Searchable Hidden Intermediates for End-to-End Models of Decomposable Sequence Tasks

Siddharth Dalmia **Brian Yan** **Vikas Raunak** **Florian Metze** **Shinji Watanabe**
Language Technologies Institute, Carnegie Mellon University, USA
{sdalmia,byan}@cs.cmu.edu

Abstract

End-to-end approaches for sequence tasks are becoming increasingly popular. Yet for complex sequence tasks, like speech translation, systems that cascade several models trained on sub-tasks have shown to be superior, suggesting that the compositionality of cascaded systems simplifies learning and enables sophisticated search capabilities. In this work, we present an end-to-end framework that exploits compositionality to learn *searchable* hidden representations at intermediate stages of a sequence model using decomposed sub-tasks. These hidden intermediates can be improved using beam search to enhance the overall performance and can also incorporate external models at intermediate stages of the network to re-score or adapt towards out-of-domain data. One instance of the proposed framework is a Multi-Decoder model for speech translation that extracts the *searchable hidden intermediates* from a speech recognition sub-task. The model demonstrates the aforementioned benefits and outperforms the previous state-of-the-art by around +6 and +3 BLEU on the two test sets of Fisher-CallHome and by around +3 and +4 BLEU on the English-German and English-French test sets of MuST-C.[1]

1 Introduction

The principle of compositionality loosely states that a complex whole is composed of its parts and the rules by which those parts are combined (Lake and Baroni, 2018). This principle is present in engineering, where task decomposition of a complex system is required to assess and optimize task allocations (Levis et al., 1994), and in natural language, where paragraph coherence and discourse analysis rely on decomposition into sentences (Johnson, 1992; Kuo, 1995) and sentence level semantics relies on decomposition into lexical units (Liu et al., 2020b).

Similarly, many sequence-to-sequence tasks that convert one sequence into another (Sutskever et al., 2014) can be decomposed to simpler sequence subtasks in order to reduce the overall complexity. For example, speech translation systems, which seek to process speech in one language and output text in another language, can be naturally decomposed into the transcription of source language audio through automatic speech recognition (ASR) and translation into the target language through machine translation (MT). Such cascaded approaches have been widely used to build practical systems for a variety of sequence tasks like hybrid ASR (Hinton et al., 2012), phrase-based MT (Koehn et al., 2007), and cascaded ASR-MT systems for speech translation (ST) (Pham et al., 2019).

End-to-end sequence models like encoder-decoder models (Bahdanau et al., 2015; Vaswani et al., 2017), are attractive in part due to their simplistic design and the reduced need for hand-crafted features. However, studies have shown mixed results compared to cascaded models particularly for complex sequence tasks like speech translation (Inaguma et al., 2020) and spoken language understanding (Coucke et al., 2018). Although direct target sequence prediction avoids the issue of error propagation from one system to another in cascaded approaches (Tzoukermann and Miller, 2018), there are many attractive properties of cascaded systems, missing in end-to-end approaches, that are useful in complex sequence tasks.

In particular, we are interested in (1) the strong search capabilities of the cascaded systems that compose the final task output from individual system predictions (Mohri et al., 2002; Kumar et al., 2006; Beck et al., 2019), (2) the ability to incorporate external models to re-score each individual system (Och and Ney, 2002; Huang and Chiang, 2007), (3) the ability to easily adapt individual components towards out-of-domain data (Koehn and Schroeder, 2007; Peddinti et al., 2015), and finally

[1] All code and models are released as part of the ESPnet toolkit: https://github.com/espnet/espnet.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1882–1896
June 6–11, 2021.

(4) the ability to monitor performance of the individual systems towards the decomposed sub-task (Tillmann and Ney, 2003; Meyer et al., 2016).

In this paper, we seek to incorporate these properties of cascaded systems into end-to-end sequence models. We first propose a generic framework to learn *searchable hidden intermediates* using an auto-regressive encoder-decoder model for any decomposable sequence task (§3). We then apply this approach to speech translation, where the intermediate stage is the output of ASR, by passing continuous hidden representations of discrete transcript sequences from the ASR sub-net decoder to the MT sub-net encoder. By doing so, we gain the ability to use beam search with optional external model re-scoring on the hidden intermediates, while maintaining end-to-end differentiability. Next, we suggest mitigation strategies for the error propagation issues inherited from decomposition.

We show the efficacy of *searchable intermediate representations* in our proposed model, called the Multi-Decoder, on speech translation with a 5.4 and 2.8 BLEU score improvement over the previous state-of-the-arts for Fisher and CallHome test sets respectively (§6). We extend these improvements by an average of 0.5 BLEU score through the aforementioned benefit of re-scoring the intermediate search with external models trained on the same dataset. We also show a method for monitoring sub-net performance using oracle intermediates that are void of search errors (§6.1). Finally, we show how these models can adapt to out-of-domain speech translation datasets, how our approach can be generalized to other sequence tasks like speech recognition, and how the benefits of decomposition persist even for larger corpora like MuST-C (§6.2).

