Improving End-to-End Speech Translation by Imitation-Based Knowledge Distillation with Synthetic Transcripts

Rebekka Hubert*
Computational Linguistics
Heidelberg University, Germany
hubert@cl.uni-heidelberg.de

Artem Sokolov
Google Research
Berlin, Germany
artemsok@google.com

Stefan Riezler
Computational Linguistics & IWR
Heidelberg University, Germany
riezler@cl.uni-heidelberg.de

Abstract

End-to-end automatic speech translation (AST) relies on data that combines audio inputs with text translation outputs. Previous work used existing large parallel corpora of transcriptions and translations in a knowledge distillation (KD) setup to distill a neural machine translation (NMT) into an AST student model. While KD allows using larger pretrained models, the reliance of previous KD approaches on manual audio transcripts in the data pipeline restricts the applicability of this framework to AST. We present an imitation learning approach where a teacher NMT system corrects the errors of an AST student without relying on manual transcripts. We show that the NMT teacher can recover from errors in automatic transcriptions and is able to correct erroneous translations of the AST student, leading to improvements of about 4 BLEU points over the standard AST end-to-end baseline on the English-German CoVoST-2 and MuST-C datasets, respectively. Code and data are publicly available.[1]

1 Introduction

The success of data-hungry end-to-end automatic speech translation (AST) depends on large amounts of data that consist of speech inputs and corresponding translations. One way to overcome the data scarcity issue is a knowledge distillation (KD) setup where a neural machine translation (NMT) expert (also called oracle) is distilled into an AST student model (Liu et al., 2019; Gaido et al., 2020). The focus of our work is the question of whether the requirement of high-quality source language transcripts, as in previous applications of KD to AST, can be relaxed in order to enable a wider applicability of this setup to AST scenarios where no manual source transcripts are available. Examples for such scenarios are low-resource settings (e.g., for languages without written form for which mostly only audio-translation data are available), or settings where one of the main uses of source transcripts in AST — pre-training the AST encoder from an automatic speech recognition (ASR) system— is replaced by a large-scale pre-trained ASR system (which itself is trained on hundreds of thousands hours of speech, but the original training transcripts are not available (Radford et al., 2022; Zhang et al., 2022b)). Relaxing the dependence of pre-training AST encoders on manual transcripts has recently been studied by Zhang et al. (2022a). Our focus is instead to investigate the influence of manual versus synthetic transcripts as input to the student model in an imitation learning (IL) approach (Lin et al., 2020; Hormann and Sokolov, 2021), and to lift this scenario to AST. To our knowledge, this has not been attempted before. We present a proof-of-concept experiment where we train an ASR model on a few hundred hours of speech, but discard the manual transcripts in IL training, and show that this ASR model is sufficient to enable large NMT models to function as error-correcting oracle in an IL setup where the AST student model works on synthetic transcripts. Focusing on the IL scenario, we show that one of the key ingredients to make our framework perform on synthetic ASR transcripts is to give the AST student access to the oracle's full probability distribution instead of only the expert's optimal actions. Furthermore, when comparing two IL algorithms of different power — either correcting the student output in a single step, or repairing outputs till the end of the sequence — we find that, at least in the setup of a reference-agnostic NMT teacher, the single-step correction of student errors is sufficient.

One of the general reasons for the success of our setup may be a reduction of data complexity and an increase of variations of outputs, similar to applications of KD in NMT (Zhou et al., 2020).

*All work was done at Heidelberg University.

[1] https://github.com/HubReb/imitkd_ast/releases/tag/v1.1

To investigate the special case of imitation-based KD on synthetic speech inputs, we provide a manual analysis of the NMT expert's behavior when faced with incorrect synthetic transcripts as input, or when having to correct a weak student's translation in the IL setting. We find that the NMT oracle can correct errors even if the source language input lacks semantically correct information, by utilizing its language modeling capability to correct the next-step token. This points to new uses of large pre-trained ASR and NMT models (besides initialization of encoder and decoder, respectively) as tools to improve non-cascading end-to-end AST.

2 Related Work

Imitation learning addresses a deficiency of sequence-to-sequence learning approaches, nicknamed *exposure bias* (Bengio et al., 2015; Ranzato et al., 2016), that manifests as the inference-time inability to recover from own errors, leading to disfluent or hallucinated translations (Wang and Sennrich, 2020). IL aims to replace the standard learning paradigm of teacher forcing (Williams and Zipser, 1989) (which decomposes sequence learning into independent per-step predictions, each conditioned on the golden truth context rather than the context the model would have produced on its own) by enriching the training data with examples of successful recovery from errors. We build upon two previous adaptations of IL to NMT (Lin et al., 2020; Hormann and Sokolov, 2021) and lift them to AST.

Knowledge distillation (Hinton et al., 2015) transfers the knowledge encoded in a large model, called teacher, to a far smaller student model by using the teacher to create soft labels and train the student model to minimize the cross-entropy to the teacher. KD has been successfully used for machine translation (Kim and Rush, 2016), speech recognition (Wong and Gales, 2016) and speech translation (Liu et al., 2019).

Synthetic speech translation training datasets have been used previously to train AST models: Pino et al. (2020) used an ASR-NMT model cascade to translate unlabeled speech data for augmentation. To obtain more machine translation (MT) training data, Jia et al. (2019); Pino et al. (2019) generated synthetic speech data with a text-to-speech model. Liu et al. (2019) applied KD between an NMT expert and an AST student with manual transcriptions as expert input to improve AST performance. Gaido et al. (2020) improved upon this by increasing the available training data by utilizing a MT model to translate the audio transcripts of ASR datasets into another language, yet they still use manual transcripts for distillation in the following finetuning phase.

