ELITR Non-Native Speech Translation at IWSLT 2020

Dominik Macháček† and **Jonáš Kratochvíl**† and **Sangeet Sagar**† and
Matúš Žilinec† and **Ondřej Bojar**† and **Thai-Son Nguyen**‡ and **Felix Schneider**‡ and
Philip Williams* and **Yuekun Yao***
†Charles University, ‡Karlsruhe Institute of Technology, *University of Edinburgh
†{surname}@ufal.mff.cuni.cz, except jkratochvil@ufal.mff.cuni.cz,
‡{firstname.lastname}@kit.edu,
*pwillia4@inf.ed.ac.uk,yyao2@exseed.ed.ac.uk

Abstract

This paper is an ELITR system submission for the non-native speech translation task at IWSLT 2020. We describe systems for offline ASR, real-time ASR, and our cascaded approach to offline SLT and real-time SLT. We select our primary candidates from a pool of pre-existing systems, develop a new end-to-end general ASR system, and a hybrid ASR trained on non-native speech. The provided small validation set prevents us from carrying out a complex validation, but we submit all the unselected candidates for contrastive evaluation on the test set.

1 Introduction

This paper describes the submission of the EU project ELITR (European Live Translator)[1] to the non-native speech translation task at IWSLT 2020 (Ansari et al., 2020). It is a result of a collaboration of project partners Charles University (CUNI), Karlsruhe Institute of Technology (KIT), and University of Edinburgh (UEDIN), relying on the infrastructure provided to the project by PerVoice company.

The non-native speech translation shared task at IWSLT 2020 complements other IWSLT tasks by new challenges. Source speech is non-native English. It is spontaneous, sometimes disfluent, and some of the recordings come from a particularly noisy environment. The speakers often have a significant non-native accent. In-domain training data are not available. They consist only of native out-domain speech and non-spoken parallel corpora. The validation data are limited to 6 manually transcribed documents, from which only 4 have reference translations. The target languages are Czech and German.

The task objectives are quality and simultaneity, unlike the previous tasks, which focused only on the quality. Despite the complexity, the resulting systems can be potentially appreciated by many users attending an event in a language they do not speak or having difficulties understanding due to unfamiliar non-native accents or unusual vocabulary.

We build on our experience from the past IWSLT and WMT tasks, see e.g. Pham et al. (2019); Nguyen et al. (2017); Pham et al. (2017); Wetesko et al. (2019); Bawden et al. (2019); Popel et al. (2019). Each of the participating institutions has offered independent ASR and MT systems trained for various purposes and previous shared tasks. We also create some new systems for this task and deployment for the purposes of the ELITR project. Our short-term motivation for this work is to connect the existing systems into a working cascade for SLT and evaluate it empirically, end-to-end. In the long-term, we want to advance state of the art in non-native speech translation.

2 Overview of Our Submissions

This paper is a joint report for two primary submissions, for online and offline sub-track of the non-native simultaneous speech translation task.

First, we collected all ASR systems that were available for us (Section 3.1) and evaluated them on the validation set (Section 3.2). We selected the best candidate for offline ASR to serve as the source for offline SLT. Then, from the ASR systems, which are usable in online mode, we selected the best candidate for online ASR and as a source for online SLT.

In the next step (Section 4), we punctuated and truecased the online ASR outputs of the validation set, segmented them to individual sentences, and translated them by all the MT systems we had available (Section 5.1). We integrated the online ASRs and MTs into our platform for online SLT

[1] http://elitr.eu

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 200–208

https://doi.org/10.18653/v1/P17

(Sections 5.2 and 5.3). We compared them using automatic MT quality measures and by simple human decision, to compensate for the very limited and thus unreliable validation set (Section 5.4). We selected the best candidate systems for each target language, for Czech and German.

Both best candidate MT systems are very fast (see Section 5.5). Therefore, we use them both for the online SLT, where the low translation time is critical, and for offline SLT.

In addition to the primary submissions, we included all the other candidate systems and some public services as contrastive submissions.

3 Automatic Speech Recognition

This section describes our automatic speech recognition systems and their selection.

3.1 ASR Systems

We use three groups of ASR systems. They are described in the following sections.

