QCRI Machine Translation Systems for IWSLT 16

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Stephan Vogel

Qatar Computing Research Institute – HBKU

{ndurrani,faimaduddin,hsajjad,svogel}@qf.org.qa

Abstract

This paper describes QCRI's machine translation systems for the IWSLT 2016 evaluation campaign. We participated in the Arabic→English and English→Arabic tracks. We built both Phrase-based and Neural machine translation models, in an effort to probe whether the newly emerged NMT framework surpasses the traditional phrase-based systems in Arabic-English language pairs. We trained a very strong phrase-based system including, a big language model, the Operation Sequence Model, Neural Network Joint Model and Class-based models along with different domain adaptation techniques such as MML filtering, mixture modeling and using fine tuning over NNJM model. However, a Neural MT system, trained by stacking data from different genres through fine-tuning, and applying ensemble over 8 models, beat our very strong phrase-based system by a significant 2 BLEU points margin in Arabic→English direction. We did not obtain similar gains in the other direction but were still able to outperform the phrase-based system. We also applied system combination on phrase-based and NMT outputs.

1. Introduction

We describe QCRI's phrase-based and Neural MT systems. We participated in the Arabic-to-English and English-to-Arabic MT tracks. Our translation engines have been historically based on the phrase-based system trained using the Moses toolkit [1], but during the course of this evaluation, we made a transition towards the newly emerged Neural MT framework [2], using Nematus, a toolkit used by the top performing team [3], during the recent WMT campaign.

An interesting challenge associated with the IWSLT campaign is the problem of domain adaptation. The in-domain data based on TED talks is available in very little quantity compared to the out-domain UN corpus [4], which has been found to be harmful previously when simply concatenated to the training [5]. In this year's IWSLT, two additional data resources Opus subtitles [6] and the QED corpus [7] were introduced. The latter was also used as an official test-set. Therefore apart from exploring phrase-based versus Neural MT, we geared ourselves towards adapting our system towards TED and QED talks in this multi-domain scenario. With these goals in mind we re-explored both model weighting and data filtering techniques, in these new data settings. Below we itemize the most successful attributes of our phrase-based system:

- We applied MML-based data selection [8] to the UN and Open Sub-title data, with the goals of filtering out harmful data.
- We trained OSM models [9] on separate corpora, and interpolated them [10] by optimizing perplexity on the tuning-set. We also tried this on the OSM models trained on the word classes [11].
- We tried the fine-tuning method of training the NNJM model on the out-domain data and fine-tuning with the in-domain TED data [12].
- We trained big language models using all the English mono data available from the WMT campaign and giga word corpus for Arabic.

We trained our Neural MT system using the Nematus toolkit. We used Bidirectional RNN's for the encoder, 1024 LSTM units, and a word embedding size of 500. Below we itemize what worked when training the neural MT system:

- We trained our baseline model on all of the UN corpus, then continued training with the Open subtitles corpus, and finally fine-tuned with the in-domain data
- We fine-tuned all of our models without freezing any layers in the network, since we had sufficient amount of data to train on.
- We used dropout when fine-tuning with in-domain data, since it is relatively small compared to the UN and Open subtitle data.
- We trained our final models with an ensemble of the last eight models, where each model was fine-tuned with the in-domain data.

Finally we applied system combination over the outputs of best Neural MT and phrase-based systems using MEMT [13]. Our efforts were mainly focused towards the AR→EN TED task. In the end we just replicated our best system for the EN→AR direction and the QED task. For our best Neural MT system, we were unable to use an ensemble in the EN→AR direction, since we could not train several comparable models to combine.

	TED	QED	UN	OPUS
Stats	240K	153K	18.5M	40M

Table 1: Number of Sentences in Parallel Data

Segmentation	ted-11	ted-12	ted-13	ted-14	Avg
MADA	27.5	30.6	30.4	26.3	28.7
Farasa	27.4	30.3	30.2	26.4	28.6

Table 2: MADA versus Farasa Tokenization

2. Data Settings and Pre/Post Processing

We trained our systems using the data made available through IWSLT 2016 campaign. This contained two in-domain data sets TED talks and QED corpus [14] and two out-domain data sets UN corpus [4] and OPUS data [6]. The statistics are shown in Table 1. For language model we trained using the target side of the parallel corpus and all the available English data from the recent WMT campaign [15], and GigaWord and OPUS mono corpus for Arabic.

