KIT's IWSLT 2018 SLT Translation System

Matthias Sperber, Ngoc Quan Pham, Thai Son Nguyen, Jan Niehues,
Markus Müller, Thanh-Le Ha, Sebastian Stüker, Alex Waibel

Institute for Antrophomaics and Robotics, Karlsruhe Institut of Technology

firstname.lastname@kit.edu

Abstract

This paper describes KIT's submission to the IWSLT 2018 Translation task. We describe a system participating in the baseline condition and a system participating in the end-to-end condition. The baseline system is a cascade of an ASR system, a system to segment the ASR output and a neural machine translation system. We investigate the combination of different ASR systems. For the segmentation and machine translation components, we focused on transformer-based architectures.

1. Introduction

The Karlsruhe Institute of Technology participated in the IWSLT 2018 Evaluation Campaign with systems for English→German Speech Translation task. We submitted system to both conditions: the baseline condition and the end-to-end condition.

The submission to the baseline condition is based on the cascaded approach described in [1]. In this evaluation campaign, we investigated the combination of different ASR systems. Furthermore, we investigated the use of transformer-based models.

This paper is structured as follows. In Section 2, we describe different speech recognition systems we employed in the campaign and how we combined them. Afterwards, we give a detailed description of the segmentation approach in Section 3 and the machine translation system in Section 4. Finally, in Section 5 we describe the end-to-end speech translation model. At the end of the paper we report the results and finish with a conclusion.

2. Speech Recognition

In this year's evaluation, we built three different types of automatic speech recognition systems. All the systems were trained using the data from the `TED-LIUM` Corpus version 2 [2].

2.1. Hybrid Model

Different from previous years, this year we built only one single HMM-based hybrid model for the speech recognition task. The hybrid acoustic modelling is constructed by stacking 5 LSTMs layers of 320 units, a projection layer of 200 units and a softmax layer to classify 8000 context-dependence phone (CD-Phone) states. As traditional approach, we used Viterbi forced alignment to provide CD-Phone state labels for the training data and the acoustic model was trained using only cross-entropy loss function.

For model training, we use SGD with an initial learning rate of 0.004 for 8 epochs and degrade it with a factor of 0.8 for other 8 epochs. We use a momentum term of 0.9 while dropout is set to 10%. Only 40 features of Mel-filterbank coefficients are fed into the LSTMs network every timestep, we did not employ any further speaker adaptation features.

After the model was successfully trained, we performed the traditional beam search decoding with the employment of the 4-gram language model. We used Janus Recognition Toolkit (JRTK) [3] as the decoding framework while the language model is built by Cantab research group [4] from WMT data.

2.2. CTC Model

Our CTC-based [5] ASR model is similar to the system described by [6]. The input to the model are 40-dimensional Mel-filterbank coefficients. We used every third speech feature of our input sequence and randomly chose the start offset during training, which has the advantage of a lower input sequence length. We trained the model to predict Byte-Pair Units, also referred to as Byte-Pair Encoding (BPE) [7].

The CTC-based model consists of four bidirectional LSTM layers with 400 units in each direction followed by a softmax layer. The size of the softmax layer depends on the number of BPE units we created. We used a dropout rate of 0.25 for all LSTM layers. We trained two models based on BPE units with 300 (small model) and 10,000 (big model) merges, respectively.

We used SGD with a learning rate of 0.0005 and a momentum term of 0.9 for training. The learning rate is halved whenever the validation token error rate does not decrease by more than 0.1%. We first trained the small model and initialized the parameters of the big model's BiLSTM layers using the smaller model's ones. We decoded the model by greedily selecting the most likely output at each time step.

2.3. Encoder-Decoder Model

Our attentional ASR model follows the listen-attend-spell [8] architecture and is similar to the system described by [9]. The model is implemented with XNMT. Compared to a conventional neural machine translation architecture, we replace the encoder with a 4-layer bidirectional pyramidal encoder with a total downsampling factor of 8. The layer size is set to 512, the target embedding size is 64, and the attention uses an MLP of size 128. Input to the model are Mel-filterbank features with 40 coefficients. For regularization, we apply variational dropout of rate 0.3 in all LSTMs, and word dropout of rate 0.1 on the target side [10]. We also fix the target embedding norm to 1 [11]. For training, we use Adam [12] with initial learning rate of 0.0003, which is decayed by factor 0.5 if no improved WER is observed. To further facilitate training, label smoothing [13] is applied. For the search, we use beam size 20 and length normalization with the exponent set to 1.5.

