. We then extracted phrase pairs with a maximum length of seven, and we scored them using maximum likelihood estimation with Kneser-Ney smoothing, thus obtaining a phrase table where each phrase pair has the standard five translation model features. We also built a lexicalized reordering model [9]: *msd-bidirectional-fe*. For language modeling, we used KenLM [10] to build a 5-gram Kneser-Ney smoothed model, trained on the target side of the training bi-text. Finally, we built a large joint log-linear model, which used standard PBSMT feature functions: language model probability, word penalty, the parameters from the phrase table, and those from the reordering model.

Tuning. We tuned the weights in the log-linear model by optimizing BLEU [11] on the tuning dataset, using PRO [12]. We allowed the optimizer to run for up to 10 iterations, and to extract 1000-best lists on each iteration.

Decoding. On tuning and testing, we used monotone-at-punctuation. On testing, we further used cube pruning.

Table 2 shows the results[3] for the baseline English-to-Arabic and Arabic-to-English SMT systems, compared to the baseline results reported on the WIT[3] webpage.

	Arabic-English		**English-Arabic**	
System	BLEU	1-TER	BLEU	1-TER
IWSLT baseline	23.6	43.0	11.9	28.6
Our baseline	24.7	45.6	12.6	29.1

Table 2: Our vs. IWSLT baseline results for English-to-Arabic and Arabic-to-English SMT, evaluated on tst2010.

4. System Settings

Below we discuss the decoder settings and extensions we experimented with, focusing on Arabic-to-English. Table 3 shows the impact of each feature when added to the baseline.

Tuning. [13] have shown that PRO tends to generate too short translations.[4] They have suggested that the root of the problem was that PRO optimizes sentence-level BLEU+1, which smooths the precision component of BLEU, but leaves the brevity penalty intact, which destroys the balance between them. They have proposed a number of fixes, the simplest and most efficient among them being to smooth the brevity penalty as well.[5] In our experiments, this yielded +0.2 BLEU for Arabic-to-English on tst2010.

Operation sequence model. The operation sequence model (OSM) is an n-gram-based model, which represents the aligned bitext into a sequence of operations, e.g., generate a sequence of source and target words or perform reordering. The model memorizes Markov chains over such sequences, thus fusing lexical generation and reordering into a single generative model. OSM offers two advantages. First, it considers bilingual contextual information that goes beyond phrase boundaries. Second, it provides a better reordering mechanism that has richer conditioning than a lexicalized reordering model: the probability of an operation is conditioned on the n previous translation and reordering decisions. We used the Moses implementation of OSM [15], which has yielded improvements at WMT'13 [16]. In our experiments, it yielded +0.6 BLEU for Arabic-to-English on tst2010.

Minimum Bayes risk decoding. We also experimented with minimum Bayes risk decoding (MBR)[17], which, instead of outputting the translation with the highest probability, prefers the one that is most similar to best n translations. In our case, using MBR did not improve over the baseline.

Translation options per input phrase. By default, Moses uses up to 20 translation options per input phrase, but [16] have shown better results with 100. In our experiments, this yielded +0.1 BLEU for Arabic-to-English on tst2010.

Transliterating OOVs. Out-of-vocabulary (OOV) words are problematic for languages with different scripts. Thus, we tried transliteration as post-processing: we extracted 1-1 word alignments from a subset of the UN bitext, and we used them to train a character-level transliteration system [18, 19] using Moses. As Table 3 shows this did not help, probably due to the small number of OOVs in tst2010.

[3] For tst2010, we report MultEval BLEU and TER0.8: on tokenized and recased output for English, and on QCRI-normalized output for Arabic. For tst2011, tst2012, and tst2013, the organizers used slightly different scorers.

[4] See [14] for a discussion about more potential issues with PRO.