Further attempts focused on improving AST models by utilizing MT data for multitask learning with speech and text data (Tang et al., 2021b,a; Bahar et al., 2019; Weiss et al., 2017; Anastasopoulos and Chiang, 2018), such as XSTNet (Ye et al., 2021) and FAT-MLM (Zheng et al., 2021).

A question orthogonal to ours, concerning the influence of pre-training encoder and/or decoder on source transcripts, has been investigated by Zhang et al. (2022a). They achieved competitive results without any pretraining via the introduction of parameterized distance penalty and neural acoustic feature modeling in combination with CTC regularization with translations as labels. Their question and solutions are orthogonal to ours and are likely to be yield independent benefits.

3 Imitation-based Knowledge Distillation

We view an auto-regressive NMT or AST system as a policy π that defines a conditional distribution over a vocabulary of target tokens $v \in V$ that is conditioned on the input x and the so far generated prefix $y_{<t}$: $\pi(v|y_{<t};x)$. This policy is instantiated as the output of the softmax layer. When training with teacher-forcing, the cross-entropy (CE) loss $\ell(\cdot)$ is minimized under the *empirical* distribution of training data D: $\mathcal{L}_{\text{CE}}(\pi) = \mathbb{E}_{(y,x)\sim D}[\sum_{t=1}^{T} \ell(y_t, \pi)]$. To perform well at test time we are interested in the expected loss under the *learned* model distribution: $\mathcal{L}(\pi) = \mathbb{E}_{(y,x)\sim \pi}[\sum_{t=1}^{T} \ell(y_t, \pi)]$.

As shown by Ross et al. (2011), the discrepancy between $\mathcal{L}$ and $\mathcal{L}_{\text{CE}}$ accumulates quadratically with the sequence length T, which in practice could manifest itself as translation errors. They proposed the Dagger algorithm which has linear worst-case error accumulation. It, however, relies on the existence of an oracle policy π^* that, conditioned on the same input x and the partially generated π's prefix $y_{<t}$, can produce a single next-step correction to $y_{<t}$. Ross and Bagnell (2014) further proposed the AggreVaTe algorithm which relies on an even more powerful oracle that can produce a full continuation in the task-loss optimal fashion: For NMT, this means continuing the $y_{<t}$ in a way that maximizes BLEU, as done for example in Hormann and

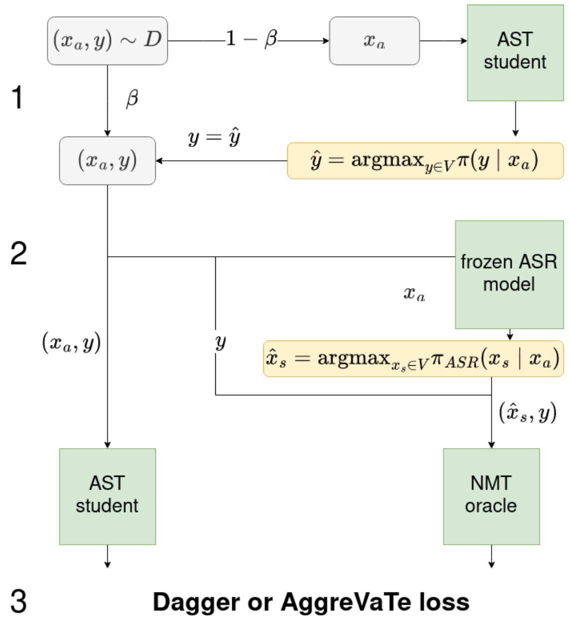


Figure 1: **Diagram of AST training with imitation learning and synthetic transcripts coming from ASR models**. (1) With probability $1-\beta$ the AST student creates a hypothesis $\hat{y}$ that replaces the reference translation y. (2) The ASR model generates the synthetic transcript $\hat{x}_s$ for the audio sample x_a to feed the NMT oracle as input. (3) Calculation of Dagger or AggreVaTe loss as shown in Algorithm 1.

Sokolov (2021).

IL for NMT We pretrain a large NMT model to serve as an oracle π^* that either simply predicts the next-step optimal output vocabulary token v_t^* given a source sentence x and any (potentially, erroneous) partial NMT student hypothesis $y_{<t}$ (Dagger):

$$v_t^* = \operatorname*{argmax}_{v \in V} \pi^*(v \mid y_{<t}; x), \tag{1}$$

or continues $y_{<t}$ till the end (AggreVaTe):

$$y_{>t}^* = \operatorname*{argmax}_{y_{>t}} \pi^*(y_{<t} + a_t + y_{\geq t} \mid y_{<t}; x), \tag{2}$$

where $y_{>t}$ is the continuation, a_t is an exploratory action, and the last argmax is implemented as beam search. The predicted v_t^* or $y_{>t}^*$ are viewed as one-step or multi-step corrections of the current policy, and the student is updated to increase the probability of the correction via the cross-entropy loss on triples (y_t, x, v_t^*) in case of Dagger, or to decrease a square loss between logit Q of the selected action a_t and the BLEU of the predicted suffix[2] from that action in case of AggreVaTe.

[2]We use the difference between the BLEU values of the full sequence and that of the prefix (Bahdanau et al., 2016).

Both algorithms proceed iteratively, where the newly generated set of triples form a provisional training data set D_i. Originally, Dagger and AggreVaTe train the student's π_i on the aggregated dataset $\cup_{j \leq i} D_j$ and use a probabilistic mixture for the current roll-out policy, which queries the oracle with probability β_i and the student otherwise. This setup guarantees that the prediction error scales at most linearly with time, unlike the quadratic scaling of the standard teacher forcing (Ross et al., 2011), which is standardly used in sequence-level KD. This makes Dagger and AggreVaTe promising candidates to improve over KD.