2.4. Rover

To combine the outputs of the different ASR systems, we used ROVER [14]. It operates on the final system output, the CTM-files. Multiple merging strategies exist to combine the outputs based on a majority vote. It is, e.g. possible to take confidences into account to further fine-tune the merging process. The key idea of ROVER is that different systems tend to produce different errors, but that no two systems produce the same error. The more different the systems those outputs are combined are, the better the result will be. Systems being very similar on the other hand will not benefit much from the system combination as they are likely to generate the same errors.

We here combined three different architectures: a traditional HMM-based ASR system, a RNN/CTC based one and an encoder-decoder based one. Hence, combining the outputs improved the WER due to the diversity of the system architectures.

3. Segmentation

Automatic speech recognition (ASR) systems typically do not generate punctuation marks or reliable casing. Using the raw output of these systems as input to MT causes a performance drop due to mismatched train and test conditions. To create segments and better match typical MT training conditions, we use a monolingual NMT system to add sentence boundaries, insert proper punctuation, and add case where appropriate before translating [15].

The idea of the monolingual machine translation system is to translate from lower-cased, unpunctuated text into text with case information and punctuation. Since we do not have any information about the sentence boundaries when inserting the punctuation and case information, we also remove them from the training data. Therefore, in the first step of the pre-processing, we randomly segment the source corpus of the training data into chunks of 20 to 30 words. Based on this randomly segmented corpus, we build the input and output data for the monolingual translation system.

For the input data, we remove all punctuation marks and lowercase all words. Since we will get lower-cased input, we cannot use the same byte-pair encoding [7] as for the machine translation system. Therefore, we train a separate byte-pair encoding on the lower-cased source data with a code size of 40k. To summarize, the source sequence consists of lower-cased BPE units without any punctuation.

For the target side, we do not want to change the words in the output sentence, but only add case and punctuation information. Therefore, we replace the sentence by features indicating case with punctuation attached. Every word is replaced by a letter *U* or *L*, whether it is upper-cased or lower-cased. Furthermore, punctuation marks following the word are directly attached to the letter.

At test time, we follow the sliding window technique described by [16]. Therefore, we created a test set with segments of length 10 starting with every word on the input data. This means, that except for the beginning and the end of the document, every word occurs ten times, at all positions within the segment. This of course dramatically increases the number of sentences in the test data. In the second step, we generate the target features by applying the monolingual translation system. In a post-processing step, we case the word as it most frequently occurs in the output. We insert punctuation marks, if there is at least one punctuation mark after the word in one of the 10 segments containing this word. If different punctuation marks are predicted, we take the most frequent one. Finally, if the punctuation mark is an end of sentence punctuation mark {*"."*,*"!"*,*"?"*}, we also start a new segment. The segmented test data with case and punctuation information is passed on to the machine translation system.

This year, we used a transformer-based NMT system to generated the punctuation marks.[1]. For the encoder and decoder we used 12 layers each using a hidden size of 512 and an inner size of the transformer model of 1024. We applied dropout and trained the models using adam. We first trained the system on the source side of the parallel data. We used the EPPS corpus, NC corpus and a filtered version of the paraCrawl corpus. In a second step, we fine-tuned the model on the TED corpus.

4. Machine Translation

Data preprocessing Our training data, while consisting the TED Talks provided by the evaluation campaign, also includes the following corpora: Europarl (1.8M sentences), News Commentary(280K), Rapid (1.2M), Common crawl (2.2M), the backtranslation data from University of Edinburgh (3.M) and the Paracrawl data (30M). The Paracrawl data is filtered by training a translation model to identify sentences with low likelihood. The final data size is around 36M

[1] https://github.com/isl-mt/NMTGMinor

sentences, with basic preprocessing steps being truecasing, tokenizing, and BPE splitting with BPE size of 40K. The development set is newstest2013 to newstest2016 from the WMT datasets for training the big models. In the adaptation phase, we use the TED talks of dev2010 to validate our models.

Modeling Our translation models are constructed with self-attention encoders and decoders, known as the Transformer networks, following the work of [17]. In this work, we extend the depth of the standard Tranformers, thanks for the residual design combined with layer normalization schemes allowing gradients to flow smoothly. Thanks to the huge amount of data as shown above, we were able to train models up to 32 blocks and still yield meaningful improvement.