[5] Available in Moses: `--proargs='--smooth-brevity-penalty'`

	Arabic-English (tst2010)	
System	BLEU	1-TER
Baseline (B)	24.7	45.6
OSM	25.3	46.1
MBR	24.7	45.7
Ttable 100	24.8	45.6
PRO-fix [13]	24.9	44.7
TRANSLIT	24.7	45.6
Drop UNK	24.8	45.7

Table 3: Impact of each feature when added to the baseline.

Dropping OOVs. An alternative to transliteration is to just drop all OOV words as part of the decoding process. We did this on both tuning and testing, and it yielded +0.2 BLEU for Arabic-to-English on tst2010.

Language model. For language modeling (LM), we used most of the available data shown in Table 1, processed with the Moses tokenizer for English, and with the Stanford ATB segmenter for Arabic. For each data source, we trained a separate 5-gram LM with Kneser-Ney smoothing. We then interpolated these models, minimizing the perplexity on the target side of dev2010.[6] Finally, we binarized them using KenLM [10] with probing and no quantization. Table 4 shows that using these LMs yields +1.5 BLEU for English, but only +0.6 for Arabic; this is probably due to less data being available for Arabic LM training.

	BLEU tst2010	
System	Arabic-English	English-Arabic
Baseline (TED LM)	24.7	10.6
Large LM	26.2	11.2

Table 4: The impact of using a large LM on tst2010.

5. Arabic Segmentation

In Arabic, various clitics such as pronouns, conjunctions and articles appear concatenated to content words such as nouns and verbs. This can cause data sparseness issues, and thus clitics are typically segmented in a preprocessing step. There are various standard segmentation schemes defined in MADA [4, 5] such as D0, D1, D2, D3 and S2, for which we used the MADA+TOKAN toolkit [20], as well as ATB, which we performed using the Stanford segmenter [6]. Table 5 shows the results when training on the TED bitext only. We can see that ATB performed the best overall with a BLEU score of 24.7, followed by S2 with a score of 24.5.

	Arabic-English (tst2010)	
System	BLEU	1-TER
SEG-D0	22.4	43.0
SEG-D1	23.6	44.2
SEG-D2	24.1	45.2
SEG-D3	24.4	45.5
SEG-S2	24.5	45.7
SEG-ATB	24.7	45.6

Table 5: Using different Arabic segmentation schemes.

	Arabic-English (tst2010)	
System	BLEU	1-TER
Moses PBSMT	24.7	45.6
cdec	24.3	44.6
Jane	24.1	43.6

Table 6: Baseline results with different decoders.

6. Decoders

In our experiments, we used several decoders. Table 6 shows the baseline results for each of them.

Moses PBSMT. We used the phrase-based model as implemented in Moses [1]. It is described in our baseline above.

cdec. We further experimented with the hierarchical cdec decoder [3]. We used its default features: forward and backward translation features, singleton features, a glue-rule probability, and a pass-through feature (to handle OOVs). We tuned the parameters using MIRA with IBM BLEU as the objective function and a k-best forest size of 250.

Jane. We also used another hierarchical phrase-based decoder: Jane 2.2 [2]. We used the standard features: phrase translation probabilities and lexical smoothing in both directions, word and phrase penalties, a distance-based distortion model, and a 5-gram LM. We optimized the weights using MERT [21] on 100-best candidates with BLEU as objective.

7. Adaptation

The IWSLT dataset contains a small in-domain corpus (TED talks) and a large out-of-domain corpus (UN). In this section, we explore various ways to make best use of the out-of-domain data to improve the baseline system.

7.1. Modified Moore-Lewis Filtering (MML)

Moore and Lewis [22] presented a method for selecting relevant sentences from out-of-domain data for language modeling. Axelrod et al. [23] further extended it to parallel corpora, considering both the source and the target side of the bi-text, as well as in-domain and out-of-domain data, when scoring each sentence pair; their method is known as modified Moore and Lewis, or MML. They have shown that MML can yield improvements in SMT quality when selecting as little as just 1% of the out-of-domain training bi-text.