3.1.1 KIT ASR

KIT has provided three hybrid HMM/ANN ASR systems and an end-to-end sequence-to-sequence ASR system.

The hybrid systems, called KIT-h-large-lm1, KIT-h-large-lm2 and KIT-hybrid, were developed to run on the online low-latency condition, and differ in the use of the language models.

The KIT-h-large-lm adopted a 4-gram language model which was trained on a large text corpus (Nguyen et al., 2017), while the KIT-hybrid employed only the manual transcripts of the speech training data. We would refer the readers to the system paper by Nguyen et al. (2017) for more information on the training data and the studies by Nguyen et al. (2020); Niehues et al. (2018) for more information about the online setup.

The end-to-end ASR, so-called KIT-seq2seq, followed the architecture and the optimizations described by Nguyen et al. (2019). It was trained on a large speech corpus, which is the combination of Switchboard, Fisher, LibriSpeech, TED-LIUM, and Mozilla Common Voice datasets. It was used solely without an external language model.

All KIT ASR systems are unconstrained because they use more training data than allowed for the task.

3.1.2 Kaldi ASR Systems

We used three systems trained in the Kaldi ASR toolkit (Povey et al., 2011). These systems were trained on Mozilla Common Voice, TED-LIUM, and AMI datasets together with additional textual data for language modeling.

Kaldi-Mozilla For Kaldi-Mozilla, we used the Mozilla Common Voice baseline Kaldi recipe.[2] The training data consist of 260 hours of audio. The number of unique words in the lexicon is 7996, and the number of sentences used for the baseline language model is 6994, i.e., the corpus is very repetitive. We first train the GMM-HMM part of the model, where the final number of hidden states for the HMM is 2500, and the number of GMM components is 15000. We then train the chain model, which uses the Time delay neural network (TDNN) architecture (Peddinti et al., 2015) together with the Batch normalization regularization and ReLU activation. We use MFCC features to represent audio frames, and we concatenate them with the 100-dimensional I-vector features for the neural network training. We recompile the final chain model with CMU lexicon to increase the model capacity to 127384 words and 4-gram language model trained with SRILM (Stolcke, 2002) on 18M sentences taken from English news articles.

Kaldi-TedLium serves as another baseline, trained on 130 hours of TED-LIUM data (Rousseau et al., 2012) collected before the year 2012. The Kaldi-TedLium model was developed by the University of Edinburgh and was fully described by Klejch et al. (2019). This model was primarily developed for discriminative acoustic adaptation to domains distinct from the original training domain. It is achieved by reusing the decoded lattices from the first decoding pass and by finetuning for TED-LIUM development and test set. The setup follows the Kaldi 1f TED-LIUM recipe. The architecture is similar to Kaldi-Mozilla and uses a combination of TDNN layers with batch normalization and ReLU activation. The input features are MFCC and I-vectors.

Kaldi-AMI was trained on the 100 hours of the AMI data, which comprise of staged meeting recordings (Mccowan et al., 2005). These data

[2] https://github.com/kaldi-asr/kaldi/tree/master/egs/commonvoice/s5

domain document	AMI AMIa	 AMIb	 AMIc	 AMId	Antrecorp Teddy	 Autocentrum	Auditing Auditing
KIT-h-large-lm1	50.71	47.96	53.11	50.43	65.92	19.25	18.54
KIT-h-large-lm2	47.82	41.71	42.10	45.77	75.87	28.59	19.81
KIT-hybrid	40.72	38.45	41.09	43.28	58.99	21.04	21.44
KIT-seq2seq	33.73	28.54	34.45	42.24	42.57	9.91	10.45
Kaldi-TedLium	42.44	38.56	41.83	44.36	61.12	18.68	22.81
Kaldi-Mozilla	52.89	56.37	58.50	58.90	68.72	45.41	34.36
Kaldi-AMI	~~28.01~~	~~23.04~~	~~26.87~~	~~29.34~~	59.66	20.62	28.39
Microsoft	53.72	52.62	56.67	58.58	87.82	39.64	24.22
Google	51.52	49.47	53.11	56.88	61.01	14.12	17.47

