The MeMAD Submission to the IWSLT 2018 Speech Translation Task

Sulubacak, Umut

2018-10-30

Sulubacak , U , Tiedemann , J , Rouhe , A , Stig-Arne , G & Kurimo , M 2018 , The MeMAD Submission to the IWSLT 2018 Speech Translation Task . in M Turchi , J Niehues & M Frederico (eds) , Proceedings of the 15th International Workshop on Spoken Language Translation (IWSLT 2018) . Bruges , pp. 89-94 , International Workshop on Spoken Language Translation , Brugge , Belgium , 29/10/2018 .

http://hdl.handle.net/10138/263703

The MeMAD Submission to the IWSLT 2018 Speech Translation Task

Umut Sulubacak * *Jörg Tiedemann* * *Aku Rouhe* † *Stig-Arne Grönroos* † *Mikko Kurimo* †

∗ Department of Digital Humanities / HELDIG
University of Helsinki, Finland
{umut.sulubacak | jorg.tiedemann}@helsinki.fi

† Department of Signal Processing and Acoustics
Aalto University, Finland
{aku.rouhe | stig-arne.gronroos | mikko.kurimo}@aalto.fi

Abstract

This paper describes the MeMAD project entry to the IWSLT Speech Translation Shared Task, addressing the translation of English audio into German text. Between the pipeline and end-to-end model tracks, we participated only in the former, with three contrastive systems. We tried also the latter, but were not able to finish our end-to-end model in time.

All of our systems start by transcribing the audio into text through an automatic speech recognition (ASR) model trained on the TED-LIUM English Speech Recognition Corpus (TED-LIUM). Afterwards, we feed the transcripts into English-German text-based neural machine translation (NMT) models. Our systems employ three different translation models trained on separate training sets compiled from the English-German part of the TED Speech Translation Corpus (TED-TRANS) and the OPENSUBTITLES2018 section of the OPUS collection.

In this paper, we also describe the experiments leading up to our final systems. Our experiments indicate that using OPENSUBTITLES2018 in training significantly improves translation performance. We also experimented with various pre- and postprocessing routines for the NMT module, but we did not have much success with these.

Our best-scoring system attains a BLEU score of 16.45 on the test set for this year's task.

1. Introduction

The evident challenge of speech translation is the transfer of implicit semantics between two different modalities. An end-to-end solution to this task must deal with the challenge posed by intermodality simultaneously with that of interlingual transfer. In a traditional pipeline approach, while speech-to-text transcription is abstracted from translation, there is then the additional risk of error transfer between the two stages. The MeMAD project[1] aims at multilingual description and search in audiovisual data. For this reason, multimodal translation is of great interest to the project.

Our pipeline submission to this year's speech translation task incorporates one ASR model and three contrastive NMT models. For the ASR module, we trained a time-delay neural network (TDNN) acoustic model using the Kaldi toolkit [1] on the provided TED-LIUM speech recognition corpus [2]. We used the transformer implementation of MarianNMT [3] to train our NMT models. For these models, we used contrastive splits of data compiled from two different sources: The n-best decoding hypotheses of the TED-TRANS [4] in-domain speech data, and a version of the OPENSUBTITLES2018 [5] out-of-domain text data (SUBS), further "translated" to an ASR-like format (SUBS-ASR) using a sequence-to-sequence NMT model. The primary system in our submission uses the NMT model trained on the whole data including SUBS-ASR, whereas one of the two contrastive systems uses the original SUBS before the conversion to an ASR-like format, and the other omits OPENSUBTITLES2018 altogether.

We provide further details about the ASR module in Section 2. Later, we provide a review of our experiments on the NMT module in Section 3. The first experiment we describe involves a pre-processing step where we convert our out-of-domain training data to an ASR-like format to avoid mismatch between source-side training samples. Afterwards, we report a postprocessing experiment where we retrain our NMT models with lowercased data, and defer case restoration to a subsequent procedure, and another where we translate several ASR hypotheses at once for each source sample, re-rank their output translations by a language model, and then choose the best-scoring translation for that sample. We present our results in Section 4 along with the relevant discussions.