2 Background and Motivation

2.1 Compositionality in Sequences Models

The probabilistic space of a sequence is combinatorial in nature, such that a sentence of L words from a fixed vocabulary $\mathcal{V}$ would have an output space $\mathcal{S}$ of size $|\mathcal{V}|^L$. In order to deal with this combinatorial output space, an output sentence is decomposed into labeled target tokens, $\mathbf{y} = (y_1, y_2, \ldots, y_L)$, where $y_l \in \mathcal{V}$.

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{L} P(y_i \mid \mathbf{x}, y_{1:i\text{-}1})$$

An auto-regressive encoder-decoder model uses the above probabilistic decomposition in sequence-to-sequence tasks to learn next word prediction, which outputs a distribution over the next target token y_l given the previous tokens $y_{1:l\text{-}1}$ and the input sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_t, \ldots, \mathbf{x}_T)$, where T is the input sequence length. In the next sub-section we detail the training and inference of these models.

2.2 Auto-regressive Encoder-Decoder Models

Training: In an auto-regressive encoder-decoder model, the ENCODER maps the input sequence $\mathbf{x}$ to a sequence of continuous hidden representations $\mathbf{h}^E = (\mathbf{h}^E_1, \mathbf{h}^E_t, \ldots, \mathbf{h}^E_T)$, where $\mathbf{h}^E_t \in \mathbb{R}^d$. The DECODER then auto-regressively maps $\mathbf{h}^E$ and the preceding ground-truth output tokens, $\hat{y}_{1:l\text{-}1}$, to $\mathbf{h}^D_l$, where $\mathbf{h}^D_l \in \mathbb{R}^d$. The sequence of decoder hidden representations form $\mathbf{h}^D = (\mathbf{h}^D_1, \mathbf{h}^D_l, \ldots, \mathbf{h}^D_L)$ and the likelihood of each output token y_l is given by SOFTMAXOUT, which denotes an affine projection of $\mathbf{h}^D_l$ to $\mathcal{V}$ followed by a softmax function.

$$\begin{aligned} \mathbf{h}^E &= \textsc{Encoder}(\mathbf{x}) \\ \hat{\mathbf{h}}^D_l &= \textsc{Decoder}(\mathbf{h}^E, \hat{y}_{1:l\text{-}1}) \quad (1) \\ P(y_l \mid \hat{y}_{1:l\text{-}1}, \mathbf{h}^E) &= \textsc{SoftmaxOut}(\hat{\mathbf{h}}^D_l) \quad (2) \end{aligned}$$

During training, the DECODER performs token classification for next word prediction by considering only the ground truth sequences for previous tokens $\hat{\mathbf{y}}$. We refer to this $\hat{\mathbf{h}}^D$ as *oracle* decoder representations, which will be discussed later.

Inference: During inference, we can maximize the likelihood of the entire sequence from the output space $\mathcal{S}$ by composing the conditional probabilities of each step for the L tokens in the sequence.

$$\begin{aligned} \mathbf{h}^D_l &= \textsc{Decoder}(\mathbf{h}^E, y_{1:l\text{-}1}) \quad (3) \\ P(y_l \mid \mathbf{x}, y_{1:l\text{-}1}) &= \textsc{SoftmaxOut}(\mathbf{h}^D_l) \\ \tilde{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y} \in \mathcal{S}} \prod_{i=1}^{L} P(y_i \mid \mathbf{x}, y_{1:i\text{-}1}) \quad (4) \end{aligned}$$

This is an intractable search problem and it can be approximated by either greedily choosing argmax at each step or using a search algorithm like beam search to approximate $\tilde{\mathbf{y}}$. Beam search (Reddy, 1988) generates candidates at each step and prunes the search space to a tractable beam size of B most likely sequences. As $B \to \infty$, the beam search result would be equivalent to equation 4.

$$\begin{aligned} \textsc{GreedySearch} &:= \operatorname*{argmax}_{y_l} P(y_l \mid \mathbf{x}, y_{1:l\text{-}1}) \\ \textsc{BeamSearch} &:= \textsc{Beam}(P(y_l \mid \mathbf{x}, y_{1:l\text{-}1})) \end{aligned}$$