[6] For Gigaword, a preliminary interpolation between models computed over two-year partitions of the corpus (e.g., 2005 and 2006) was necessary because of memory limitations of the machines we used to train the LMs.

System	Training	BLEU	1-TER
baseline	IWSLT	24.7	45.6
MML1	IWSLT+2%UN	24.4	45.6
MML2	IWSLT+3%UN	24.4	45.6
MML3	IWSLT+4%UN	24.3	45.1
MML4	IWSLT+5%UN	24.2	45.6
MML5	IWSLT+100%UN	21.9	42.8

Table 7: Arabic-to-English: training on the IWSLT bi-text plus various MML-filtered UN bi-texts.

We experimented with MML, selecting varying percentages of out-of-domain UN data. Note that this additional data impacts all models: the translation model, the reodering model, and the language model. However, in order to allow for more fair head-to-head comparison, in Table 7 we show experimental results where we limit the LM training data to IWSLT only. We can see that each MML-adapted system suffers a drop in performance compared to the baseline system, which can be attributed to the differences between the in-domain and the out-of-domain data in terms of sentence structure, vocabulary, and style. Note that using just 2% and 3% of UN data works best, but this is still worse than not using UN data at all.

7.2. Merging Translation and Reordering Models

Given the negative results with MML, we also tried an alternative way to make use of the out-of-domain UN data, namely phrase table merging as described in [24, 25]. In the merged phrase table, we kept either (a) both phrases, or (b) the one coming from the in-domain data only. In either case, we added three additional binary features for each phrase pair indicating whether it came from (i) the in-domain data, (ii) the out-of-domain data, and (iii) both. Similarly, we merged reordering models, where we preferred the scores from the in-domain model. We further experimented with merging a phrase table for IWSLT with one for 3% of UN.

The results are shown in Table 8; note that this time we use the large interpolated language model presented in Table 4.. We show results for merging IWSLT with 3% of the UN data (MER1, MER2) as well as with the full UN (MER3, MER4), with duplicates kept (MER1, MER3) or removed (MER2, MER4). For comparison, we also show the baseline of using IWSLT only. We can see that using the full UN data works best, yielding +0.6 BLEU points of improvement.

7.3. Backoff Phrase Tables

The Moses toolkit allows for the use of a *backoff* phrase table in addition to a *main* phrase table. The phrases from the backoff phrase table are used when the translation of a phrase is unknown to the main phrase table. The backoff order determines the maximum phrase length for which this operation is allowed.

System	Training	BLEU	1-TER
baseline	IWSLT	26.2	46.6
MER1	IWSLT & 3%UN	26.2	46.4
MER2	IWSLT & 3%UN, no-dup	26.5	46.7
MER3	IWSLT & UN	26.6	47.0
MER4	IWSLT & UN, no-dup	26.8	47.1

Table 8: Arabic-to-English: phrase table merging.

In our experiments, we considered the phrase table built using the in-domain data as the main phrase table, and that built using the full UN data as the backoff phrase table. We tried n-grams of different orders for the backoff. Table 9 shows the results for backoff orders of 4, 5 and 6; again, we use the large interpolated language model presented in Table 4.. We can see that backoff orders of 4 and 5 performed best, achieving results that are very similar to what we obtained with phrase tables merging: comparing Table 9 to Table 8, we see the same BLEU score of 26.8, and a bit different 1-TER score. We believe that this indicates that the UN data is mostly useful for specific cases, e.g., to translate unknowns, but that it should not be blindly concatenated to the in-domain data because this hurts the performance.

System	Backoff order	BLEU	1-TER
baseline	0	26.2	46.6
BO1	4	26.8	47.2
BO2	5	26.8	47.2
BO3	6	26.7	47.2

Table 9: Arabic-to-English: phrase table backoff.