We segmented Arabic data using both MADAMIRA and Farasa. We found MADAMIRA [16] performed 0.1 BLEU points better than Farasa [17] (See Table 2) and decided to use it for the competition. We tokenized the English side using standard tokenizer of Moses. For English→Arabic, outputs were detokenized using MADA detokenizer. Before scoring the output, we normalized them and reference translations using the QCRI normalizer [5].

3. Phrase-based System

3.1. Baseline Settings

We trained phrase-based Moses system, with the settings described in [18]: a maximum sentence length of 80, Fast-Aligner for word-alignments [19], an interpolated Kneser-Ney smoothed 5-gram language model [20], lexicalized reordering model [21], a 5-gram operation sequence model [22], a 14-gram NNJM model [23], with the baseline settings described in [24]. We used default distortion limit, 100-best translation options, phrase-length of 5, monotone-over-punctuation heuristic, cube-pruning with a limit of 1000 during tuning 5000 during test. We used k-best batch MIRA [25] for tuning. We used cased BLEU [26] to measure progress.

3.2. Data Selection

Due to our experience from previous competitions, we were wary of the fact that simply adding the UN data is harmful for the AR→ MT system, we therefore selected data through MML filtering [8]. We selected 2.5%, 3.75%, 5%, 10% and 30% of the UN data and trained MT pipeline by concatenating the selected data with the in-domain data. We did not include Opus data (40 Million Sentences) and NNJM in these experiments to get the results quickly. Table 3 shows the results. We found 3.75% (≈685k sentences) to be the optimal threshold. Alternative to data selection, we tried training in- and out-domain phrase-tables separately and using the out-domain phrase-table only as a back-off. Second last row of Table 3 shows results. While it gave improvement on top of the baseline system, it was slightly behind MML filtering.

Percentage	ted-11	ted-12	ted-13	ted-14	Avg
ID	27.5	30.6	30.4	26.3	28.7
2.5%	27.3	30.6	31.5	27.0	29.1
3.75%	27.1	30.7	31.6	27.2	29.2
5%	27.1	30.4	31.4	27.1	29.1
10%	27.0	30.2	31.5	27.2	29.0
30%	26.3	29.5	30.9	26.7	28.4
Full	25.3	28.7	29.4	25.5	27.3
Back-off PT	27.0	30.4	31.5	27.3	29.1
3.75%+$\frac{1}{2}$OPUS	28.2	32.4	32.3	28.6	30.4

Table 3: Data Selection using MML and Back-off PT

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	28.2	32.4	32.3	28.6	30.4
+bigLM	28.3	32.8	33.2	29.2	30.9

Table 4: Bigger Language Model

We then tried to find optimal cut-off on the OPUS data, and selected 20 Million sentences (half of the Opus). Our best systems used 3.75% of the UN data and half of the Opus data. Adding the selected Opus data gave an average improvement of +1.2 BLEU points.

3.3. Language Model

We trained bigger language model by using all the available English data from the recent WMT campaign[1] and target-side of the parallel data. A Kneser-Ney smoothed 5-gram language model was trained on each sub-corpus individually and then interpolated to minimize perplexity on the target part of the monolingual data. We were able to obtain a gain of +0.5 using bigger language model. See Table 4.

3.4. Interpolation of Operation Sequence Models

The OSM model has been a regular feature of the phrase-based pipeline in the competition grade systems. It is a joint sequence translation model which integrates reordering. [10] recently found that an OSM model trained on plain concatenation of data is sub-optimal and can be improved by training OSM models on each domain individually and interpolating them by minimizing perplexity on the in-domain tune-set. Table 5 shows that using interpolated OSM model (OSM_i) instead of the one trained on plain concatenation (OSM_c) gives an average improvement of +0.6 BLEU points.