Our hyper-parameters of the Transformer models (except depth) follow the *Base* configuration of the original work. The layer size for hidden layers is 512, while the the inner size of feed-forward network inside each block is 2048. The attention layers (including self-attention and attention between decoders and encoders) are multi-head attention layers with 8 heads. We also added label smoothing to regularize the cross-entropy loss. For the network depth, we trained models consisting of 4, 8, 6, 12, 16 and 32 blocks (for both encoder and decoder). Not only are deep models very demanding in terms of computation, they also consume a considerate amount of memory. In order to make training feasible, we used the checkpointing technique [18] by employ re-computation of the network activations during the backward pass to reduce the memory cost for the models.

Training procedure We group mini-batches to fill up our GPU's memory depending on the network size. 12-layer models can fit the memory with batch size containing 2048 words, while deeper models requires batch size reduction to avoid out-of-memory. For updating the networks' parameters, we accumulate gradients up to 25000 target words before doing an update. The learning rate is scheduled as in [17] but we doubled the initial learning rate and extend the warm up duration to 8000 steps. All models including the $32-layer$ config train with 100000 updates. Each model, except the 32-layer one has an additional variation with dropout (added to the residual connection and the inner feed-forward hidden layers).

Domain adaptation After training on all datasets, we further fine tune each model on the TED Talks specifically. Such technique is known to improve the model's performance greatly on the specifically adapted domain [19].

Noise adaptation Since the ASR output is fundamentally different than the collected natural data, we apply a noise model [20] on the TED training data which randomly replace words by sampling. The model is further fine-tuned on the noisy data in the same fashion as domain adaptation.

Final models The output is generated from the ensemble of five models: 12-layer, 12-layer with dropout, 16-layer, 16-layer with dropout and 32-layer.

5. End-to-End Models

We extend the attentional ASR described under Section 2.3 to perform translation by replacing the source-language target tokens by tokens from the target language. We use a refined encoder that performs downsampling to make memory requirements manageable and adds depth for improved accuracy, following one of the variants described by [21]: we stack several blocks consisting of a bidirectional LSTM, a network-in-network (NiN) projection, and batch normalization. After the last block, we add a final bidirectional LSTM layer. NiN denotes a simple linear projection applied at every time step, performing downsampling by concatenating pairs of adjacent projection inputs.

For better results and to be able to include the TEDLIUM corpus in the training, we devise a multi-task training strategy that trains auxiliary models on related tasks while sharing a subset of the parameters with the main ST model. Precisely, besides the main ST task we include an ASR task that shares encoder and attention, an MT task that shares attention and decoder, and an transcript auto-encoder task that shares only the attention. The ASR task is trained on the TEDLIUM corpus, whereas the other tasks are trained the respective subset of the 3-way TED corpus provided for the IWSLT 2018 evaluation. It should be noted that we did not put any efforts into cleaning this data, despite it being relatively noisy.

6. Experiments

We evaluated the models presented in the last section on the provided test sets. Note that for all experiments we used the audio segmentation tool [22] provided in the IWSLT docker container.

6.1. Cascaded

Our experiments involve different configurations regarding three main components in the cascade. For the ASR component, we present three different setups: the ROVER combination of two CTC and one Encoder-Decoder systems (dubbed as ROVER-1) and finally the ROVER combination of CTC,Encoder Decoder and Hybrid systems (dubbed as ROVER-2).

For the text segmenter, we showed the models trained on two data sizes (small and large), together with the larger models being adapted with domains and adapted with noise. Similarly, we showed the Translation models with additional adaptation towards domain and noisy inputs.

ASR	WER
ROVER-1	21.2%
HYBRID	17.6%
ROVER-2	16.7%

Table 1: Word-Error Rate of different ASR configurations on the tst2014 English set.

ASR	SEG	MT	BLEU
ROVER-1	Small	Transformer	13.77
ROVER-1	Small	D. Adapted	16.26
ROVER-1	Large	D. Adapted	18.2
ROVER-1	D.Adapted	D. Adapted	19.15
ROVER-1	N.Adapted	N.Adapted	19.32
HYBRID	D.Adapted	N.Adapted	21.33
HYBRID	N.Adapted	N.Adapted	21.41
ROVER-2	D.Adapted	N.Adapted	**22.61**
ROVER-2	N.Adapted	N.Adapted	21.35

Table 2: SLT English→German results. We report the BLEU scores (after re-segmentation) on the tst2014 test data. Note: Noise-adapted models (N.Adapted) were already adapted to the TED domain (D.Adapted) previously.