Table 1: WER rates of individual documents in the development set. Kaldi-AMI scores on AMI domain are striked through because they are unreliable due to an overlap with the training data.

domain	document	sents.	tokens	duration	references
Antrecorp	Teddy	11	171	1:15	2
Antrecorp	Autocentrum	12	174	1:06	2
Auditing	Auditing	25	528	5:38	1
AMI	AMIa	220	1788	15:09	1
AMI	AMIb	614	4868	35:17	0
AMI	AMIc	401	3454	24:06	0
AMI	AMId	281	1614	13:01	0

Table 2: The size of the development set `iwslt2020-nonnative-minidevset-v2`. The duration is in minutes and seconds. As "references" we mean the number of independent referential translations into Czech and German.

	WER weighted average AMI	 Antrecorp	 Auditing	avg domain
KIT-seq2seq[1]	**32.96**	**26.10**	**10.45**	**23.17**
Kaldi-TedLium	40.91	39.72	22.81	34.48
Kaldi-Mozilla	56.82	56.96	34.36	49.38
Kaldi-AMI	~~25.79~~	39.97	28.39	31.38
Microsoft	54.80	63.52	24.22	47.51
Google	51.88	37.36	17.47	35.57
KIT-h-large-lm1	50.24	42.38	**18.54**[2]	37.05
KIT-h-large-lm2	43.32	52.02	19.81	38.38
KIT-hybrid	**40.24**[1]	**39.85**[1]	21.44	**33.84**

Table 3: Weighted average WER for the domains in validation set, and their average. The top line-separated group are offline ASR systems, the bottom are online. Bold numbers are the lowest considerable WER in the group. Kaldi-AMI score on AMI is not considered due to overlap with training data. Bold names are the primary (marked with [1]) and secondary (marked with [2]) candidates.

were recorded mostly by non-native English speakers with a different microphone and acoustic environment conditions. The model setup used follows the Kaldi 1i ami recipe. Kaldi-AMI cannot be reliably assessed on the AMI part of the development due to the overlap of training and development data. We have decided not to exclude this overlap so that we do not limit the amount of available training data for our model.

3.1.3 Public ASR Services

As part of our baseline models, we have used Google Cloud Speech-to-Text API[3] and Microsoft Azure Speech to Text.[4] Both of these services provide an API for transcription of audio files in WAV format, and they use neural network acoustic models. We kept the default settings of these systems.

The Google Cloud system supports over 100 languages and several types of English dialects (such as Canada, Ireland, Ghana, or the United Kingdom). For decoding of the development and test set, we have used the United Kingdom English dialect option. The system can be run either in real-time or offline mode. We have used the offline option for this experiment.

The Microsoft Azure Bing Speech API supports fewer languages than Google Cloud ASR but adds more customization options of the final model. It can be also run both in real-time or offline mode. For the evaluation, we have used the offline mode and the United Kingdom English (en-GB) dialect.

3.2 Selection of ASR Candidates

We processed the validation set with all the ASR systems, evaluated WER, and summarized them in Table 1. The validation set (Table 2) contains three different domains with various document sizes, and the distribution does not fully correspond to the test set. The AMI domain is not present in the test set at all, but it is a part of Kaldi-AMI training data. Therefore, a simple selection by an average WER on the whole validation set could favor the systems which perform well on the AMI domain, but they

[3] https://cloud.google.com/speech-to-text

[4] https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

could not be good candidates for the other domains.

In Table 3, we present the weighted average of WER in the validation domains. We weight it by the number of gold transcription words in each of the documents. We observe that Kaldi-AMI has a good performance on the AMI domain, but it is worse on the others. We assume it is overfitted for this domain, and therefore we do not use it as the primary system.

For offline ASR, we use KIT-seq2seq as the primary system because it showed the lowest error rate on the averaged domain.

The online ASR systems can exhibit somewhat lower performance than offline systems. We select KIT-h-large-lm1 as the primary online ASR candidate for Auditing, and KIT-hybrid as primary for the other domains.

Our second primary offline ASR is Kaldi-AMI.