7.4. Best Adaptation

In the remainder of this paper, we will consider the MER4 system as our best adapted system. Note that when we also use OSM trained on the IWSLT bi-text, the BLEU score further improves by +0.6 points. Table 10 shows these results.

System	BLEU	1-TER
MER4	26.8	47.1
MER4+OSM$_{in}$	27.4	47.9

Table 10: Arabic-to-English: our best adapted system MER4 combined with OSM.

8. Arabic-to-English Machine Translation

We built several Arabic-to-English SMT systems based on the settings described in the previous sections; we further used system combination to produce our final translation. Below we give details about the individual systems.

System	Training	BLEU	1-TER
SEG-D1	IWSLT-3%UN	25.5	45.7
SEG-D2	IWSLT-3%UN	26.3	46.5
SEG-D3	IWSLT-3%UN	26.4	47.2
SEG-S2	IWSLT-3%UN	26.7	47.3
SEG-ATB	IWSLT-3%UN	27.0	47.4
cdec	IWSLT	25.4	45.4
cdec-UN	IWSLT-3%UN	25.3	45.6
Jane	IWSLT	24.7	42.5
FF	**IWSLT-100%UN**	**27.5**	**47.9**

Table 11: Arabic-to-English SMT systems (tst2010).

Segmentation. We built five phrase-based SMT systems, each using a different MADA segmentation scheme for the Arabic side: D1, D2, D3, S2 and ATB. We did not segment the complete UN data with each of these segmentation schemes due to time constraints. Instead, we used the 3% UN data filtered using MML to build a phrase table, which we then merged with the phrase table for IWSLT, preferring IWSLT phrase pairs in case of duplicates; this yielded systems coresponding to the MER2 line in Table 8. We further used OSM and MBR.

Decoder. We used three decoders: one phrase-based (Moses) and two hierarchical (cdec and Jane). Note that most of the settings described in Section 4 are applicable to the phrase-based decoder only. We trained cdec and Jane on the IWSLT data only, while still using the large interpolated LM. For cdec, we further built another system which was trained on a concatenation of the IWSLT data and the 3% UN data.

Full featured run. Finally, we further extended the MER4-OSM$_{in}$ system (see Table 10), which uses the complete UN data and the adapted OSM, with two additional settings: (i) MBR and (ii) ttable 100. This is our best indiviual run that does not use system combination, which we will call Full Featured (FF) below. We submitted it as our contrastive run to the competition.

Table 11 summarizes the results for all our Arabic-to-English SMT systems.

8.1. System Combination Results

We recombined hypotheses produced by various subsets of the systems in Table 11 using the Multi-Engine MT system (MEMT) [26]. The results are presented in Table 12. We can see that combining all segmentations yields +0.4 BLEU over our best individual system FF. Further adding cdec to the combination, yields another +0.2 BLEU; this was our primary system for Arabic-to-English.

8.2. Official Results

Table 13 shows the official results of our Arabic-to-English contrastive and primary systems. PRM is our primary system, a system combination of all systems in Table 11.

System	BLEU	1-TER
FF	27.5	47.9
FF, SEG-ALL	27.9	47.4
FF, cdec-UN	27.7	47.2
FF, cdec-UN, Jane	27.6	47.4
FF, SEG-ALL, cdec, cdec-UN	**28.1**	**47.6**

Table 12: Arabic-to-English syscomb (tst2010).

System	tst2011		tst2012		tst2013	
	BLEU	1-TER	BLEU	1-TER	BLEU	1-TER
FF	26.9	44.8	28.7	49.7	30.0	48.9
PRM	27.8	44.8	30.3	50.5	30.5	48.6

Table 13: Arabic-to-English: official scores (mteval-v13a).

9. English-to-Arabic Machine Translation

For English-to-Arabic translation, we experimented with different segmentation schemes: D0, D1, D2, D3, S2 (using MADA), and ATB (using the Stanford segmenter). Note that this is more complicated here than for Arabic-to-English because the segmentation is on the target side; thus, for English-to-Arabic SMT, there is need for (i) a separate LM for each segmentation, and (ii) desegmentation of the output.