2. Speech Recognition

The first step in our pipeline is automatic speech recognition. The organizers provide a baseline ASR implementation,

[1] https://www.memad.eu/

which consists of a single, end-to-end trained neural network using a Listen, Attend and Spell (LAS) architecture [6]. The baseline uses the XNMT toolkit [7]. However, we were not able to compile the baseline system, so we trained our own conventional, hybrid TDNN-HMM ASR system using the Kaldi toolkit.

2.1. Architecture

Our ASR system uses the standard Kaldi recipe for the TED-LIUM dataset (release 2), although we filter out some data from the training set to comply with the IWSLT restrictions. The recipe trains a TDNN acoustic model using the lattice-free maximum mutual information criterion [8]. The audio transcripts and large amount of out-of-domain text data included with the TED-LIUM dataset are used to train a heavily pruned 4-gram language model for first-pass decoding and less pruned 4-gram model for rescoring.

2.2. Word Error Rates

The LAS architecture has achieved state-of-the-art word error rates (WER) on a task with two orders of magnitude more training data than here [9], but on smaller datasets hybrid TDNN-HMM ASR approaches are still considerably better. Table 1 shows the results of our ASR model contrasted with those reported by XNMT in [7], on the TED-LIUM development and test sets.

Model	Dev WER	Test WER
TDNN + large 4-gram	8.24	8.83
LAS	15.83	16.16

Table 1: Word error rates on the TED-LIUM dataset.

3. Text-Based Translation

The ASR stage of our pipeline effectively converts the task of speech translation to text-based machine translation. For this stage, we build a variety of NMT setups and assess their performances. We experiment variously with the training architecture, different compositions of the training data, and several pre- and postprocessing methods. We present these experiments in detail in the subsections to follow, and then discuss their results in Section 4.

3.1. Data Preparation

We used the development and test sets from 2010's shared task for validation during training, and the test sets from the tasks between 2013 and 2015 for testing performance during development. In all of our NMT models, we preprocessed our data using the punctuation normalization and tokenization utilities from Moses [10], and applied byte-pair encoding [11] through full-cased and lowercased models as relevant, trained on the combined English and German texts in TED-TRANS and SUBS using 37,000 merge operations to create the vocabulary.

We experiment with attentional sequence-to-sequence models using the Nematus architecture [12] with tied embeddings, layer normalization, RNN dropout of 0.2 and source/target dropout of 0.1. Token embeddings have a dimensionality of 512 and the RNN layer units a size of 1024. The RNNs make use of GRUs in both, encoder and decoder. We use validation data and early stopping after five cycles (1,000 updates each) of decreasing cross-entropy scores. During training we apply dynamic mini-batch fitting with a workspace of 3GB. We also enable length normalization.

For the experiments with the transformer architecture we apply the standard setup with six layers in encoder and decoder, eight attention heads and a dynamic mini-batch fit to 8GB of work space. We also add recommended options such as transformer dropout of 0.1, label smoothing of 0.1, a learning rate of 0.0003, a learning-rate warmup with a linearly increasing rate during the first 16,000 steps, a decreasing learning rate starting at 16,000 steps, a gradient clip norm of 5 and exponential smoothing of parameters.

All translations are created with a beam decoder of size 12.

3.1.1. ASR Output for TED Talks

Translation models trained on standard language are not a good fit for a pipeline architecture that needs to handle noisy output from the ASR component discussed previously in Section 2. Therefore, we ran speech recognition on the entire TED-TRANS corpus in order to replace the original, human-produced English transcriptions with ASR output, which has realistic recognition errors.