3.5. NNJM Adaptation

We also explored the award winning Neural Network Joint Model (NNJM) in our pipeline and tried to adapt it towards the in-domain data. We trained an NNJM models on the UN and Opus data for 25 epochs and then fine-tuned [12] it by

[1] http://www.statmt.org/wmt16/translation-task.html

System	ted-11	ted-12	ted-13	ted-14	Avg
OSM_c	28.3	32.8	33.2	29.2	30.9
OSM_i	29.0	33.5	33.8	29.7	31.5

Table 5: Interpolated Operation Sequence Model

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	29.0	33.5	33.8	29.7	31.5
+NNJM	29.8	34.1	34.4	30.1	32.1
+FT(UN)	29.7	33.8	33.9	30.2	31.9
+FT(OPUS)	**30.1**	**34.1**	**34.6**	**30.3**	**32.3**

Table 6: Neural Network Joint Model + Different Adaptation Methods

running for 25 more epochs on the in-domain data. Because the data is huge, the entire training took 1.5 months of wall-clock time. Table 6 shows results. The NNJM model gave significant improvement (+0.6) on top of baseline which does not include it. We found fine-tuning method to give slight gains (+0.2) when the baseline model was trained on the Opus data. On the contrary, fine-tuning did not help when the model trained was on UN.

3.6. Class-based Models

We explored the use of automatic word clusters in phrase-based models [11]. We used 50 classes, obtained by running `mkcls`. The clusters ids were included in the phrase-table. We additionally trained in-domain language model using word-classes and interpolated OSM on word-classes. But we only saw very small improvements using word classes.

3.7. Handling Unknown Words

We tried to handle OOV words using `drop-oov` and through transliteration [27, 28]. The former worked slightly better and was used in the best system. Of course the gains from the two methods are additive because they are addressing different OOVs, but there's no good way to automatically find which word to drop and which one to transliterate.

3.8. Final System

Table 9 shows incremental progress on this Arabic→English language pair. Our best system included MML selected UN and Opus corpora, big language model, interpolated OSM and fine-tuned NNJM models. We we used `drop-oov` option to handle unknown words.

3.9. English-to-Arabic Systems

We did not do detailed experiments for the English→Arabic direction because of computational limitations, but simply replicated what worked for the Arabic→English direction.

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	30.1	34.1	34.6	30.3	32.3
Class-based	30.3	34.2	34.7	30.4	32.4

Table 7: Using Word Classes

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	30.3	34.2	34.7	30.4	32.4
Drop-OOV	**30.5**	**34.2**	**35.0**	**30.5**	**32.6**
Transliteration	30.4	34.2	34.7	30.6	32.5

Table 8: Handling OOVs

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	27.4	30.3	30.2	26.4	28.7
+Selected UN	27.1	30.7	31.6	27.2	29.2
+Selected OPUS	28.2	32.4	32.3	28.6	30.4
+bigLM	28.3	32.8	33.2	29.2	30.9
+OSM_i	29.0	33.5	33.8	29.7	31.5
+NNJM	29.8	34.1	34.4	30.1	32.1
+FT(OPUS)	30.1	34.1	34.6	30.3	32.3
Drop-OOV	**30.5**	**34.2**	**35.0**	**30.5**	**32.6**

Table 9: Incremental Progress Arabic-to-English System

Table 10 shows progress on this language pair. The baseline system (ID) was trained on the the TED data and target side of all the permissible parallel data. In the second row, we added all the parallel data except for the UN. In the third row we additionally added the UN data that we selected in the Arabic→English direction. Additional parallel data gives an average improvement of +1.4 BLEU point. Then we added an NNJM model trained on in-domain TED data on top of this system to improve it by +0.8. Adding GigaWord and monolingual OPUS data (another 20M Sentences other than the target-side of the parallel data) gave an improvement of +0.3. Finally we replaced the baseline NNJM with the one trained on OPUS data and fine-tuned with the in-domain data to get our best system.

3.10. QED Systems

We simply replicated QED systems by replacing QED corpus to be in-domain data, instead of TED data. We used the same UN data that we selected for our Arabic→English system, therefore our phrase-tables remain the same. The main changes are caused when training adapted OSM and NNJM models. For NNJM we simply fine tune with QED corpus instead of the TED corpus. For interpolated OSM, we concatenated TED and QED corpus and build OSM on it, which is then interpolated with the OSM models trained on the selected UN and Opus data. We used IWSLT tuning to get the interpolation weights. This way the OSM sub-model

System	ted-11	ted-12	ted-13	ted-14	Avg
ID	14.8	15.6	16.7	14.5	15.4
+Parallel	15.5	16.4	18.2	16.3	16.6
+MML(UN)	15.6	16.3	18.4	16.7	16.8
+NNJM	16.5	17.4	19.2	17.4	17.6
+bigLM	16.6	17.6	20.0	17.4	17.9
+NNJM(FT)	**16.7**	**17.9**	**20.2**	**17.7**	**18.1**

Table 10: Incremental Progress English-to-Arabic System

System	ted-11	ted-12	ted-13	ted-14	Avg
30%+FT(TED)	27.2	31.4	30.8	27.1	29.1
30%+TED+FT(TED)	27.2	30.8	30.1	25.8	28.5

Table 11: Fine Tuning on Out-domain versus Concatenation – Models run for 3 epochs

created from TED+QED corpus gets best weights. We also retrained the language model in this similar fashion. We used the tuning weights obtained from our best TED systems and replaced the TED adapted OSM, NNJM and language models with their QED adapted variants.