A separate LM for each segmentation. Since the segmentation is on the target side, it applies to the language model as well. This means that if we wanted to experiment with different segmentations, we needed a separate language model for each of them, which is time- and resource-consuming. In practical terms, this prevented us from building strong language models for D0, D1, D2, D3 and S2, for which we used an LM trained on the Arabic side of the IWSLT bi-text only. It was for the ATB segmentation only that we could build a strong LM through interpolation, similarly to our Arabic-to-English LM, that also used the Gigaword Arabic, UN, and News Commentary data (see Table 1).

Desegmentation. Unlike the Arabic-to-English direction, where the segmentation was on the input side and thus the output was unaffected, here the segmentation had to be undone. For example, if we use an ATB-segmented target side, we end up with an ATB-segmented translation output, which we have to desegment in order to obtain proper Arabic. Desegmentation is not a trivial task since it involves some morphological adjustments, see [27] for a broader discussion. For desegmentation, we used the best approach described in [27]; in fact, we used their implementation.

Normalization. Translating into Arabic is tricky because the Arabic spelling is often inconsistent in terms of punctuation (using both Arabic UTF8 and English punctuation symbols), digits (appearing as both Arabic and Indian characters), diacritics (can be used or omitted, and can often be wrong), spelling (there are many errors in the spelling of some Arabic characters, esp. *Alef* and *Ta Marbuta*; also, *Waa* appears sometimes separated). These problems are especially frequent in informal texts such as TED talks. Thus, we normalized Arabic to make it more consistent. We first concatenated back the conjunction *Waa* when detached (it is almost never detached in proper Arabic). We then used MADA to normalize the following: (i) punctuation: converted Arabic UTF8 punctuation to English, (ii) digits: converted all Indian digits to the standard Arabic digits 0,1,. . .,9, (iii) diacritics: dropped them all, (iv) spelling: fixed potential errors in the different forms of *Alef*, *Alef Maqsura*, *Ta Marbuta*, etc. Finally, we converted all instances of "..", which are common in informal Arabic text, but are never used in English, to "...".

Tokenization and detokenization. We further had to perform tokenization and detokenization. Note that this is different from segmentation: segmentation is about splitting words into multiple words, while tokenization is mainly about separating punctuation from words. For tokenization, we used the Europarl tokenizer: note that it does not work on general Arabic text (e.g., because it cannot handle the UTF8 Arabic punctuation symbols), but it works just fine on our normalized Arabic. For detokenizing the final Arabic desegmented output, we used the Moses detokenizer; again, it only works because it sees normal English punctuation.

Scoring the Arabic SMT output. While the systems participating in IWSLT'2013 were supposed to output proper Arabic, directly scoring their output against the references with the NIST scoring tool v13a is problematic because of the above-described inconsistencies in Arabic, which also happen in the references for the tuning and the testing sets (in addition to training). Since these variations are quite random and depend on the style of the author of each piece of text, it does not make sense for a translation system to try to model them. Yet, they can affect evaluation scores a lot![7] Thus, we normalize both the SMT output and the reference with the QCRI normalizer: it applies the above-described normalization and also performs tokenization. Then, we calculate a BLEU and a TER score using MultEval, which does not perform internal tokenization (unlike the NIST scoring tool). This scoring procedure is official for the English-to-Arabic translation direction at IWSLT'2013.

9.1. Individual and Combined Systems

The results for the individual systems are shown in Table 14. We can see that ATB performs best, which is to be expected since it uses a much larger LM. However, adding the UN bitext in phrase table combination had a very minor impact on BLEU, only adding +0.2 points to FF.

Similarly to the Arabic-to-English system, we used MEMT to combine the outputs of several systems. The challenge was to make these outputs compatible: they were to be (1) desegmented, and (2) re-segmented using the ATB scheme. This allowed us to perform system combination using the large Arabic ATB language model.