To generate additional speech recognition errors to the training transcripts, we selected the top-50 decoding hypotheses. We did the same also for the development data to test our approach. We can now sample from those ASR hypotheses to create training data for our translation models that use the output of English ASR as its input. We experimented with various strategies varying from a selection of the top n ASR candidates to different mixtures of hypotheses of different ranks of confidence. Some of these are shown in Table 2. In the end, there was not a lot of variance between the scores resulting from this selection, and we decided to use the top-10 ASR outputs in the remaining experiments to encourage some tolerance for speech recognition errors in the system.

3.1.2. Translating Written English to ASR-English

The training data that includes audio is very limited and much larger resources are available for text-only systems. Especially useful for the translation of TED talks is the collection of movie subtitles in OPENSUBTITLES2018. For English-German, there is a huge amount of movie subtitles (roughly 22 million aligned sentences with over 170 million

Training data	Model	BLEU
TED-ASR-TOP-1	AMUN	16.65
TED-ASR-TOP-10	AMUN	16.28
TED-ASR-TOP-50	AMUN	15.88
TED-ASR-TOP-1	TRANSFORMER	18.25
TED-ASR-TOP-10	TRANSFORMER	17.90
TED-ASR-TOP-50	TRANSFORMER	18.14

Table 2: Translating the development test set with different models and different selections of ASR output and German translations from the parallel TED-TRANS training corpus.

tokens per language) that can be used to boost the performance of the NMT module.

The problem is, of course, that the subtitles come in regular language, and, again, we would see a mismatch between the training data and the ASR output in the speech translation pipeline. In contrast to approaches that try to normalize ASR output to reflect standard text-based MT input such as [13], we had the idea to transform regular English into ASR-like English using a translation model trained on a parallel corpus of regular TED talk transcriptions and the ASR output generated for the TED talks that we described in the previous section. We ran a number of experiments to test the performance of such a model. Some of the results are listed in Table 3.

Training data	Model	BLEU
TED-ASR-TOP-10	AMUN	61.87
TED-ASR-TOP-10	TRANSFORMER	61.91
TED-ASR-TOP-50	AMUN	61.82

Table 3: Translating English into ASR-like English using a model trained on TED-TRANS and tested on the development test set with original ASR output as reference.

As expected, the BLEU scores are rather high as the target language is the same as the source language, and we only mutate certain parts of the incoming sentences. The results show that there is not such a dramatic difference between the different setups (with respect to the model architecture and the data selection) and that a plain attentional sequence-to-sequence model with recurrent layers (AMUN) performs as well as a transformer model (TRANSFORMER) in this case. This makes sense, as we do not expect many complex long-distance dependencies that influence translation quality in this task. Therefore, we opted for the AMUN model trained on the top-10 ASR outputs, which we can decode efficiently in a distributed way on the CPU nodes of our computer cluster. With this we managed to successfully translate 99% of the entire SUBS collection from standard English into ASR-English. We refer to this set as SUBS-ASR.

We did a manual inspection on the result as well to see what the system actually learns to do. Most of the transformations are quite straightforward. The model learns to lowercase and to remove punctuation as our ASR output does not include it. However, it also does some other modifications that are more interesting from the viewpoint of an ASR module. While we do not have systematic evidence, Table 4 shows a few selected examples that show interesting patterns. First of all, it learns to spell out numbers (see *"2006"* in the first example). This is done consistently and quite accurately from what we have seen. Secondly, it replaces certain tokens with variants that resemble possible confusions that could come from a speech recognition system. The replacement of *"E.U."* with *"you"* and *"Stasi"* with *"stars he"* in these examples are quite plausible and rather surprising for a model that is trained on textual examples only. However, to conclude that the model learns some kind of implicit acoustic model would be a bit far-fetched, even though we would like to investigate the capacity of such an approach further in the future.