In our implementation, we follow Lin et al. (2020), who save memory via training on individual D_i in each iteration i, instead of training on the set union. They further speed up training by keeping the reference translation y with probability β_i, and otherwise generate a translation $\hat{y}$ of the source sentence x from the student policy (see Algorithm 1). For each t in the algorithm, AggreVaTe needs to generate an exploration token a_t and calculate the BLEU it would lead to, according to the oracle continuation starting off this action.

IL for AST Adapting Dagger and AggreVaTe to an AST student is relatively straightforward (see Figure 1): We feed the NMT oracle the source language transcript x_s of the audio data sample x_a that is also given to the AST student. We define an algorithm IKD (imitation knowledge distillation) that optimizes the cross-entropy of the student's policy w.r.t. the optimal expert prediction:

$$\mathcal{L}_{\text{IKD}}(\pi) = \mathbb{E}\left[-\sum_{t=1}^{T} \log \pi(v_t^* \mid y_{<t}; x_a)\right], \tag{3}$$

with v_t^* as in (1). Algorithm IKD$^+$ optimizes the cross-entropy w.r.t. the expert's policy:

$$\mathcal{L}_{\text{IKD}^+}(\pi) = \mathbb{E}\left[-\sum_{v \in V} \pi^*(v \mid y_{<t}; x_s) \cdot \log \pi(v \mid y_{<t}; x_a)\right]. \tag{4}$$

An important modification to these objectives that we propose in this work is to replace the gold source language transcripts x_s fed to the NMT oracle by synthetic transcripts generated by a pretrained ASR model. We call this algorithm SynthIKD, with a respective SynthIKD$^+$ variant.

4 Experiments

We experiment with English-German AST on the CoVoST2 (Wang et al., 2021) (430 hours) and

Algorithm 1: Dagger/AggreVaTe for distillation in NMT; combined from (Lin et al., 2020) and (Hormann and Sokolov, 2021).

Data: Let D be original bi-text dataset, π^* the NMT oracle policy, I the total number of iterations, T the max sequence length, Q the final logits, and B the batch size.

Initialize π_1 arbitrarily.

for $i = 1 \dots I$ **do**
- Initialize $D_i \leftarrow \emptyset$
- **for** $b = 1 \dots B$ **do**
  - Sample an example $(x, y) \sim D$.
  - Sample uniformly $u \sim [0, 1]$
  - **if** $u > \beta_i$ **then**
    - Generate $\hat{y}$ from π_i given x.
    - Replace y with $\hat{y}$.
  - **if** *Dagger* **then**
    - **for** $t = 1 \dots T$ **do**
      - Predict $v_t^* = \underset{v \in V}{\operatorname{argmax}} \, \pi^*(v \mid y_{<t}; x)$
      - Append $(y_{<t}, x, v_t^*)$ to D_i
  - **else** `// AggreVaTe`
    - Sample uniformly $t \in \{1, .., T\}$.
    - Predict $a_t = \underset{v \in V}{\operatorname{argmax}} \, \pi(v \mid y_{<t}; x)$
    - Predict $y_{>t}^* = \underset{y_{>t}}{\operatorname{argmax}} \, \pi^*(y_{>t} \mid y_{<t} + a_t; x)$
    - Append $(y_{<t}, x, a_t, \mathrm{BLEU}(y_{>t}^*))$ to D_i
- $\mathcal{L}_{\text{Dagger}} = \mathbb{E}_{D_i}\left[-\sum_{t=1}^{T} \log \pi_i(v_t^* \mid y_{<t}; x)\right]$
- $\mathcal{L}_{\text{AggreVaTe}} = \mathbb{E}_{D_i}\left[\sum_{t=1}^{T} \left(\sigma(Q(a_t \mid y_{<t}; x)) - \mathrm{BLEU}(y_{>t}^*)\right)^2\right]$
- Let $\pi_{i+1} = \pi_i - \alpha_i \cdot \frac{\partial \mathcal{L}}{\partial \pi_i}$.

the MuST-C (Di Gangi et al., 2019) datasets (408 hours)[3]. As expert model, we use the Transformer from Facebook's submission to WMT19 (Ng et al., 2019), which is based on the Big Transformer architecture proposed by (Vaswani et al., 2017). Our sequence-to-sequence models for students are RNNs and Base Transformers. All models are based on the `fairseq` framework (Ott et al., 2019; Wang et al., 2020), but use different settings of meta-parameters and preprocessing than the default models. More details on models, meta-parameters and training settings are given in the Appendix A.

Our training setups are summarized in Table 1. We compare our trained student models with several baseline approaches: "Standard" denotes AST trained by teacher forcing on ground truth targets with a label smoothing (Szegedy et al., 2016) factor of 0.1. KD^+ (Liu et al., 2019) denotes word-level knowledge distillation between the expert's and student's full output probability. IKD and IKD^+ denote imitation knowledge distillation, where student model is corrected by the empirical distribution of the optimal expert actions or the full expert policy (Lin et al., 2020), respectively. SynthIKD and $\mathrm{SynthIKD}^+$ are our variants with synthetic transcripts. We used the same same exponential decay schedule ($\beta = \frac{1}{T}$) used by (Lin et al., 2020) as early experiments showed that this performed best in our setup.