4 Punctuation and Segmentation

All our ASR systems output unpunctuated, often all lowercased text. The MT systems are designed mostly for individual sentences with proper casing and punctuation. To overcome this, we first insert punctuation and casing to the ASR output. Then, we split it into individual sentences by the punctuation marks by a rule-based language-dependent Moses sentence splitter (Koehn et al., 2007).

Depending on the ASR system, we use one of two possible punctuators. Both of them are usable in online mode.

4.1 KIT Punctuator

The KIT ASR systems use an NMT-based model to insert punctuation and capitalization in an otherwise unsegmented lowercase input stream (Cho et al., 2012, 2015). The system is a monolingual translation system that translates from raw ASR output to well-formed text by converting words to upper case, inserting punctuation marks, and dropping words that belong to disfluency phenomena. It does not use the typical sequence-to-sequence approach of machine translation. However, it considers a sliding window of recent (uncased) words and classifying each one according to the punctuation that should be inserted and whether the word should be dropped for being a part of disfluency. This gives the system a constant input and output size, removing the need for a sequence-to-sequence model.

While inserting punctuation is strictly necessary for MT to function at all, inserting capitalization and removing disfluencies improves MT performance by making the test case more similar to the MT training conditions (Cho et al., 2017).

4.2 BiRNN Punctuator

For other systems, we use a bidirectional recurrent neural network with an attention-based mechanism by Tilk and Alumäe (2016) to restore punctuation in the raw stream of ASR output. The model was trained on 4M English sentences from CzEng 1.6 (Bojar et al., 2016) data and a vocabulary of 100K most frequently occurring words. We use CzEng because it is a mixture of domains, both originally spoken, which is close to the target domain, and written, which has richer vocabulary, and both original English texts and translations, which we also expect in the target domain. The punctuated transcript is then capitalized using an English tri-gram truecaser by Lita et al. (2003). The truecaser was trained on 2M English sentences from CzEng.

5 Machine Translation

This section describes the translation part of SLT.

5.1 MT Systems

See Table 4 for the summary of the MT systems. All except de-LSTM are Transformer-based neural models using Marian (Junczys-Dowmunt et al., 2018) or Tensor2Tensor (Vaswani et al., 2018) back-end. All of them, except de-T2T, are unconstrained because they are trained not only on the data sets allowed in the task description, but all the used data are publicly available.

5.1.1 WMT Models

WMT19 Marian and WMT18 T2T models are Marian and T2T single-sentence models from Popel et al. (2019) and Popel (2018). WMT18 T2T was originally trained for the English-Czech WMT18 news translation task, and reused in WMT19. WMT19 Marian is its reimplementation in Marian for WMT19. The T2T model has a slightly higher quality on the news text domain than the Marian model. The Marian model translates faster, as we show in Section 5.5.

5.1.2 IWSLT19 Model

The IWSLT19 system is an ensemble of two English-to-Czech Transformer Big models trained using the Marian toolkit. The models were originally trained on WMT19 data and then finetuned

system	back-end	source-target	constrained	reference
WMT19 Marian	Marian	en→cs	no	Popel et al. (2019), Section 5.1.1
WMT18 T2T	T2T	en→cs	no	Popel et al. (2019), Section 5.1.1
IWSLT19	Marian	en→cs	no	Wetesko et al. (2019), Section 5.1.2
OPUS-A	Marian	en↔{cs,de+5 l.}	no	Section 5.1.3
OPUS-B	Marian	en↔{cs,de+39 l.}	no	Section 5.1.3
T2T-multi	T2T	en↔{cs,de,en+39 l.}	no	Section 5.1.4
T2T-multi-big	T2T	en↔{cs,de,en+39 l.}	no	Section 5.1.4
de-LSTM	NMTGMinor	en→de	no	Dessloch et al. (2018), Section 5.1.6
de-T2T	T2T	en→de	yes	Section 5.1.5

Table 4: The summary of our MT systems.

on MuST-C TED data. The ensemble was a component of Edinburgh and Samsung's submission to the IWSLT19 Text Translation task. See Section 4 of Wetesko et al. (2019) for further details of the system.