Original	**Because in the summer of 2006, the E.U. Commission tabled a directive.**
ASR-REF	because in the summer of two thousand and six the e u commission tabled directive
ASR-OUT	because in the summer of two thousand and six you commission tabled a directive

Original	**Stasi was the secret police in East Germany.**
ASR-REF	what is the secret police in east germany
ASR-OUT	stars he was the secret police in east germany

Table 4: Examples from the translations to ASR-like English. In the first column, ASR-REF refers to the top decoding hypothesis from the ASR model, while ASR-OUT is the output of the model translating the output to an ASR-like format.

In Section 4, we report on the effect of using synthetic ASR-like data on the translation pipeline.

3.2. Recasing Experiments

Our first attempt at a post-processing experiment involved using case-insensitive translation models, and deferring case restoration to a separate process unconditioned by the source side that we would apply after translation. We used the Moses toolkit [10] to train a recaser model on TED-TRANS. Afterwards, we re-trained a translation model on TED-ASR-TOP-10 and SUBS-ASR after lowercasing the training and validation sets, re-translated the development test set with this model, and then used the recaser to restore cases in the lowercase translations that we obtained. As shown in Table 5, evaluating the translations produced through these additional steps yielded scores that were very similar to those obtained by the original case-sensitive translation models, and the result of this experiment was inconclusive.

Training data	BLEU	BLEU-LC
TED-ASR-TOP-10+SUBS-ASR	19.79	20.43
TED-ASR-TOP-10+SUBS-ASR-LC	19.73	20.91

Table 5: Case-sensitive models (TRANSFORMER) versus lowercased models with subsequent recasing. Recasing causes a larger drop than the model gains from training on lowercased training data. BLEU-LC refers to case-insensitive BLEU scores.

3.3. Reranking Experiments

In addition to using different subsets of the n-best lists output by the ASR model as additional training samples for the translation module, we also tried reranking alternatives using KenLM [14]. We initially generated a tokenized and lowercased version of TED-TRANS with all punctuation stripped, and then trained a language model on this set. We used this model to score and rerank samples in the 50-best lists, and then generated a new top-10 subset from this reranked version. However, when we re-trained translation models from these alternative sets, we observed that the model trained on the top-10 subsets before reranking exhibited a significantly better translation performance. We suspect that this is because, while the language model is useful for assessing the surface similarity of the ASR outputs to the source-side references, it was not uncommon for it to assign higher scores to ASR outputs that are semantically inconsistent with the target-side references, causing the NMT module to produce erroneous translations.

Similarly, we experimented with another language model trained on the target side of TED-TRANS, without the preprocessing. We intended this model to score and rerank outputs of the translation models, rather than the ASR module. To measure the effect of this language model, we fed the audio of our internal test set split through the ASR module, and produced 50-best lists for each sample. Afterwards, we used the language model to score and rerank the alternative transcripts for each sample produced by translating this set, and then selected the highest-scoring output for each sample. As in the previous language model experiment, employing this additional procedure significantly crippled the performance of our translation models.

4. Results

The results on development data reveal expected tendencies that we report below. First of all, as consistent with a lot of related literature, we can see a boost in performance when switching from a recurrent network model to the transformer model with multiple self-attention mechanisms. Table 6 shows a clear pattern of the superior performance of the transformer model that is also visible in additional runs that we do not list here. Secondly, we can see the importance of additional training data even if they come from slightly different domains. The vast amount of movie subtitles in OPENSUBTITLES2018 boosts the performance by about 3 absolute BLEU points. Note that the scores in Table 6 refer to models that do not use subtitles transformed into ASR-like English (SUBS-ASR) and which are not fine-tuned to TED talk translations.

Training data	Model	BLEU
TED-ASR-TOP-10	AMUN	16.28
TED-ASR-TOP-10+SUBS	AMUN	19.93
TED-ASR-TOP-10	TRANSFORMER	17.90
TED-ASR-TOP-10+SUBS	TRANSFORMER	20.44

Table 6: Model performance on the development test set when adding movie subtitles to the training data.