Variant	Expert Input	Loss
Standard	-	CE
KD^+ (Liu et al., 2019)	gold	CE
$\mathrm{SynthKD}^+$	synthetic	CE
IKD (Lin et al., 2020)	gold	$\mathcal{L}_{\text{IKD}}$
IKD^+ (Lin et al., 2020)	gold	$\mathcal{L}_{\text{IKD+}}$
SynthIKD (ours)	synthetic	$\mathcal{L}_{\text{IKD}}$
$\mathrm{SynthIKD}^+$ (ours)	synthetic	$\mathcal{L}_{\text{IKD+}}$

Table 1: Summary of training variants: "Standard" denotes AST trained via cross-entropy (CE) on ground truth targets with a label smoothing. KD^+ denotes word-level knowledge distillation between the expert's and student's full output probability. IKD and IKD^+ denote imitation knowledge distillation where student model is corrected by the optimal expert action or the full expert policy (Lin et al., 2020), respectively. SynthIKD and $\mathrm{SynthIKD}^+$ are our variants with synthetic transcripts. Expert Input indicates whether the NMT expert is given the original transcripts from the dataset or synthetic transcripts created by ASR. All IKD methods use the exponential decay schedule for β that (Lin et al., 2020) found to work best.

All AST models' encoders are initialized with the encoder of the corresponding ASR model, trained on the respective datasets with cross-entropy and the label-smoothing factor of 0.1. Because of the relatively small size of these datasets, our experiments should seen as proof-of-concept, showing that ASR models trained on a few hundred hours of audio provide synthetic transcripts of sufficient quality to enable imitation-based KD for AST. The standalone performance of our ASR models is listed in Table 2.

4.1 Feasibility of Oracle Correction

The idea of using synthetic transcripts in place of gold transcripts has merit only if the NMT oracle's translations have higher quality than the translations the AST model generates. Therefore, we first

[3] We also experimented with a smaller Europarl-ST dataset and to save space we report results in Appendix B. Overall, they are similar to these on larger datasets.

Model	CoVoST2		MuST-C	
	dev	test	dev	test
RNN	26.68	33.94	23.42	24.44
Transformer	20.93	26.60	21.10	20.68

Table 2: WER$\downarrow$ results for ASR models pretrained on CoVoST2 and MuST-C. These models are used to create the synthetic transcripts for respective experiments. Standard development and test splits were used for CoVoST2. For MuST-C, we tested on `tst-COMMON`.

verify if the NMT oracle is capable of completing an AST models' partial hypotheses $y_{<t}$ while improving quality at the same time.

We follow Lin et al. (2020) and let the AST models trained with label-smoothed CE on ground truth targets translate the audio input with greedy decoding up to a randomly chosen time step. Then, we feed the NMT expert the gold transcript as input and the partial translation as prefix, and let the oracle finish the translation with greedy decoding.

As Table 2 shows, the out-of-the-box ASR performance is relatively low (high WER), so errors in synthetic transcripts will be propagated through the NMT oracle. The question is whether the expert's continuation can be of higher quality than the student's own predictions despite the partially incorrect synthetic transcripts. In Table 3, lines 1 and 2 (or, 5 and 6) set the lower (end-to-end) and upper (cascade) bounds on the performance. We see that the NMT expert is able to complete the student hypotheses successfully (lines 3, 4 and 7, 8), bringing gains in both gold and synthetic setups, and reaching the upper bound (lines 3 vs. 2 and 7 vs. 6) for gold ones. Although the mistakes in the synthetic transcripts do result in lower BLEU scores (lines 4 and 8) they still improve over the AST student complete translations (lines 1 and 5).

4.2 Main Results

Table 4 shows the main results of applying Algorithm 1 for training an AST student with imitation-based knowledge distillation on CoVoST2 and MuST-C.

Dagger First we present results for the Dagger algorithm. In Table 4, for both CoVoST2 and MuST-C models, Dagger with the Transformer architecture outperforms all baselines[4], and matching full teacher distributions (the '+'-versions of losses) gives consistent gains. Distillation with RNNs, on the other hand, fails to improve BLEU scores over baselines, most likely due to their overall lower translation quality. This leads to the student hypotheses that are too far from the reference so that the expert's one-step corrections are not able to correct them.

The results show that Transformers and RNNs with synthetic transcripts show statistically insignificant differences in performance to the ones that are using gold transcripts. This is notable since the partially synthetic transcripts provided to the NMT oracle are often incorrect, yet do not result in a noticeable effect on the final student performance if used in the IL framework. A similar observation can be made when comparing the use of gold transcripts versus synthetic transcripts: Transformers on both datasets perform comparably and erroneous transcripts do not seem to harm the trained AST model.

AggreVaTe Finally, we evaluate the performance of AggreVaTe both with gold and synthetic transcripts. During training we targeted and evaluated with the non-decomposable BLEU metric (i.e. training with sentence-BLEU and evaluating with corpus-BLEU) as well as with the decomposable TER metric (Table 5). Following Hormann and Sokolov (2021) we warm-started AggreVaTe with differently trained standard or Dagger models, and trained with AggreVaTe objectives for up to 50 epochs with early stopping on respective development sets.

Surprisingly, we found that AggreVaTe does not bring additional benefits on top of Dagger despite the promise for a better matching between training and inference objectives. Also there is no significant difference between the results with the TER rewards objective and sentence-BLEU rewards on both CoVoST2 and MuST-C. We explain these results by the sufficiency of one-step corrections to correct a "derailed" student, with little benefit of continuing demonstration till the end of translation. The fact that Dagger turns out to reap all of the benefits from training with IL is good news in general, since running beam search during training (to get AggreVaTe's full continuations) is more expensive than greedily selecting one action (as does Dagger).