5.1.3 OPUS Multi-Lingual Models

The OPUS multilingual systems are one-to-many systems developed within the ELITR project. Both were trained on data randomly sampled from the OPUS collection (Tiedemann, 2012), although they use distinct datasets. OPUS-A is a Transformer Base model trained on 1M sentence pairs each for 7 European target languages: Czech, Dutch, French, German, Hungarian, Polish, and Romanian. OPUS-B is a Transformer Big model trained on a total of 231M sentence pairs covering 41 target languages that are of particular interest to the project[5] After initial training, OPUS-B was finetuned on an augmented version of the dataset that includes partial sentence pairs, artificially generated by truncating the original sentence pairs (similar to Niehues et al., 2018). We produce up to 10 truncated sentence pairs for every one original pair.

5.1.4 T2T Multi-Lingual Models

T2T-multi and T2T-multi-big are respectively Transformer and Transformer Big models trained on a Cloud TPU based on the default T2T hyper-parameters, with the addition of target language tokens as in Johnson et al. (2017). The models were trained with a shared vocabulary on a dataset of English-to-many and many-to-English sentence pairs from OPUS-B containing 42 languages in total, making them suitable for pivoting. The models do not use finetuning.

5.1.5 de-T2T

de-T2T translation model is based on a Tensor2Tensor translation model model using training hyper-parameters similar to Popel and Bojar (2018). The model is trained using all the parallel corpora provided for the English-German WMT19 News Translation Task, without back-translation. We use the last training checkpoint during model inference. To reduce the decoding time, we apply greedy decoding instead of a beam search.

5.1.6 KIT Model

KIT's translation model is based on an LSTM encoder-decoder framework with attention (Pham et al., 2017). As it is developed for our lecture translation framework (Müller et al., 2016), it is finetuned for lecture content. In order to optimize for a low-latency translation task, the model is also trained on partial sentences in order to provide more stable translations (Niehues et al., 2016).

5.2 ELITR SLT Platform

We use a server called Mediator for the integration of independent ASR and MT systems into a cascade for online SLT. It is a part of the ELITR platform for simultaneous multilingual speech translation (Franceschini et al., 2020). The workers, which can generally be any audio-to-text or text-to-text processors, such as ASR and MT systems, run inside of their specific software and hardware environments located physically in their home labs around Europe. They connect to Mediator and offer a service. A client, often located in another lab, requests Mediator for a cascade of services, and Mediator connects them. This platform simplifies the cross-institutional collaboration when one institution offers ASR, the other MT, and the third tests them as a client. The platform enables using the SLT pipeline easily in real-time.

[5] The 41 target languages include all EU languages (other than English) and 18 languages that are official languages of EUROSAI member countries. Specifically, these are Albanian, Arabic, Armenian, Azerbaijani, Belorussian, Bosnian, Georgian, Hebrew, Icelandic, Kazakh, Luxembourgish, Macedonian, Montenegrin, Norwegian, Russian, Serbian, Turkish, and Ukrainian.

5.3 MT Wrapper

The simultaneous ASR incrementally produces the recognition hypotheses and gradually improves them. The machine translation system translates one batch of segments from the ASR output at a time. If the translation is not instant, then some ASR hypotheses may be outdated during the translation and can be skipped. We use a program called MT Wrapper for connecting the output of self-updating ASR with non-instant NMT systems.

MT Wrapper has two threads. The receiving thread segments the input for our MTs into individual sentences, saves the input into a buffer, and continuously updates it. The translating thread is a loop that retrieves the new content from the buffer. If a segment has been translated earlier in the current process, it is outputted immediately. Otherwise, the new segments are sent in one batch to the NMT system, stored to a cache and outputted.

For reproducibility, the translation cache is empty at the beginning of a process, but in theory it could be populated by a translation memory. The cache significantly reduces the latency because the punctuator often oscillates between two variants of casing or punctuation marks within a short time.

MT Wrapper has a parameter to control the stability and latency. It can mask the last k words of incomplete sentences from the ASR output, as in Ma et al. (2019) and Arivazhagan et al. (2019), considering only the currently completed sentences, or only the "stable" sentences, which are beyond the ASR and punctuator processing window and never change. We do not tune these parameters in the validation. We do not mask any words or segments in our primary submission, but we submit multiple non-primary systems differing in these parameters.

