1 Supplemental Material

1.1 Translator Baseline

In this paper, we use translator model as baseline to compare with our cross-lingual transfer algorithm. Here we adopt the open-source toolkit for neural machine translation (NMT) in openNMT[1] and train this model on the IWSLT2014 english-to-german and english-to-italian translation dataset. Our training data consists of 178K sentence pairs and validation data consists of 2K sentence pairs. We train the NMT model for roughly five epochs until it converges on the validation dataset. The source and target word vector size are both set to 500, the attention-based RNN hidden size for encoder and decoder are also both 500. We adopt the input feeding strategy to further boost the performance and apply beam size of 8 to search for the most probable target candidate. We leave the UNKs out during beam search to reduce noises. Here we first show the results of our trained model against some state-of-art algorithms to verify that our trained translation model makes sense: We can see from Table 1 that our pre-trained model is able to achieve reasonable results on the common translation dataset. Here we demonstrate some examples for translated dialogs:

Model	BLEU
Actor-Critic (Bahdanau et al., 2016)	28.53
BSO (Wiseman and Rush, 2016)	26.63
Variational Attention (Deng et al., 2018)	33.09
Our OpenNMT (Klein et al., 2017)	30.2

Table 1: The test set results on IWSLT for different algorithms

- Human(**correct**): gibt es irgendwelche eritrean restaurants in die stadt? (is there any eritrean restaurants in the town?)

- Machine: no, there are no eritrean restaurants in town. would you like a different restaurant?

- Human(**missing**): Wie steht es mit essen? (Is there any ~~Chinese~~ food?)

- Machine: there is a wide variety of chinese restaurants, do you have an area preference or a price preference to narrow it down?

- Human(**correct**): ich wrde den osten der stadt mgen. (I want to try something in the east)

- Machine: yu garden is a chinese restaurant in the east area.

- Human(**correct**): knnte ich die adresse bekommen, und postcode garden? (Can I obtain the address and postcode of the restaurant?)

- Machine: phone is xxx-xxxx, address and postcode are xxx new market road

Another example is shown below:

- Human(**missing**): ich mchte ein restaurant in jedem gebiet finden , und serves essen? (I want to find a restaurant anywhere to serve ~~Scottish~~ food?)

- Machine: there are no restaurants serving scottish food . would you like to try something else?

- Human(**correct**): ja, wie steht es mit modernen europischen essen? (how about moderate european food?)

- Machine: eraina is a great european eatery. what other information can i provide to you?

- Human(**correct**): was ist das telefon? (What is its telephone?)

- Machine: their phone number is numnum may i help you with anything else?

- Human(**correct**): nein, vielen dank. (no, thanks.)

The above generated German dialogs are shown with their English translation to help readers to understand. We have observed that the most frequent two error types are the entity missing and broken grammar. Since IWSLT2014 lies in another domain, its vocabulary is mis-aligned with the spoken dialog vocabulary. Therefore, the NMT decoder fails to generate many ontology words. These errors in the translated dialog corpus can remarkably degrade the DST performance.

[1] https://github.com/OpenNMT/OpenNMT-py

1.2 NBT system setting

In our implementation, we set the word embedding size to be $H = 300$ for both monolingual and bilingual embedding. We directly use the ontology's corresponding lexicon embedding as its own embedding. Therefore, the ontology embedding also shares the same dimension. We restrict the length of utterance to be a maximum of 40 and filter the utterances beyond that length during training. Our CNN has three independent filter size (1, 2, 3) to extract unigram, bigram and trigram information out of the utterance. The intermediate feature dimension of CNN is also set to 300, finally we add the three filters to construct the utterance representation. We also apply dropout strategy after the three gates.

1.3 Learning Curve

Here we demonstrate the learning curve for XL-NBT-D in Figure 1 and XL-NBT-C in Figure 2.

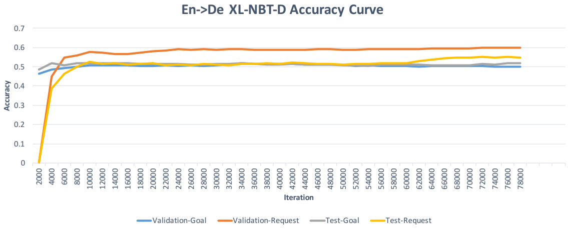

Figure 1: The learning curve for Transfer Learning (XL-NBT-D).

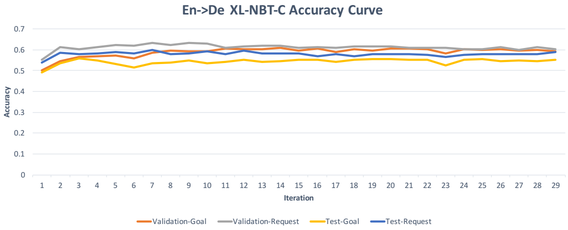

Figure 2: The learning curve for Transfer Learning (XL-NBT-C).

The rise of our transfer learning is very steady, we average multiple runs as our final reported score in the paper.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M Rush. 2018. Latent alignment and variational attention. *arXiv preprint arXiv:1807.03756*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*.