Regarding the whole cascade performance, as can be seen from Table 2, the score dramatically increased with the help of domain adaptation (2.5 BLEU points improved from adapting the translation model, and an additional 3.2 points from having a stronger adapted segmentation model (Large into D.Adaptation). Additional noise-adaptation on the segmentation and translation models improves the result by 0.2. We can also see that the HYBRID ASR model is much better than the ROVER-1 configuration, thanks to 17% improvement in word error rate on the English speech input. As a result, the whole cascade is improved by significant 3 BLEU points. The best ASR configuration - ROVER-2 - finalized the best result at 22.61. Notably, we have to use the segmentation model configuration without noisy-adaptation to achieve this, the counterpart fell short by 1 BLEU point.

6.2. End-to-End

We conduct preliminary experiments on the well-established Fisher Spanish-English Speech Translation Corpus [23] to confirm the model's accuracy. We obtain 35.3 BLEU points on Fisher/Test, 2.8 points better than a cascaded model using a similar architecture and the same training data. We then train the end-to-end model described in Section 5 on TEDLIUM2 for the ASR task, and on IWSLT2018's provided end-to-end TED data for the remaining tasks. We test the resulting model on the tst2013 dataset and obtain 10.3 BLEU points without casing, and 9.3 BLEU points case-sensitive scoring. This is still much worse than the cascaded models described above, despite the promising preliminary results. As potential reasons we identify the rather noisy provided training data, mismatch between the manual segmentation at training time and the automatic segmentation at test time, and the lack of additional data (beyond TED) included in the training.

7. Conclusions

In this evaluated we build a cascaded speech translation model as well as an end-to-end model for the English to German Speech Translation task.

For the cascaded approach, we see that the combination of several ASR systems reaches the best performance. Furthermore, we get the best single performance by using a hybrid model.

For the end-to-end model, we cannot achieve the same performance as the cascaded approach. One challenge is the integration of the significantly larger data available for the cascaded models.

8. Acknowledgements

9. References

[1] T. Zenkel, M. Sperber, J. Niehues, M. Müller, N.-Q. Pham, S. Stüker, and A. Waibel, "Open Source Toolkit for Speech to Text Translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 111, pp. 125–135, October 2018. [Online]. Available: https://ufal.mff.cuni.cz/pbml/111/art-zenkel-et-al.pdf

[2] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *LREC*, 2014.

[3] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 83–86.

[4] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5391–5395.

[5] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proceedings of the 23rd international conference on Machine learning - ICML 2006*. ACM Press, 2006.

[6] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, "Subword and crossword units for ctc acoustic models," *Interspeech*, 2018.

[7] 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)*. Association for Computational Linguistics, 2016.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016.

[9] G. Neubig, M. Sperber, X. Wang, M. Felix, A. Matthews, S. Padmanabhan, Y. Qi, D. S. Sachan, P. Arthur, P. Godard, *et al.*, "Xnmt: The extensible neural machine translation toolkit," Boston, USA, 2018.

[10] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Neural Information Processing Systems Conference (NIPS)*, Barcelona, Spain, 2016.

[11] T. Nguyen and D. Chiang, "Improving lexical choice in neural machine translation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2015.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[14] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997.

[15] E. Cho, J. Niehues, and A. Waibel, "NMT-based segmentation and punctuation insertion for real-time spoken language translation," in *Interspeech 2017*. ISCA, aug 2017.

[16] ——, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, 2012. [Online]. Available: https://isl.anthropomatik.kit.edu/pdf/Cho2012.pdf

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[18] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[19] E. Cho, J. Niehues, T.-L. Ha, M. Sperber, M. Mediani, and A. Waibel, "Adaptation and combination of nmt systems: The kit translation systems for iwslt 2016," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT), Seattle, WA*, 2016.

[20] M. Sperber, J. Niehues, and A. Waibel, "Toward Robust Neural Machine Translation for Noisy Input Sequences," in *International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[21] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-to-End Speech Recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[22] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Interspeech*, 2013.

[23] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus," in *International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013. [Online]. Available: http://cs.jhu.edu/ gkumar/papers/post2013improved.pdf