5.4 Quality Validation

For comparing the MT candidates for SLT, we processed the validation set by three online ASR systems, translated them by the candidates, aligned them with reference by mwerSegmenter (Matusov et al., 2005) and evaluated the BLEU score (Post, 2018; Papineni et al., 2002) of the individual documents. However, we were aware that the size of the validation set is extremely limited (see Table 2) and that the automatic metrics as the BLEU score estimate the human judgment of the MT quality reliably only if there is a sufficient number of sentences or references. It is not the case of this validation set.

Therefore, we examined them by a simple comparison with source and reference. We realized that the high BLEU score in the Autocentrum document is induced by the fact that one of the translated sentences matches exactly matches a reference because it is a single word "thanks". This sentence increases the average score of the whole document, although the rest is unusable due to mistranslated words. The ASR quality of the two Antrecorp documents is very low, and the documents are short. Therefore we decided to omit them in comparison of the MT candidates.

We examined the differences between the candidate translations on the Auditing document, and we have not seen significant differences, because this document is very short. The AMIa document is longer, but it contains long pauses and many isolated single-word sentences, which are challenging for ASR. The part with a coherent speech is very short.

Finally, we selected the MT candidate, which showed the highest average BLEU score on the three KIT online ASR systems both on Auditing and AMIa document because we believe that averaging the three ASR sources shows robustness against ASR imperfections. See Table 5 and Table 6 for the BLEU scores on Czech and German. The selected candidates are IWSLT19 for Czech and OPUS-B for German. However, we also submit all other candidates as non-primary systems to test them on a significantly larger test set. We use these candidates both for online and offline SLT.

5.5 Translation Time

We measured the average time, in which the MT systems process a batch of segments of the validation set (Table 7). If the ASR updates are distributed uniformly in time, than the average batch translation time is also the expected delay of machine translation. The shortest delay is almost zero; in cases when the translation is cached or for very short segments. The longest delay happens when an ASR update arrives while the machine is busy with processing the previous batch. The delay is time for translating two subsequent batches, waiting and translating.

We suppose that the translation time of our primary candidates is sufficient for real-time translation, as we verified in on online SLT test sessions.

We observe differences between the MT systems.

MT	document	gold	KIT-hybrid	KIT-h-large-lm1	KIT-h-large-lm2	avg KIT
OPUS-B	Teddy	42.8463	2.418	2.697	1.360	2.158
IWSLT19	Teddy	51.397	1.379	2.451	1.679	1.836
WMT19 Marian	Teddy	49.328	1.831	1.271	1.649	1.584
WMT18 T2T	Teddy	54.778	1.881	1.197	1.051	1.376
OPUS-A	Teddy	25.197	1.394	1.117	1.070	1.194
T2T-multi	Teddy	36.759	1.775	0.876	0.561	1.071
WMT18 T2T	Autocentrum	42.520	12.134	13.220	14.249	13.201
WMT19 Marian	Autocentrum	39.885	10.899	10.695	12.475	11.356
OPUS-B	Autocentrum	29.690	12.050	10.873	9.818	10.914
IWSLT19	Autocentrum	37.217	9.901	8.996	8.900	9.266
OPUS-A	Autocentrum	30.552	9.201	9.277	8.483	8.987
T2T-multi	Autocentrum	20.011	6.221	2.701	3.812	4.245
IWSLT19	AMIa	**22.878**	5.377	2.531	3.480	**3.796**
WMT18 T2T	AMIa	21.091	5.487	2.286	3.411	3.728
WMT19 Marian	AMIa	22.036	4.646	2.780	3.739	3.722
OPUS-B	AMIa	19.224	4.382	3.424	2.672	3.493
OPUS-A	AMIa	15.432	3.131	2.431	2.500	2.687
T2T-multi	AMIa	13.340	2.546	2.061	1.847	2.151
IWSLT19	Auditing	**9.231**	1.096	3.861	2.656	**2.538**
OPUS-B	Auditing	6.449	1.282	3.607	2.274	2.388
OPUS-A	Auditing	8.032	1.930	4.079	0.900	2.303
WMT19 Marian	Auditing	8.537	1.087	3.571	1.417	2.025
WMT18 T2T	Auditing	9.033	1.201	2.935	1.576	1.904
T2T-multi	Auditing	3.923	1.039	1.318	1.110	1.156

Table 5: Validation BLEU scores in percents (range 0-100) for SLT into Czech from ASR sources. The column “gold” is translation from the gold transript. It shows the differences between MT systems, but was not used in validation.

