QCRI Live Speech Translation System

Fahim Dalvi, Yifan Zhang, Sameer Khurana, Nadir Durrani, Hassan Sajjad
Ahmed Abdelali, Hamdy Mubarak, Ahmed Ali, Stephan Vogel
Qatar Computing Research Institute
Hamad bin Khalifa University, Doha, Qatar
{faimaduddin, yzhang}@qf.org.qa

Abstract

We present QCRI's Arabic-to-English speech translation system. It features modern web technologies to capture live audio, and broadcasts Arabic transcriptions and English translations simultaneously. Our Kaldi-based ASR system uses the Time Delay Neural Network architecture, while our Machine Translation (MT) system uses both phrase-based and neural frameworks. Although our neural MT system is slower than the phrase-based system, it produces significantly better translations and is memory efficient.[1]

1 Introduction

We present our Arabic-to-English SLT system consisting of three modules, the Web application, Kaldi-based Speech Recognition culminated with Phrase-based/Neural MT system. It is trained and optimized for the translation of live talks and lectures into English. We used a Time Delayed Neural Network (TDNN) for our speech recognition system, which has a word error rate of 23%. For our machine translation system, we deployed both the traditional phrase-based Moses and the emerging Neural MT system. The trade-off between efficiency and accuracy (BLEU) barred us from picking only one final system. While the phrase-based system was much faster (translating 24 tokens/second versus 9.5 tokens/second), it was also roughly 5 BLEU points worse (28.6 versus 33.6) compared to our Neural MT system. We therefore leave it up to the user to decide whether they care more about translation quality or speed. The *real-time* factor for the entire pipeline is 1.18 using Phrase-based MT and 1.26 using Neural MT.

[1]The demo is available at https://st.qcri.org/demos/livetranslation.

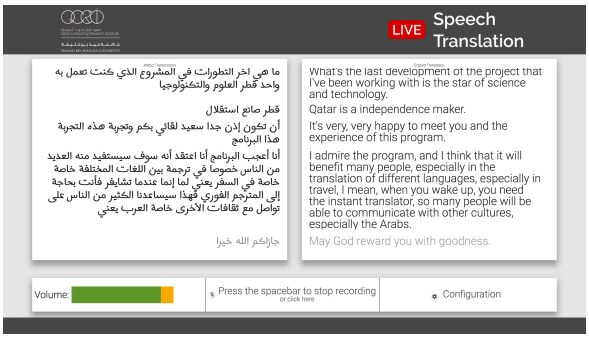

Figure 1: Speech translation system in action. The Arabic transcriptions and English translations are shown in *real-time* as they are decoded.

Our system is also robust to common English code-switching, frequent acronyms, as well as dialectal speech. Both the Arabic transcriptions and the English translations are presented as results to the viewers. The system is built upon modern web technologies, allowing it to run on any browser that has implemented these technologies. Figure 1 presents a screen shot of the interface.

2 System Architecture

The QCRI live speech translation system is primarily composed of three fairly independent modules: the web application, speech recognition, and machine translation. Figure 2 shows the complete work-flow of the system. It mainly involves the following steps: 1) Send audio from a broadcast instance to the ASR server; 2) Receive transcription from the ASR server; 3) Send transcription to MT server; 4) Receive translation from the MT server; 5) Sync results with backend system and 6) Multiple watch instances sync results from the backend. Steps 1-5 are constantly repeated as new audio is received by the system through the broadcast page. Step 6 is also periodically repeated to get the latest results on the watch page. Both the speech recognition and machine translation mod-

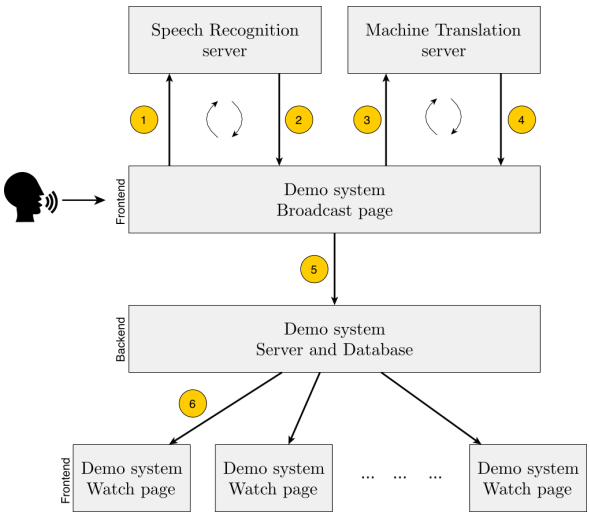

Figure 2: Demo system overview

ules have a standard API that can be used to send and receive information. The *Web Application* connects with the API and runs independently of the system used for transcription and translation.

