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 of 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.
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 \( \pi \) 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 \( D: \mathcal{L}_{CE}(\pi) = \mathbb{E}_{(y,x) \sim D} \sum_{t=1}^{T} \ell(y_t, \pi) \). 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 \( \mathcal{L}_{CE} \) 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 \( \pi^* \) that, conditioned on the same input \( x \) and the partially generated \( \pi^* \)’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
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 \( \pi_t \) on the aggregated dataset \( \bigcup_{j \leq i} D_j \) and use a probabilistic mixture for the current roll-out policy, which queries the oracle with probability \( \beta_i \) 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 \( \beta_i \), 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 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:

\[
L_{IKD}(\pi) = \mathbb{E} \left[ -\sum_{t=1}^{T} \log \pi(v_t | y_{<t}; x_a) \right],
\]

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

\[
L_{IKD^+}(\pi) = \mathbb{E} \left[ -\sum_{v \in V} \pi^*(v | y_{<t}; x_s) \cdot \log \pi(v | y_{<t}; x_a) \right].
\]

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
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 optimal expert action or the full expert policy (Lin et al., 2020), respectively. SynthIKD and 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 $\beta$ that (Lin et al., 2020) found to work best.

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 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 $\beta$ that (Lin et al., 2020) found to work best.

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...
Table 2: WER↓ 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>CoVoST2 dev</th>
<th>CoVoST2 test</th>
<th>MuST-C dev</th>
<th>MuST-C test</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>26.68</td>
<td>33.94</td>
<td>23.42</td>
<td>24.44</td>
</tr>
<tr>
<td>Transformer</td>
<td>20.93</td>
<td>26.60</td>
<td>21.10</td>
<td>20.68</td>
</tr>
</tbody>
</table>

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-
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.

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 t.st-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.

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↑ | TER↓ | MuST-C BLEU↑ | TER↓  
--- | --- | --- | --- | --- | --- | ---  
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↑ | TER↓ | BLEU↑ | TER↓  
--- | --- | --- | --- | --- | ---  
AggreVaTe | Standard | gold | 18.7 | 14.6 | 68.2 | 75.8 | 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  

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

<table>
<thead>
<tr>
<th>Training</th>
<th>CoVoST2</th>
<th>MuST-C</th>
</tr>
</thead>
</table>
| dev | test | dev | test | dev | test  
training on translated gold transcripts | 18.1 | 14.9 | 20.0 | 20.0  
Standard | 21.3 | 17.6 | 23.4 | 23.1  
KD⁺ | 22.6 | 18.6 | 23.5 | 23.7  
training on translated synthetic transcripts | 17.8 | 14.2 | 19.2 | 19.2  
Standard | 20.2 | 16.5 | 22.1 | 22.5  
KD⁺ | 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).

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
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


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


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, Lianliang 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.


<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
</tr>
<tr>
<td><strong>original dataset</strong></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>13.8</td>
</tr>
<tr>
<td>KD+</td>
<td>17.4</td>
</tr>
<tr>
<td>SynthKD+</td>
<td>17.5</td>
</tr>
<tr>
<td>IKD+</td>
<td>17.0</td>
</tr>
<tr>
<td>SynthIKD+</td>
<td>17.0</td>
</tr>
<tr>
<td><strong>translated gold training set</strong></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>15.3</td>
</tr>
<tr>
<td>KD+</td>
<td><strong>18.2</strong></td>
</tr>
<tr>
<td>IKD</td>
<td>16.8</td>
</tr>
<tr>
<td>IKD+</td>
<td>17.1</td>
</tr>
<tr>
<td><strong>synthetic translated training set</strong></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>14.7</td>
</tr>
<tr>
<td>KD+</td>
<td>17.0</td>
</tr>
<tr>
<td>IKD</td>
<td>16.1</td>
</tr>
<tr>
<td>IKD+</td>
<td>16.3</td>
</tr>
</tbody>
</table>

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.

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-
<table>
<thead>
<tr>
<th>Model</th>
<th>Hyperparameter</th>
<th>CoVoST2</th>
<th>MuST-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>learning rate</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td></td>
<td>max-tokens</td>
<td>60000</td>
<td>40000</td>
</tr>
<tr>
<td></td>
<td>scheduler</td>
<td>fixed</td>
<td>fixed</td>
</tr>
<tr>
<td></td>
<td>warmup-updates</td>
<td>20000</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>encoder freezing updates</td>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>dropout</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>KD+</td>
<td>learning rate</td>
<td>1e-3</td>
<td>2e-3</td>
</tr>
<tr>
<td></td>
<td>max-tokens</td>
<td>50000</td>
<td>30000</td>
</tr>
<tr>
<td></td>
<td>warmup-updates</td>
<td>25000</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>max-update</td>
<td>250000</td>
<td>250000</td>
</tr>
<tr>
<td></td>
<td>encoder-freezing updates</td>
<td>20000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>scheduler</td>
<td>inverse square root</td>
<td>inverse square root</td>
</tr>
<tr>
<td>Transformer</td>
<td>learning rate</td>
<td>2e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td></td>
<td>max-tokens</td>
<td>50000</td>
<td>40000</td>
</tr>
<tr>
<td></td>
<td>max-update</td>
<td>60000</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>scheduler</td>
<td>inverse square root</td>
<td>inverse square root</td>
</tr>
<tr>
<td></td>
<td>warmup-updates</td>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>dropout</td>
<td>0.15</td>
<td>0.1</td>
</tr>
<tr>
<td>AST</td>
<td>learning rate</td>
<td>2e-3</td>
<td>2e-3</td>
</tr>
<tr>
<td></td>
<td>max-update</td>
<td>30000</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>encoder-freezing updates</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>KD+</td>
<td>max-tokens</td>
<td>50000</td>
<td>20000</td>
</tr>
</tbody>
</table>

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 $y_{<t} = "Er wurde später von der Canadian Cancer Society und der Weltgesundheits"$.

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.