The size and the model type of WMT19 Marian and WMT18 T2T are the same (see Popel et al., 2019), but they differ in implementation.

WMT19 Marian is slightly faster than IWSLT19 model because the latter is an ensemble of two models. OPUS-B is slower than OPUS-A because the former is bigger. Both are slower than WMT19 Marian due to multi-targeting and different preprocessing. WMT19 Marian uses embedded SentencePiece (Kudo and Richardson, 2018), while the multi-target models use an external Python process for BPE (Sennrich et al., 2016). The timing may be affected also by different hardware.

At the validation time, T2T-multi and T2T-multi-big used suboptimal setup.

6 Conclusion

We presented ELITR submission for non-native SLT at IWSLT 2020. We observe a significant qualitative difference between the end-to-end offline ASR methods and hybrid online methods. The component that constrains the offline SLT from real-time processing is the ASR, not the MT.

We selected the best candidates from a pool of pre-existing and newly developed components, and submitted our primary submissions, although the size of the development set limits us from a reliable validation. Therefore, we submitted all our unselected candidates for contrastive evaluation on the test set. For the results, we refer to Ansari et al. (2020).

Acknowledgments

The research was partially supported by the grants 19-26934X (NEUREM3) of the Czech Science Foundation, H2020-ICT-2018-2-825460 (ELITR) of the EU, 398120 of the Grant Agency of Charles University, and by SVV project number 260 575.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

Naveen Arivazhagan, Colin Cherry, Isabelle Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2019. Re-translation strategies for long form, simultaneous, spoken language translation. *ArXiv*, abs/1912.03393.

MT	document	gold	KIT-hybrid	KIT-h-large-lm1	KIT-h-large-lm2	avg KIT
de-T2T	Teddy	45.578	2.847	3.181	1.411	2.480
OPUS-A	Teddy	29.868	1.873	1.664	1.139	1.559
de-LSTM	Teddy	3.133	2.368	2.089	1.254	1.904
OPUS-B	Teddy	41.547	2.352	1.878	1.454	1.895
T2T-multi	Teddy	31.939	1.792	3.497	1.661	2.317
de-T2T	Autocentrum	36.564	9.031	6.229	3.167	6.142
OPUS-A	Autocentrum	26.647	8.898	13.004	2.324	8.075
de-LSTM	Autocentrum	19.573	10.395	13.026	2.322	8.581
OPUS-B	Autocentrum	28.841	10.153	12.134	9.060	10.449
T2T-multi	Autocentrum	22.631	8.327	8.708	6.651	7.895
de-T2T	AMIa	34.958	8.048	5.654	7.467	7.056
OPUS-A	AMIa	30.203	7.653	5.705	5.899	6.419
de-LSTM	AMIa	31.762	7.635	6.642	1.843	5.373
OPUS-B	AMIa	38.315	8.960	7.613	6.837	7.803
T2T-multi	AMIa	28.279	6.202	3.382	3.869	4.484
de-T2T	Auditing	38.973	11.589	17.377	18.841	15.936
OPUS-A	Auditing	38.866	10.355	19.414	18.540	16.103
de-LSTM	Auditing	21.780	10.590	12.633	11.098	11.440
OPUS-B	Auditing	38.173	10.523	18.237	17.644	15.468
T2T-multi	Auditing	22.442	7.896	8.664	11.269	9.276

Table 6: Validation BLEU scores in percents (range 0-100) for MT translations into German from ASR outputs and from the gold transcript.

MT	**avg ± std dev**
T2T-multi	2876.52 ±1804.63
T2T-multi-big	5531.30 ±3256.81
IWSLT19	275.51 ± 119.44
WMT19 Marian	184.08 ± 89.17
WMT18 T2T	421.11 ± 201.64
OPUS-B	287.52 ± 141.28
OPUS-A	263.31 ± 124.75

Table 7: Average and standard deviation time for translating one batch in validation set, in milliseconds. Bold are the candidate systems for online SLT.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of edinburgh's submissions to the wmt19 news translation task. In *WMT*.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *TSD*.