The effect of pre-processing by producing ASR-like English in the subtitle corpus is surprisingly negative. If we look at the scores in Table 7, we can see that the performance actually drops in all cases when considering only the untuned systems. We did not really expect that with the rather positive impression that we got from the manual inspection of the English-to-ASR translation discussed earlier. However, it is interesting to see the effect of fine-tuning. Fine-tuning here refers to a second training procedure that continues training with pure in-domain data (TED talks) after training the general model on the entire data set until convergence on validation data. Table 7 shows an interesting effect that may explain the difficulties of the integration of the synthetic ASR data. The fine-tuned model actually outperforms the model trained on standard data, which is due to a substantial jump from untuned models to the tuned version. The difference between those models with standard data is, on the other hand, only minor.

	BLEU	
Training data	Untuned	Tuned
TED-ASR-TOP-10+SUBS	20.44	20.58
TED-ASR-TOP-10+SUBS-ASR	19.79	20.80

Table 7: Training with original movie subtitles versus subtitles with English transformed into ASR-like English, before and after fine-tuning on TED-ASR-TOP-10 as pure in-domain training data (TRANSFORMER).

The synthetic ASR data look more similar to the TED-ASR data and, therefore, the model might get more confused between in-domain and out-of-domain data than it does for the model trained on the original subtitle data in connection with TED-ASR. Fine-tuning to TED-ASR brings the model back on track again and synthetic ASR data becomes modestly beneficial.

Also of note is the contrast between the evaluation scores we obtained in development and those from the official test set. The translations we submitted obtain the BLEU scores shown in Table 8 on this year's test set.

Training data	BLEU
TED-ASR-Top-10	14.34
TED-ASR-Top-10+Subs	16.45
TED-ASR-Top-10+Subs-ASR	15.80

Table 8: BLEU scores from our final models (Transformer)—respectively, the 2nd contrastive, 1st contrastive, and primary submission—on this year's test set. The scores from the two models with Subs in their training data were obtained after fine-tuning on TED-ASR-Top-10.

5. Conclusions

Apart from employing well-established practices such as normalization and byte-pair encoding as well as the benefits of using the transformer architecture, the only substantial boost to translation performance came from our data selection for the NMT module. The NMT module of our best-performing system on this year's test set was trained on TED-ASR-Top-10 and the raw Subs, and later fine-tuned on TED-ASR-Top-10.

Although we ran many experiments to improve various steps of our speech translation pipeline, their influence on translation performance has been marginal at best. The effects of training with different TED-ASR subsets were hard to distinguish. While using Subs-ASR in training seemed to provide a modest improvement in development, this effect was not carried over to the final results on the test set. The later experiments with lowercasing and recasing had an ambiguous effect, and those with reranking had a noticeably negative outcome.

In future work, our aim is to further investigate what factors in a good speech translation model, and continue experimenting in relation to these on the NMT module. We will also try to improve our TDNN-HMM ASR module by replacing the n-grams with an RNNLM, and try see how our complete end-to-end speech-to-text translation model performs after having sufficient training time.

6. Acknowledgements

This work has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069, and by the Academy of Finland in the project 313988. In addition the Finnish IT Center for Science (CSC) provided computational resources.

7. References

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[2] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an automatic speech recognition dedicated corpus," in *LREC*, 2012, pp. 125–129.

[3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121. [Online]. Available: http://www.aclweb.org/anthology/P18-4020

[4] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[5] J. Tiedemann, "News from OPUS - A collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009, vol. V, pp. 237–248.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[7] 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," *arXiv preprint arXiv:1803.00188*, 2018.

[8] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[9] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, *et al.*, "State-of-the-art speech recognition

with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[10] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL16*, 2015.

[12] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. M. Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," *CoRR*, vol. abs/1703.04357, 2017. [Online]. Available: http://arxiv.org/abs/1703.04357

[13] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov 2006, pp. 158–165.

[14] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2011, pp. 187–197.