System	Training	BLEU	1-TER
SEG-D0	IWSLT	12.3	30.2
SEG-D1	IWSLT	12.6	30.6
SEG-D2	IWSLT	12.5	30.7
SEG-D3	IWSLT	12.5	30.5
SEG-S2	IWSLT	12.5	30.2
SEG-ATB	IWSLT, big-LM	13.6	31.3
cdec	IWSLT	12.7	29.8
Jane	IWSLT	12.2	28.8
FF	**IWSLT+UN, big-LM**	**13.8**	**31.4**

Table 14: English-to-Arabic SMT systems (tst2010).

System	BLEU	1-TER
FF	13.8	31.4
FF, SEG-ALL, cdec	**13.7**	**30.2**

Table 15: English-to-Arabic syscomb (tst2010).

We tried many system combinations, but we were unable to improve over FF. Table 15 shows our best combination; even though it yielded -0.1 BLEU points on tst2010, we submitted it as primary, to be consistent with Arabic-to-English.

9.2. Official Results

Table 16 shows the official results of our English-to-Arabic contrastive and primary runs. We can see that the system combination performed slightly better, after all.

10. English-to-Arabic Spoken Translation

Translating the ASR output poses several additional challenges over translating properly transcribed text such as (1) finding sentence boundaries, (2) restoring case, and (3) restoring punctuation. Note that for this year's competition, speech segmentation was provided by the organizers, which solves (1). We further trained our English-to-Arabic SMT system on lowercase English input, thus eliminating the need for (2). Lastly, we addressed (3) by considering two levels of punctuation restoration. As a baseline, we just inserted a full stop at the end of each sentence. Next, we treated punctuation marks as hidden events occurring between words. Thus, the problem was reduced to finding the most likely tag sequence using an n-gram language model.

System	tst2011		tst2012		tst2013	
	BLEU	1-TER	BLEU	1-TER	BLEU	1-TER
FF	15.15	31.66	15.68	35.28	15.68	35.82
PRM	15.54	30.81	15.54	34.43	15.78	34.57

Table 16: English-to-Arabic: our official results (calculated using the QCRI normalizer, then MultEval).

[7] E.g., the score for the organizer's baseline system goes up from 9.61 (after tokenization with Europarl) to 11.89 when using the QCRI normalizer.

For this purpose, we used the *hidden-ngram* tool from the SRILM toolkit [28]. We trained the LM on the tokenized monolingual English portion of the IWSLT training data. The list of punctuation marks (tags) included the following: *comma (,), semi-colon (;), colon (:), quotation marks ("), question marks (?), period (.),* and *ellipsis (...)* .

For our contrastive SLT system, we reused the best English-to-Arabic system from the previous section (FF). Table 17 shows the results for different methods for punctuation restoration. Note that decoding with a simple full stop addition improved the score by about +1.3 BLEU points. Further restoring the rest of the punctuation marks yielded an additional improvement of +1.3 BLEU points. As a reference, we also include the *Oracle* input, i.e., the MT text input (with the same sentence segmentation as the ASR's 1-best).

System	**tst2010**	
	BLEU	1-TER
Raw 1-best input	6.2	21.1
+ full stop at the end	7.5	23.6
+ punctuation restoration	8.8	23.7
Text input (Oracle)	14.0	31.3

Table 17: English-to-Arabic SLT: punctuation restoration.

10.1. System Combination Results

Similarly to the English-to-Arabic text translation, we used MEMT to combine the output of several systems. The combined output yielded +0.1 BLEU points over the best system.

10.2. Official Results

Table 18 shows the official results for our English-to-Arabic SLT submissions: contrastive (FF single-best) and primary (PRM, system combination). The systems are the same as for English-to-Arabic text translation.