2.1 Web application

The web application has two major components; the frontend and the backend. The frontend is created using the React Javascript framework to handle the dynamic User Interface (UI) changes such as transcription and translation updates. The backend is built using NodeJS and MongoDB to handle sessions, data associated with these sessions and authentication. The frontend presents the user with three pages; the landing page, the watch page and the broadcast page. The landing page allows the user to either create a new session or work with an existing one. The watch page regularly syncs with the backend to get the latest partial or final transcriptions and translations. The broadcast page is meant for the primary speaker. This page is responsible for recording the audio data and collecting the transcriptions and translations from the ASR and MT systems respectively. Both partial transcriptions and translations are also presented to the speaker as they are being decoded. To avoid very frequent and abrupt changes, the rate of update of partial translations was configured based on a `MIN_NEW_WORDS` parameter, which defines the minimum number of new words required in the partial transcription to trigger the translation service. Both the partial and the final results are also synced to the backend as they are made available, so that the viewers on the watch page can experience the live translation.

2.2 Speech transcription

We use the Speech-to-text transcription system that was built as part of QCRI's submission to the 2016 Arabic Multi-Dialect Broadcast Media Recognition (MGB) Challenge. Key features of the transcription system are given below:

Data: The training data consisted of 1200 hours of transcribed broadcast speech data collected from Aljazeera news channel. In addition we had 10 hours of development data (Ali et al., 2016). We used data augmentation techniques such as Speed and Volume perturbation which increased the size of the training data to three times the original size (Ko et al., 2015).

Speech Lexicon: We used a Grapheme based lexicon (Killer and Schultz, 2003) of size 900k. The lexicon is constructed using the words that occur more than twice in the training transcripts.

Speech Features: Features used to train all the acoustic models are 40 dimensional hi-resolution Mel Frequency Cepstral Coefficients (MFCC_hires), extracted for each speech frame, concatenated with 100 dimensional i-Vectors per speaker to facilitate speaker adaptation (Saon et al., 2013).

Acoustic Models: We experimented with three acoustic models; Time Delayed Neural Networks (TDNNs) (Peddinti et al., 2015), Long Short-Term Memory Recurrent Neural Networks (LSTM) and Bi-directional LSTM (Sak et al., 2014). Performance of the BLSTM acoustic model in terms of *Word Error Rate* is better than the TDNN, but TDNN has a much better *real-time* factor while decoding. Hence, for the purpose of the speech translation system, we use the TDNN acoustic model. The TDNN model consists of 5 hidden layers, each layer containing 1024 hidden units and is trained using Lattice Free Maximum Mutual Information (LF-MMI) modeling framework in Kaldi (Povey et al., 2016). Word Error Rate comparison of different acoustic models can be seen in Table 1. For further details, see Khurana and Ali, 2016).

Language Model: We built a Kneser Ney smoothed trigram language model. The vocab size is restricted to the 100k most frequent words to improve the decoding speed and the *real-time factor* of the system. The choice of using a trigram model instead of an RNN as in our offline systems was essential in keeping the decoding speed at a reasonable value.

Decoder Parameters: Beam size for the de-

Model	%WER
TDNN	23.0
LSTM	20.9
BLSTM	19.3

Table 1: Recognition results for the LF-MMI trained recognition systems. LM used for decoding is tri-gram. Data augmentation is used before training

coder was tuned to give the best real-time factor with a reasonable drop in accuracy. The final value was selected to be 9.0.

2.3 Machine translation

The MT component is served by an API that connects to several translation systems and allows the user to seamlessly switch between them. We had four systems to choose from for our demo, two of which were Phrase-based systems, and the two were Neural MT systems trained using Nematus (Sennrich et al., 2016).

PB-Best: This is a competition-grade phrase-based system, also used for our participation at the IWSLT'16 campaign (Durrani et al., 2016). It was trained using all the freely available Arabic-English data with state-of-the-art features such as a large language model, lexical reordering, interpolated OSM (Durrani et al., 2013) and NNJM features (Devlin et al., 2014).

PB-Pruned: The PB-best system is not suitable for real time translation and has high memory requirements. To increase the efficiency, we dropped the OSM and NNJM features, heavily pruned the language model and used MML-filtering to select a subset of training data. The resulting system was trained on 1.2 M sentences, 10 times less the original data.

NMT-GPU: This is our best system[2] that we submitted to the IWSLT'16 campaign (Durrani et al., 2016). The advantage of Neural models is that their size does not scale linearly with the data, and hence we were able to train using all available data without sacrificing translation speed. This model runs on the GPU.

NMT-CPU: This is the same model as 3, but runs on the CPU. We use the AmuNMT (Junczys-Dowmunt et al., 2016) decoder to use our neural models on the CPU. Because of computation constraints, we reduced the beam size from 12 to 5 with a minimal loss of 0.1 BLEU points.

The primary factors in our final decision were 3-fold; overall quality, translation time and computational constraints. The translation time has to be small for a live translation system. The performance of the four systems on the official IWSLT test-sets is shown in Figure 3.