4. Neural Machine Translation

4.1. Pre/Post-processing

We used a similar pre/post-processing pipeline for Neural MT as our phrase-based systems (Section 2), and additionally applied BPE [29] before training them. Our BPE models are trained separately for both the Arabic and English datasets instead of jointly training them, since the character set differs between the languages. We limited the number of operations to 59,500, as suggested in [29]. We experimented with BPE models trained on the TED data, and on the concatenation of the TED and out-domain data. We did not see any considerable difference in performance between these models. Thus we used the BPE model trained on the TED data for the experiments reported in this paper.

4.2. Baseline

We used default parameters in Nematus to train our systems: a batch size of 80, source and target vocabulary of 50K entries each, 1024 LSTM units, and the embedding layer size of 500. Baseline system were trained using only TED corpus.

4.3. Fine Tuning on Concatenation versus OD

The best phrase based systems are usually trained by concatenating in and out-domain data. On the other hand, deep learning systems are trained on the out-domain data first, and then fine-tuned with in-domain data. We experimented with both strategies. In the interest of time we selected 30% of the UN data using MML filtering (Table 3). We trained two systems, one by concatenating the in-domain data with the selected (30%) UN data and other just on the selected data. Then we fine-tuned both the models with the in-domain TED data after running them for 3 epochs. Table 11 shows that fine-tuning a system trained on out-domain data only, outperforms the system fine-tuned on concatenation.

4.4. Fine-tuning Variants and Dropouts

The default version of Nematus applies fine-tuning by freezing the weights of embedding layer. The intuition behind freezing a layer is to not allow the weights in that layer to change with additional data. This is sometimes useful when we can learn certain layers better from out-domain data. One such layer in our case is the word embedding layer. We tried a variation in which we do not freeze any layer. This latter variant was found to outperform the default setting (See Table 12).

System	ted-11	ted-12	ted-13	ted-14	Avg
5%	25.8	29.4	29.3	25.0	27.4
5% (Frozen)	24.7	27.7	27.4	23.9	25.9
30%	27.2	30.8	30.1	25.8	28.5
30% (Frozen)	26.5	30.4	28.9	25.0	27.7

Table 12: Fine-tuning with/without freezing the Embeddings

System	ted-11	ted-12	ted-13	ted-14	Avg
5% + FT(TED)	25.8	29.4	29.3	25.0	27.4
30% + FT(TED)	28.4	32.7	32.9	27.8	30.4
Full + FT(TED)	28.1	32.3	31.6	27.0	29.8
30% + FT(OPUS)	26.1	30.6	32.5	27.1	29.1
Full + FT(OPUS)	**28.2**	**31.7**	**34.3**	**29.2**	**30.8**

Table 13: Data selection

Dropouts are found to be useful in NN training, when the training data is small. We experimented with using dropouts in our experiments, but did not find any significant difference. Hence we decided to use it only when fine-tuning with the in-domain data (TED/QED), since both of the other datasets (UN and OPUS) were big and did not pose any risk of inducing the problem of overfitting.

4.5. Data Selection

Since we found data selection useful in the phrase based system, we also trained our neural systems using 5%, 30% and 100% of the UN data. In these experiments, we concatenated the 5% and 30% of the UN data with the in-domain data. To evaluate the most promising models, we trained all of the models until the learning plateaued, and then fine-tuned these models with in-domain data.[2] The results are shown in in Table 13. Using only 5% of the data proved harmful, and the system did not generalize as well as the other models. The model trained on 30% of the data performed better than the model trained on all the data, by 0.7 BLEU points.

In our subsequent experiments we tried to verify if this finding holds when we add the OPUS data. We therefore trained two systems by fine-tuning 30% selected UN data or full UN data using OPUS. Here the results flipped and the we found that model that used all of the UN data performed better (Compare last two rows in Table 13). Therefore, we decided to focus our efforts on the model trained on the entire UN data for all of the following experiments.