4.3 Quality of Synthetic Transcripts

In this section, we investigate explanations for the high performance of Dagger on synthetic tran-

[4] p-value < 0.005 using the paired approximate randomization test (Riezler and Maxwell, 2005)

Architecture	Hypotheses	#	Decoding Setup	Source Transcripts	dev-BLEU↑
RNN	full	1	AST	-	11.9
		2	ASR transcribes, NMT expert translates	-	21.8
	partial	3	AST starts, NMT expert completes	gold	21.9
		4	AST starts, NMT expert completes	synthetic	15.6
Transformer	full	5	AST	-	16.7
		6	ASR transcribes, NMT expert translates	-	25.4
	partial	7	AST starts, NMT expert completes	gold	25.4
		8	AST starts, NMT expert completes	synthetic	19.9

Table 3: Feasibility experiment: BLEU score on CoVoST2 development set of NMT expert's completion of AST model full or partial hypotheses with greedy decoding; *gold* denotes the usage of the dataset's source language transcripts as NMT inputs and *synthetic* denotes synthetic transcripts created by the respective ASR model.

Achitecture		Models	CoVoST2 dev	CoVoST2 test	MuST-C dev	MuST-C test
RNN	baseline	Standard	13.6	10.0	14.6	14.1
		KD$^+$	**14.6**	**11.1**	**17.9**	**17.2**
		IKD$^+$	13.1	10.1	15.7	14.9
	ours	SynthKD$^+$	14.1	10.6	16.9	15.9
		SynthIKD$^+$	12.8	9.7	16.3	15.1
Transformer	baseline	Standard	18.4	14.2	19.5	19.4
		KD$^+$	21.3	17.7	17.7	22.2
		IKD$^+$	**21.8**	18.4	23.2	23.3
	ours	SynthKD$^+$	21.7	18.0	22.5	22.6
		SynthIKD$^+$	**21.8**	**18.5**	**23.5**	**23.5**

Table 4: Main results: RNN and Transformer student models trained on expert inputs and loss variants of Table 1, using Dagger for IL. We used the `tst-COMMON` as the test set for MuST-C. (Synth)IKD is not included since its performance is worse than (Synth)KD$^+$. Transformers trained with IL outperform all baselines, while pure KD is the best for generally lower-quality RNN-based models. Synthetic transcripts do not harm performance for Transformer student models.

scripts: The first hypothesis is that synthetic transcripts are already "good enough" and per-step IL corrections add nothing on top. Second, the gains could be due to the known NMT "auto-correcting" ability and due to general robustness to the quality of the source (cf. the success of back-translation in NMT), and all benefits could be reached with KD alone. To test both hypotheses, we create new training datasets where we replace references with translated gold or synthetic transcripts by the same NMT expert with beam size 5. Evaluating on the unmodified references, we trained Transformer-based baselines and the IL model from Lin et al. (2020) on these two new corpora.

As Table 6 shows, Transformer KD$^+$ trained on translated gold transcripts outperforms its counterparts trained on translated synthetic transcripts, confirming errors in the synthetic transcripts. This refutes the first hypothesis.

Regarding the second hypothesis, we compare the KD$^+$ to IKD$^+$ from the synthetic translated part in Table 6. Were "auto-correction" sufficient we would see similar performance in both lines. This rejects the second hypothesis and suggests that IL adds value on top of general NMT robustness to inputs.

4.4 Qualitative Analysis

Here, we perform a human evaluation of successful IL corrections, aiming at an explanation of the performance of Dagger on synthetic transcripts.

We randomly sample 100 examples from the CoVoST2 training set on which the ASR Transformer has a non-zero sentence-wise word error rate, and compare the NMT expert's probability distributions over time for the given synthetic transcripts. From the WER histogram in Figure 2 we see that most of the sentences have a single-digit number of errors.

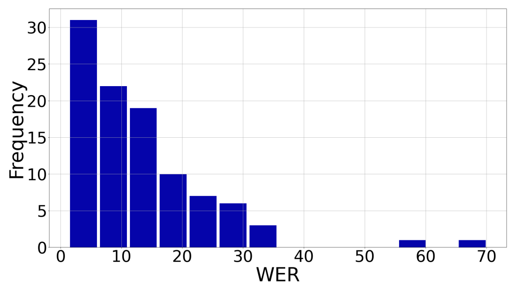


Figure 2: Histogram of sentence-wise WER of ASR Transformer on 100 samples from CoVoST2.

As WER cannot be used to differentiate between small but inconsequential (to the understanding of the sentence) errors and mistakes that change the meaning of the sentence, we further compare the generated transcript to the gold transcript *and* look

IL Algorithm	Model	Data	CoVoST2 BLEU↑ dev	CoVoST2 BLEU↑ test	CoVoST2 TER↓ dev	CoVoST2 TER↓ test	MuST-C BLEU↑ dev	MuST-C BLEU↑ test	MuST-C TER↓ dev	MuST-C TER↓ test
Dagger	Standard	gold	18.4	14.2	69.1	77.1	19.5	19.4	70.8	69.4
	IKD$^+$	gold	21.8	18.4	63.7	70.0	23.2	23.3	67.4	65.6
	SynthIKD$^+$	synth	21.8	**18.5**	63.6	69.8	**23.5**	23.5	67.2	65.6
	Warm-start Model	**Data**	**BLEU↑ dev**	**BLEU↑ test**	**TER↓ dev**	**TER↓ test**	**BLEU↑ dev**	**BLEU↑ test**	**TER↓ dev**	**TER↓ test**
			sentence-BLEU reward-to-go							
AggreVaTe	Standard	gold	18.7	14.6	68.2	76.0	19.9	19.9	70.2	68.1
	Standard	synth	18.7	14.6	68.2	75.9	20.0	19.7	70.1	68.7
	IKD$^+$	gold	**22.1**	**18.5**	**63.1**	69.6	**23.5**	23.4	67.4	65.7
	SynthIKD$^+$	synth	**22.1**	**18.5**	**63.1**	69.7	**23.5**	**23.6**	**67.0**	65.6
			TER reward-to-go							
	Standard	gold	18.7	14.7	67.8	75.4	20.0	19.9	70.0	68.5
	Standard	synth	18.7	14.6	67.9	75.6	19.9	19.6	69.8	68.4
	IKD$^+$	gold	22.0	**18.5**	**63.1**	**69.4**	23.3	23.4	67.3	65.5
	SynthIKD$^+$	synth	**22.1**	**18.5**	**63.1**	69.6	**23.5**	**23.6**	**67.0**	**65.3**

Table 5: Comparison of Dagger with warm-started AggreVaTe with a maximum of 50 epochs on CoVoST2 and MuST-C.