Eunah Cho, Jan Niehues, Kevin Kilgour, and Alex Waibel. 2015. Punctuation insertion for real-time spoken language translation. In *IWSLT*.

Eunah Cho, Jan Niehues, and Alex Waibel. 2012. Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *IWSLT*.

Eunah Cho, Jan Niehues, and Alex Waibel. 2017. Nmt-based segmentation and punctuation insertion for real-time spoken language translation. In *INTERSPEECH*.

Florian Dessloch, Thanh-Le Ha, Markus Müller, Jan Niehues, Thai-Son Nguyen, Ngoc-Quan Pham, Elizabeth Salesky, Matthias Sperber, Sebastian Stüker, Thomas Zenkel, and Alexander Waibel. 2018. KIT lecture translator: Multilingual speech translation with one-shot learning. In *COLING: System Demonstrations*.

Dario Franceschini et al. 2020. Removing european language barriers with innovative machine translation technology. In *LREC IWLTP*. In print.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt et al. 2018. Marian: Fast neural machine translation in C++. In *ACL System Demonstrations*.

Ondrej Klejch, Joachim Fainberg, Peter Bell, and Steve Renals. 2019. Lattice-based unsupervised test-time adaptation of neural networkacoustic models. *CoRR*, abs/1906.11521.

Philipp Koehn et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL Interactive Poster and Demonstration Sessions*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP: System Demonstrations*.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. TRuEcasIng. In *ACL*.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *ACL*.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *International Workshop on Spoken Language Translation*, pages 148–154, Pittsburgh, PA, USA.

Iain Mccowan, J Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, M Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska Masson, Wilfried Post, Dennis Reidsma, and P Wellner. 2005. The ami meeting corpus. *Int'l. Conf. on Methods and Techniques in Behavioral Research*.

Markus Müller, Thai Son Nguyen, Jan Niehues, Eunah Cho, Bastian Krüger, Thanh-Le Ha, Kevin Kilgour, Matthias Sperber, Mohammed Mediani, Sebastian Stüker, et al. 2016. Lecture translator-speech translation framework for simultaneous lecture translation. In *NAACL: Demonstrations*.

Thai-Son Nguyen, Markus Müller, Sebastian Sperber, Thomas Zenkel, Sebastian Stüker, and Alex Waibel. 2017. The 2017 KIT IWSLT Speech-to-Text Systems for English and German. In *IWSLT*.

Thai Son Nguyen, Jan Niehues, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Muller, Matthias Sperber, Sebastian Stueker, and Alex Waibel. 2020. Low latency asr for simultaneous speech translation.

Thai-Son Nguyen, Sebastian Stueker, Jan Niehues, and Alex Waibel. 2019. Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation. *arXiv preprint arXiv:1910.13296*.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic transcription for low-latency speech translation. In *Interspeech*.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. In *Proc. Interspeech 2018*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*.

Ngoc-Quan Pham, Thai-Son Nguyen, Thanh-Le Ha, Juan Hussain, Felix Schneider, Jan Niehues, Sebastian Stüker, and Alexander Waibel. 2019. The iwslt 2019 kit speech translation system. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*.

Ngoc-Quan Pham, Matthias Sperber, Elizabeth Salesky, Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2017. Kit's multilingual neural machine translation systems for iwslt 2017. In *The International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan.

Martin Popel. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel and Ondrej Bojar. 2018. Training Tips for the Transformer Model. *PBML*, 110.

Martin Popel, Dominik Macháček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. 2019. English-czech systems in wmt19: Document-level transformer. In *WMT*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.

Daniel Povey et al. 2011. The kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Anthony Rousseau, Paul Deléglise, and Yannick Esteve. 2012. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *ICLSP*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *LREC*.

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Joanna Wetesko, Marcin Chochowski, Pawel Przybysz, Philip Williams, Roman Grundkiewicz, Rico Sennrich, Barry Haddow, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. Samsung and University of Edinburgh's System for the IWSLT 2019. In *IWSLT*.