4.6. Ensemble

Ensembling models has shown to give a consistent boost in performance in past best performing systems [3]. We therefore experimented with several variations. We found the best

[2]Because we were running experiments in parallel, we were not aware at this point that fine-tuning on out-domain is a better strategy

System	ted-11	ted-12	ted-13	ted-14	Avg
Full + FT(OPUS)	28.2	31.7	34.3	29.2	30.8
+ FT(TED)	31.8	36.2	36.1	30.8	33.7
Ensemble (8)	32.5	37.0	37.2	31.5	34.6

Table 14: Ensembling over 8 Fine-tuned Models

System	ted-11	ted-12	ted-13	ted-14	Avg
Baseline	24.0	26.4	25.2	22.4	24.5
UN	15.9	17.9	20.0	16.3	17.5
+OPUS	28.2	31.7	34.3	29.2	30.8
+TED	31.8	36.2	36.1	30.8	33.7
+Ensemble	**32.5**	**37.0**	**37.2**	**31.5**	**34.6**

Table 15: Arabic-to-English NMT System progress

performing combination by fine-tuning the last eight models of the UN+OPUS system, and then ensemble these eight fine-tuned models. Performance improvements from the ensemble are shown in Table 14. The second row shows systems when we fine tune our best system in Table 13 with the in-domain TED data. In the last row we perform ensemble.

4.7. Final System

Our final system was trained by first using all of the UN data. We then continued training on OPUS data. Once learning had plateaued on the OPUS data, we took the last eight models which were very similar in performance, and fine-tuned each of the them using TED data. We then combined these eight fine-tuned models in an ensemble as our final system. The progress is shown in Table 15. We used the same strategy for the QED systems by fine-tuning the last eight OPUS models with QED data, and combining these in an ensemble.

4.8. English-to-Arabic Systems

We used insights gained from our Arabic-to-English system experiments to train our English→Arabic systems. Our final model for both TED and QED was first trained on all of the UN data, followed by the OPUS data, and finally fine-tuned with the in-domain data. The progress is shown in Table 16.

5. System Combination

We combined hypotheses produced by our best Phrase-based and Neural MT systems. For this purpose we used Multi-Engine MT system, or MEMT [13]. The results are shown in Table 17. We did not gain any substantial improvements using system combination. Small improvements were obtained in the Arabic→English direction baring test-2012. On

System	ted-11	ted-12	ted-13	ted-14	Avg
UN	9.1	9.3	11.2	9.4	9.8
+OPUS	10.8	11.2	13.4	10.9	11.6
+TED	**17.1**	**18.9**	**20.1**	**17.7**	**18.5**

Table 16: English-to-Arabic NMT System progress

System	ted-11	ted-12	ted-13	ted-14	Avg
Arabic →English					
Phrase-based	30.5	34.2	35.0	30.5	32.6
Neural MT	32.5	37.0	37.2	31.5	34.6
System Comb	32.8	36.5	37.4	31.7	34.6
English →Arabic					
Phrase-based	16.7	17.9	20.2	17.7	18.1
Neural MT	17.1	18.9	20.1	17.7	18.5
System Comb	16.8	19.1	20.7	17.6	18.6

Table 17: Results for System Combination

System	ted-15	ted-16	qed-15
Arabic →English			
Primary	34.1	31.8	28.1
Contrastive	33.7	31.5	28.1
English →Arabic			
Primary	19.5	18.4	23.1
Contrastive	19.5	18.1	22.9

Table 18: Results on Official Test Sets

the contrary significant improvement was obtained only in test-2013 in the English→Arabic direction. Table 18 shows results on the official test-sets.

6. Summary

We trained a very strong phrase-based system with SOTA features such as OSM, NNJM and big LM. The system improved greatly by applying domain adaptation. To this end we applied MML-based filtering, interpolated OSM and fine-tuning of NNJM models. Overall, our phrase-based system achieved a gain of 4 BLEU points on top of the baseline system. We also applied data selection for training our NMT. However, the NMT systems quickly overfit and did not perform well. Our experiments showed that the NMT system trained on the full UN data performed best, and the final NMT system made use of all the available out-of-domain data. However, the training was performed incrementally, starting with UN data for 50k iterations, fine tuned on OPUS for 25k more iterations and then fine tuned the final model using TED talks for a few iterations. We simply replicated our settings to train QED systems. Finally we applied system combination of the two systems using MEMT.