Training	CoVoST2 dev	CoVoST2 test	MuST-C dev	MuST-C test
training on translated gold transcripts				
Standard	18.1	14.9	20.0	20.0
KD$^+$	21.3	17.6	23.4	23.1
IKD$^+$	22.6	18.6	23.5	23.7
training on translated synthetic transcripts				
Standard	17.8	14.2	19.2	19.2
KD$^+$	20.2	16.5	22.1	22.5
IKD$^+$	21.0	17.4	23.0	23.1

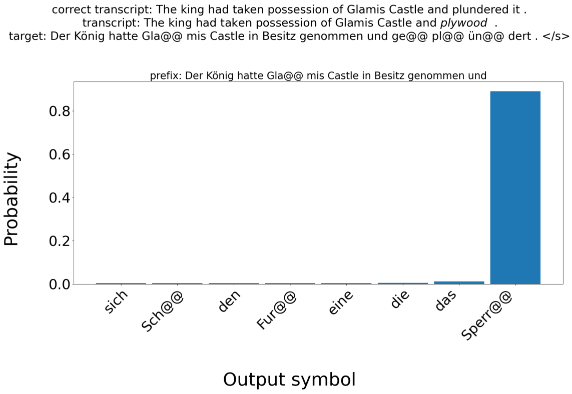
Table 6: BLEU scores of Transformer models trained on the training set with original references replaced by translations of gold and synthetic transcripts in comparison to using the original training set (lower part of Table 4).

Error Type	Freq
omitted tokens	2
surface form error	17
contentual error, correct target in top-1	5
contentual error, correct target in top-8	12
critical error, expert predicts correctly due to prefix	32
critical error, expert does not predict correctly	32

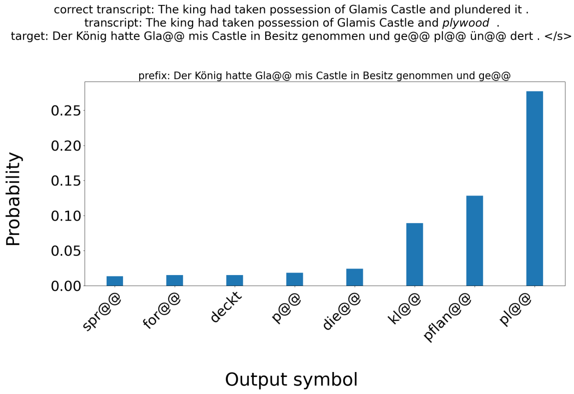
Table 7: Error types in the synthetic transcripts created by the ASR model.

at the top-8 output probabilities of the expert at each time step for each sample to classify each error in the synthetic transcripts. We further feed the sampled sentences to the NMT expert and find that in 36 out of 100 samples (all but the last two lines in Table 7), the expert is able to generate output probability distributions that favor the correct target token despite errors in the transcript. Although the expert can put large probability mass on the correct target token, whether it does so depends on the error type in the generated transcript. The expert is often able to deal with surface form errors, such as different spellings, punctuation errors and different word choice (17 occurrences). When the synthetic transcripts contain critical errors, e.g. partially hallucinated transcript, the expert is still able to produce the correct translation if the missing or wrong information can be still inferred from the prefix (32 occurrences).

Next, we verify that the decoder language modeling capability is what primarily drives the correction process. We do this by feeding parts of reference translations as prefix conditioned on erroneous synthetic transcripts. Consider the transcript "The king had taken possession of Glamis Castle and plywood." generated by the ASR model. Its gold transcript reads "plundered it" instead of "plywood". In Figure 3 we illustrate output probabilities that the expert generates in the last time-steps. Assume as in Figure 3a that the expert has been given the prefix "Der König hatte Glamis Castle in Besitz genommen und". According to the output probabilities, the next output symbol is the subword unit "Sperr" and would not be a proper correction. At the next timestep, however, the last symbol in the prefix is the subword unit "ge" and, as Figure 3b shows, the expert, being driven by its decoder language modeling capability, puts highest probabilities on subword units that are most likely

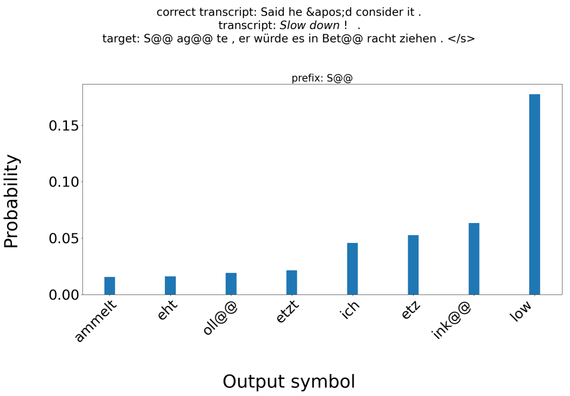


(a) with $y_{<t}$ = “Der König hatte Glamis Castle in Besitz genommen und ”

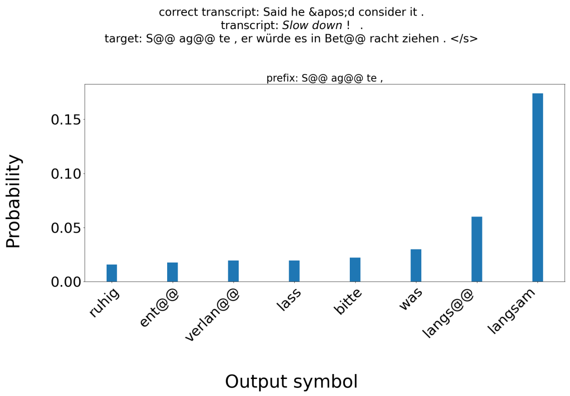


(b) with $y_{<t}$ = “Der König hatte Glamis Castle in Besitz genommen und ge”

Figure 3: NMT expert top-8 output probabilities when translating the incorrect synthetic transcript “The king had taken possession of Glamis Castle and plywood it.”