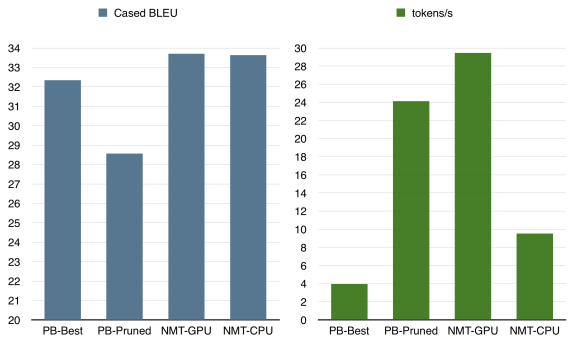

Figure 3: Performance and Translation speed of various MT systems

We also computed the translation speed of each of the systems.[3] The results shown in Figure 3 depict the significant time gain we achieved using the pruned phrase based system. However, with a 5 BLEU point difference in translation quality, we decided to compromise and use the slower **NMT-CPU** in our final demo. We also allow the user to switch to the phrase-based system, if translation speed is more important. We did not use **NMT-GPU** since it is very costly to put into production with its requirement for a dedicated GPU card.

Finally, we added a customized dictionary and translated unknown words by transliterating them in a post-decoding step (Durrani et al., 2014).

2.4 Combining Speech recognition and Machine translation

To evaluate our complete pipeline, we prepared three in-house test sets. The first set was collected from an in-house promotional video, while the other two sets were collected in a quiet office environment.

[2]without performing ensembling

[3]**PB-Pruned** and **NMT-CPU** were run using a single CPU thread on our standard demo machine using a `Intel(R) Xeon(R) E5-2660 @ 2.20GHz` processor. **PB-Best** was run on another machine using a `Intel(R) Xeon(R) E5-2650 @ 2.00GHz` processor due to memory constraints. Finally, **NMT-GPU** was run using an `NVidia GeForce GTX TITAN X` GPU card.

We analyzed the real time performance of the entire pipeline, include the lag induced by translation after the transcription is complete. With an average *real-time factor* of 1.1 for the speech recognition, our system keeps up with normal speech without any significant lag. The distribution of the real time factor speech recognition and translation for the in-house test sets is shown in Figure 4.

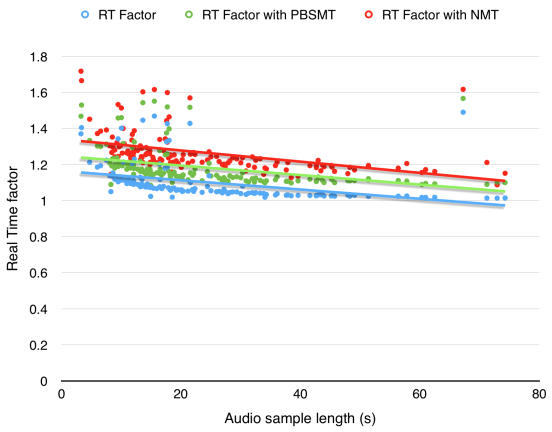

Figure 4: Real-time analysis for all audio segments in our in-house test sets

3 Conclusion

This paper presents QCRI live speech translation system for real world settings such as lectures and talks. Currently, the system works very well for Arabic including frequent dialectal words, and also supports code-switching for most common acronyms and English words. Our future aim is to improve the system in several ways; by having a tighter integration between the speech recognition and translation components, incorporating more dialectal speech recognition and translation, and by improving punctuation recovery of the speech recognition system which will help machine translation to produce better translation quality.

References

Ali, A., Bell, P., Glass, J., Messaoui, Y., Mubarak, H., Renals, S., and Zhang, Y. (2016). The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. In *SLT*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). 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)*.

Durrani, N., Dalvi, F., Sajjad, H., and Vogel, S. (2016). QCRI machine translation systems for IWSLT 16. In *Proceedings of the 15th International Workshop on Spoken Language Translation*, IWSLT '16, Seattle, WA, USA.

Durrani, N., Fraser, A., Schmid, H., Hoang, H., and Koehn, P. (2013). Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 399–405, Sofia, Bulgaria. Association for Computational Linguistics.

Durrani, N., Sajjad, H., Hoang, H., and Koehn, P. (2014). Integrating an unsupervised transliteration model into statistical machine translation. In *EACL*, volume 14, pages 148–153.

Junczys-Dowmunt, M., Dwojak, T., and Hoang, H. (2016). Is neural machine translation ready for deployment? A case study on 30 translation directions. *CoRR*, abs/1610.01108.

Khurana, S. and Ali, A. (2016). QCRI advanced transcription system (QATS) for the arabic multi-dialect broadcast media recognition: MGB-2 challenge. In *Spoken Language Technology Workshop (SLT) 2016 IEEE*.

Killer, M. and Schultz, T. (2003). Grapheme based speech recognition. In *INTERSPEECH*.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (2015). Audio augmentation for speech recognition. In *INTERSPEECH*.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*, pages 3214–3218.

Povey, D., Peddinti, V., Galvez, D., Ghahrmani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for asr based on lattice-free mmi.

Sak, H., Senior, A. W., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *ASRU*, pages 55–59.

Sennrich, R., Haddow, B., and Birch, A. (2016). Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.