While it is computationally expensive, we found training a neural MT system much simpler than a competitive phrase-based system, where a lot of sub-components need to be optimized independently to reach the best configuration. On the contrary, an NMT system requires least supervision. Secondly once a neural system is trained, the effort can be easily reused to adapt the system towards another domain, as in this case we simply fine-tuned our UN+OPUS system with the QED corpus. On the contrary, almost all the sub-component of a phrase-based system had to be retrained to adapt the system towards QED corpus.

7. 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. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/pdf/1409.0473v6.pdf

[3] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for wmt 16," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 371–376. [Online]. Available: http://www.aclweb.org/anthology/W16-2323

[4] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1.0," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*, 2016.

[5] H. Sajjad, F. Guzmn, P. Nakov, A. Abdelali, K. Murray, F. A. Obaidli, and S. Vogel, "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 2013.

[6] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), may 2016.

[7] A. Abdelali, F. Guzman, H. Sajjad, and S. Vogel, "The AMARA corpus: Building parallel language resources for the educational domain," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May 2014.

[8] 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*, ser. EMNLP '11, Edinburgh, United Kingdom, 2011.

[9] N. Durrani, A. Fraser, and H. Schmid, "Model with minimal translation units, but decode with phrases," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*, Atlanta, Georgia, USA, 2013.

[10] N. Durrani, H. Sajjad, S. Joty, A. Abdelali, and S. Vogel, "Using joint models for domain adaptation in statistical machine translation," in *Proceedings of the Fifteenth Machine Translation Summit (MT Summit XV)*. Florida, USA: AMTA, November 2015.

[11] N. Durrani, P. Koehn, H. Schmid, and A. Fraser, "Investigating the usefulness of generalized word representations in smt," in *Proceedings of the 25th Annual Conference on Computational Linguistics*, ser. COLING'14, Dublin, Ireland, 2014, pp. 421–432.

[12] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domain," in *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

[13] K. Heafield and A. Lavie, "CMU system combination in WMT 2011," in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 145–151. [Online]. Available: http://kheafield.com/professional/avenue/wmt_2011.pdf

[14] F. Guzmán, H. Sajjad, S. Vogel, and A. Abdelali, "The AMARA corpus: Building resources for translating the web's educational content," in *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, December 2013.

[15] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri, "Findings of the 2016 conference on machine translation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 131–198. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2301

[16] A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. M. Roth, "MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic," in *Proceedings of the Language Resources and Evaluation Conference*, ser. LREC '14, Reykjavik, Iceland, 2014, pp. 1094–1101.

[17] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for arabic," in *Proceedings of the 2016 Conference of the*

North American Chapter of the Association for Computational Linguistics: Demonstrations. San Diego, California: Association for Computational Linguistics, June 2016, pp. 11–16. [Online]. Available: http://www.aclweb.org/anthology/N16-3003

[18] A. Birch, M. Huck, N. Durrani, N. Bogoychev, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2014 evaluation," in *Proceedings of the 11th International Workshop on Spoken Language Translation*, ser. IWSLT '14, Lake Tahoe, CA, USA, 2014.

[19] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *Proceedings of NAACL'13*, 2013.

[20] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197. [Online]. Available: http://kheafield.com/professional/avenue/kenlm.pdf

[21] M. Galley and C. D. Manning, "A Simple and Effective Hierarchical Phrase Reordering Model," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, October 2008, pp. 848–856. [Online]. Available: http://www.aclweb.org/anthology/D08-1089

[22] N. Durrani, H. Schmid, A. Fraser, P. Koehn, and H. Schütze, "The Operation Sequence Model – Combining N-Gram-based and Phrase-based Statistical Machine Translation," *Computational Linguistics*, vol. 41, no. 2, pp. 157–186, 2015.

[23] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014.

[24] S. Joty, H. Sajjad, N. Durrani, K. Al-Mannai, A. Abdelali, and S. Vogel, "How to Avoid Unwanted Pregnancies: Domain Adaptation using Neural Network Models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015.

[25] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. NAACL-HLT '12, Montréal, Canada, 2012.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Morristown, NJ, USA, 2002, pp. 311–318.

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

[28] N. Durrani, H. Sajjad, H. Hoang, and P. Koehn, "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 2014.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725. [Online]. Available: http://www.aclweb.org/anthology/P16-1162