(a) with $y_{<t}$ = “S”



(b) with $y_{<t}$ = “Sagte , ”

Figure 4: NMT expert top-8 output probabilities when translating the incorrect synthetic transcript “Slow down!”

to produce a fluent output (the correct one “pl@@”, and less probable “pflan@@” and “kl@@” rather then paying attention to the (wrong) information in the synthetic transcripts.

Similar situations can be observed in samples with entirely wrong synthetic transcripts. In Figure 4, the expert has received the synthetic transcript “Slow down!” as input, which shares no meaning with the gold transcript “Said he’d consider it.” As shown in Figure 4a, the expert assigns the highest probability to “@@low” if it is given the prefix “S” (as the expert has a shared vocabulary, it can complete the output this way), which turns the partial translation into an exact copy of the transcript. Again, the top-8 predictions do not share similar meaning with the transcript. After, in Figure 4b, the expert has received the prefix “Sagte,”, it still attempts to complete $y_{<t}$ by generating output symbols that would turn y into a valid translation of this wrong transcript (“langsam” (slow), “ruhig” (quiet), “langs@@”)) with the rest of options being mostly driven by language modeling rather then reproducing source semantics (“ent@@”, “verlan@@”).

Overall, with the SynthIKD$^+$ training, the expert induces smoothed output distributions and fluency on the student more than it enforces the student to predict one-hot labels produced by the expert as is done by sequence-level KD.

5 Conclusion

We showed that a pretrained NMT model can successfully be used as an oracle for an AST student, without requiring gold source language transcripts as in previous approaches to imitation learning for AST. This widens the applicability of imitation learning approaches to datasets that do not contain manual transcripts or to pre-trained ASR models for which training transcripts are not available. Our qualitative analysis suggests an explanation of the fact that the NMT oracle is robust against mismatches between manual and synthetic transcripts by its large language model capabilities that allow it to continue the prefix solely based on its learned contextual knowledge.

6 Limitations

There are several limitations of this study. First, it is done on one language pair although we believe this should not qualitatively change the results. Second, only one set of standard model sizes was evaluated for AST student and NMT expert; we expect it be in line with reported findings for NMT (Ghorbani et al., 2021). Finally, while alluding to the potential of using large pre-trained ASR models instead of manual transcripts for IL-based AST, our current work must be seen as a proof-of-concept experiment where we train ASR models on a few hundred hours of audio, and discard the manual transcripts in IL training, showing the feasibility of our idea.

Acknowledgements

The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech 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)*, pages 82–91, New Orleans, Louisiana. Association for Computational Linguistics.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019. A comparative study on end-to-end speech to text translation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 792–799.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 6224–6228. IEEE.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: FBK@IWSLT2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online. Association for Computational Linguistics.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2021. Scaling laws for neural machine translation. *CoRR*, abs/2109.07740.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Luca Hormann and Artem Sokolov. 2021. Fixing exposure bias with imitation learning needs powerful oracles. *CoRR*, abs/2109.04114.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6121–6133, Online. Association for Computational Linguistics.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-End Speech Translation with Knowledge Distillation. In *Proc. Interspeech 2019*, pages 1128–1132.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D. McCarthy, and Deepak Gopinath. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-Training for End-to-End Speech Translation. In *Proc. Interspeech 2020*, pages 1476–1480.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *CoRR*, abs/2212.04356.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Stéphane Ross and Andrew Bagnell. 2014. Reinforcement and imitation learning via interactive no-regret learning. *CoRR*, abs/1406.5979.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA. PMLR.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A general multi-task learning framework to leverage text data for speech to text tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. CoVoST 2 and Massively Multilingual Speech Translation. In *Interspeech*, pages 2247–2251.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *ACL*.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2625–2629. ISCA.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Jeremy H.M. Wong and Mark J.F. Gales. 2016. Sequence Student-Teacher Training of Deep Neural Networks. In *Proc. Interspeech 2016*, pages 2761–2765.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Proc. of INTERSPEECH*.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022a. Revisiting end-to-end speech-to-text translation from scratch. In *International Conference on Machine Learning*, pages 26193–26205. PMLR.

Yu Zhang, Daniel S. Park, Wei Han, James Qin, Anmol Gulati, Joel Shor, Aren Jansen, Yuanzhong Xu, Yanping Huang, Shibo Wang, Zongwei Zhou, Bo Li, Min Ma, William Chan, Jiahui Yu, Yongqiang Wang, Liangliang Cao, Khe Chai Sim, Bhuvana Ramabhadran, Tara N. Sainath, Francoise Beaufays, Zhifeng Chen, Quoc V. Le, Chung-Cheng Chiu, Ruoming Pang, and Yonghui Wu. 2022b. BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *International Conference on Machine Learning*, pages 12736–12746. PMLR.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations (ICLR)*.