System	**tst2013**	
	BLEU	1-TER
FF	10.27	26.24
PRM	10.33	26.28

Table 18: English-to-Arabic SLT: our official results (calculated using the QCRI normalizer, then MultEval).

11. Conclusion

We have presented the Arabic-English and English-Arabic SMT systems developed by the Qatar Computing Research Institute for the IWSLT'2013 evaluation campaign on spoken language translation. We experimented with three decoders and various settings thereof, we tried different domain adaptation methods, and we performed system combination. For the Arabic side, we also used various segmentation schemes.

For domain adaptation, we achieved best results with the full UN data and phrase table merging. The SMT systems built using different MADA segmentation schemes for Arabic (the ATB segmentation was strongest) and using different decoders (Moses performed better than cdec and Jane.) added diversity and were useful for system combination.

For English-to-Arabic, we observed that the gains from the various decoding settings, domain adaptation and system combination were all lower compared to those for the Arabic-to-English system. We plan to investigate this in future work.

Finally, we proposed normalization for Arabic output evaluation, which was adopted as official for IWSLT'2013.

Acknowledgements. We would like to thank Nizar Habash for sharing the Arabic desegmentation code.

12. References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic, 2007.

[2] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *Proceedings of the Workshop on Statistical Machine Translation and Metrics MATR (WMT'10)*, Uppsala, Sweden, 2010.

[3] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models," in *Proceedings of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, 2010.

[4] N. Habash and F. Sadat, "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, 2006.

[5] I. Badr, R. Zbib, and J. R. Glass, "Segmentation for English-to-Arabic statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'08)*, Columbus, OH, USA, 2008.

[6] S. Green and J. DeNero, "A class-based agreement model for generating accurately inflected translations," in *Proceedings of the Association for Computational Linguistics (ACL'12)*, Jeju Island, Korea, 2012.

[7] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[8] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL'03)*, Edmonton, Canada, 2003.

[9] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'05)*, Pittsburgh, PA, USA, 2005.

[10] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'11)*, Edinburgh, UK, 2011.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA, USA, 2002.

[12] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Edinburgh, UK, 2011.

[13] P. Nakov, F. Guzmán, and S. Vogel, "Optimizing for sentence-level BLEU+1 yields short translations," in *Proceedings of the International Conference on Computational Linguistics (COLING'12)*, Mumbai, India, 2012.

[14] P. Nakov, F. Guzman, and S. Vogel, "A tale about PRO and monsters," in *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[15] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can Markov models over minimal translation units help phrase-based SMT?" in *Proceedings of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[16] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburgh's machine translation systems for European language pairs," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'13)*, Sofia, Bulgaria, 2013.

[17] S. Kumar and W. Byrne, "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, 2004.

[18] H. Sajjad, A. Fraser, and H. Schmid, "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, 2011.

[19] ——, "A statistical model for unsupervised and semi-supervised transliteration mining," in *Proceedings of the Association for Computational Linguistics (ACL'12)*, Jeju, Korea, 2012.

[20] O. Rambow, N. Habash, and R. Roth, "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2009.

[21] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'03)*, Sapporo, Japan, 2003.

[22] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, 2010.

[23] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, Edinburgh, UK, 2011.

[24] P. Nakov, "Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'08)*, Columbus, OH, USA, 2008.

[25] P. Nakov and H. T. Ng, "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, 2009.

[26] K. Heafield, G. Hanneman, and A. Lavie, "Machine translation system combination with flexible word ordering," in *Proceedings of the Workshop on Statistical Machine Translation (WMT'09)*, Athens, Greece, 2009.

[27] A. El Kholy and N. Habash, "Orthographic and morphological processing for English–Arabic statistical machine translation," *Machine Translation*, vol. 26, no. 1-2, 2012.

[28] A. Stolcke *et al.*, "SRILM – an extensible language modeling toolkit." in *Proceedings of the International Speech Communication Association (INTERSPEECH'02)*, Denver, CO, USA, 2002.