Model	BLEU↑ dev	test
original dataset		
Standard	13.8	14.4
KD$^+$	17.4	17.8
SynthKD$^+$	17.5	18.0
IKD$^+$	17.0	17.1
SynthIKD$^+$	17.0	17.0
translated gold training set		
Standard	15.3	15.3
KD$^+$	**18.2**	**18.4**
IKD	16.8	17.0
IKD+	17.1	17.5
synthetic translated training set		
Standard	14.7	15.3
KD$^+$	17.0	16.8
IKD	16.1	16.0
IKD+	16.3	16.6

Table A.1: Results on Europarl-ST

A Models, Meta-parameters, and Training Settings

We use the speech-to-text module of the `fairseq` framework (Ott et al., 2019; Wang et al., 2020) for all experiments and train both RNNs with convolutional layers for time dimension reduction as in Berard et al. (2018) and small Transformers as in Wang et al. (2020), which consist of a convolutional subsampler of two convolutional blocks, followed by 12 encoder layers and 6 decoder layers. The dimension of the self-attention layer is 256 and the number of attention heads is set to 4. For the NMT oracle, we use the trained Transformer model from the Facebook's submission to WMT19 (Ng et al., 2019) [5], which is based on the big Transformer (Vaswani et al., 2017) which has 6 encoder and decoder layers, 16 attention heads and the dimension of 1024, with a larger feed-forward layer size of 8192. This NMT oracle had been trained on all available WMT19 shared task en-de training data and on back-translated english and german portions of the News crawl dataset.

For all models we use Adam (Kingma and Ba, 2015) with gradient clipping at norm 10 and stop training if the development set loss has not improved for 10 epochs. For RNN architectures, we return the best model on the development set and for Transformers, we create each model by averaging over the last 10 checkpoints. For inference, a beam size of 5 was used and we report case-sensitive detokenized BLEU (Papineni et al., 2002) computed with sacreBLEU (Post, 2018). We tested for statistical significance with the paired approximate randomization test (Riezler and Maxwell, 2005).

For all experiments, we preprocess the datasets as follows: We extract log mel-scale filterbanks with a povey window, 80 bins, a pre-emphasis filter of 0.97, a frame length of 25 ms and a frame shift of 10 ms. We discard samples with less than five or more than 3000 frames and subtract the mean of the waveform from each frame and zero-pad the FFT input. For the text data, we normalize punctuation, remove non-printable characters, use the Moses tokenizer (Koehn et al., 2007) for tokenization and segment the text data into subword units with byte-pair encoding (Sennrich et al., 2016). We used a random seed of 1 for all experiments.

We list the final used and best performing hyperparameters in Table A.2. Parameters that do not differ between the training methods are not repeated in the table. We determine the batch size by defining a maximum number of input frames in the batch.

[5] As the WMT19 submission consists of an ensemble of models, we use the `model1.pt` for our experiments.

B Europarl-ST

We performed additional experiments on the Europarl-ST dataset (Iranzo-Sánchez et al., 2020) that provides 83 hours of speech training data. We train RNNs with a learning rate of 0.002 and a max-tokens size of 40,000 for a total of 80,000 updates. All other hyper-parameters are the same as listed for MuST-C in Table A.2. We only trained RNNs on the Europarl-ST dataset due to the small amount of available training data. We present the results in Table A.1.

Both improvements over standard training and by training on both the gold-translated and synthetic-translated translated training data correspond with the results presented in the main body of this work. Hence, the results presented here hold for relatively small datasets, too.

C Additional Example of NMT Expert Correction

Here we give another example of the NMT expert predicting the correct output token despite receiv-

Model	**Hyperparameter**	**CoVoST2**	**MuST-C**
RNN			
standard	learning rate	1e-3	1e-3
	max-tokens	60000	40000
	scheduler	fixed	fixed
	warmup-updates	20000	20000
	encoder freezing updates	10000	10000
	dropout	0.2	0.2
KD^{+}	learning rate	1e-3	2e-3
	max-tokens	50000	30000
	warmup-updates	25000	20000
	max-update	250000	250000
	encoder-freezing updates	20000	10000
	scheduler	inverse square root	inverse square root
Transformer			
ASR	learning rate	2e-3	1e-3
	max-tokens	50000	40000
	max-update	60000	100000
	scheduler	inverse square root	inverse square root
	warmup-updates	10000	10000
	dropout	0.15	0.1
AST			
standard	learning rate	2e-3	2e-3
	max-update	30000	100000
	encoder-freezing updates	1000	-
KD^{+}	max-tokens	50000	20000

Table A.2: list of hyperparameters that are dependent on model and dataset; we list only parameters which differ from the previous model's

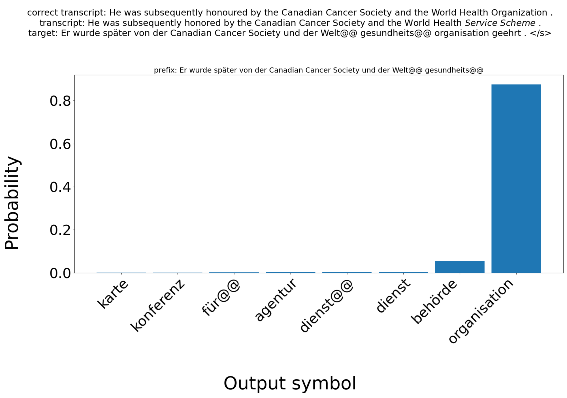


Figure C.1: NMT expert top-8 output probabilities with $y_{<t}$ = " Er wurde später von der Canadian Cancer Society und der Weltgesundheits".

ing a transcript with incomplete or false information.

Figure C.1 shows the expert's output probabilities in response to receiving factually false information in the transcript. The ASR model transcribed "World Health Organization" as "World Health Service Scheme", yet the expert produces a probability distribution that is skewed in favor of the correct proper name due to its learned context knowledge. Note that the probability of generating the correct output token "organisation" (organization) is above 0.8.