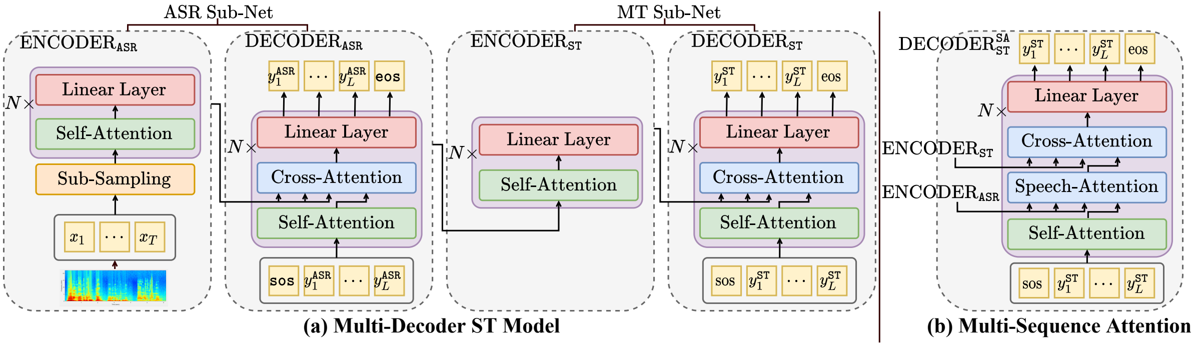

Figure 1: The left side present the schematics and the information flow of our proposed framework applied to ST, in a model we call the Multi-Decoder. Our model decomposes ST into ASR and MT sub-nets, each of which consist of an encoder and decoder. The right side displays a Multi-Sequence Attention variant of the DECODER$_{\text{ST}}$ that is conditioned on both speech information via the ENCODER$_{\text{ASR}}$ and transcription information via the ENCODER$_{\text{ST}}$.

In approximate search for auto-regressive models, like beam search, the DECODER receives alternate candidates of previous tokens to find candidates with a higher likelihood as an overall sequence. This also allows for the use of external models like Language Models (LM) or Connectionist Temporal Classification Models (CTC) for re-scoring candidates (Hori et al., 2017).

3 Proposed Framework

In this section, we present a general framework to exploit natural decompositions in sequence tasks which seek to predict some output $\mathcal{C}$ from an input sequence $\mathcal{A}$. If there is an intermediate sequence $\mathcal{B}$ for which $\mathcal{A} \to \mathcal{B}$ sequence transduction followed by $\mathcal{B} \to \mathcal{C}$ prediction achieves the original task, then the original $\mathcal{A} \to \mathcal{C}$ task is decomposable.

In other words, if we can learn $P(\mathcal{B} \mid \mathcal{A})$ then we can learn the overall task of $P(\mathcal{C} \mid \mathcal{A})$ through $\max_{\mathcal{B}}(P(\mathcal{C} \mid \mathcal{A}, \mathcal{B})P(\mathcal{B} \mid \mathcal{A}))$, approximated using Viterbi search. We define a first encoder-decoder SUB$_{\mathcal{A}\to\mathcal{B}}$NET to map an input sequence $\mathcal{A}$ to a sequence of decoder hidden states, $\mathbf{h}^{D_{\mathcal{B}}}$. Then we define a subsequent SUB$_{\mathcal{B}\to\mathcal{C}}$NET to map $\mathbf{h}^{D_{\mathcal{B}}}$ to the final probabilistic output space of $\mathcal{C}$. Therefore, we call $\mathbf{h}^{D_{\mathcal{B}}}$ *hidden intermediates*. The following equations shows the two sub-networks of our framework, SUB$_{\mathcal{A}\to\mathcal{B}}$NET and SUB$_{\mathcal{B}\to\mathcal{C}}$NET, which can be trained end-to-end while also exploiting compositionality in sequence tasks. [2]

[2]Note that this framework does not use locally-normalized softmax distributions but rather the hidden representations, thereby avoiding label bias issues when combining multiple sub-systems (Bottou et al., 1997; Wiseman and Rush, 2016).

SUB$_{\mathcal{A}\to\mathcal{B}}$NET:

$$
\begin{aligned}
\mathbf{h}^E &= \text{ENCODER}_{\mathcal{A}}(\mathcal{A}) \\
\hat{\mathbf{h}}_l^{D_{\mathcal{B}}} &= \text{DECODER}_{\mathcal{B}}(\mathbf{h}^E, \hat{\mathbf{y}}^{\mathcal{B}}_{1:l-1}) \\
P(y_l^{\mathcal{B}} \mid \hat{\mathbf{y}}^{\mathcal{B}}_{1:l-1}, \mathbf{h}^E) &= \text{SOFTMAXOUT}(\hat{\mathbf{h}}_l^{D_{\mathcal{B}}})
\end{aligned}
\quad (5)
$$

SUB$_{\mathcal{B}\to\mathcal{C}}$NET:

$$
P(\mathcal{C} \mid \hat{\mathbf{h}}_l^{D_{\mathcal{B}}}) = \text{SUB}_{\mathcal{B}\to\mathcal{C}}\text{NET}(\hat{\mathbf{h}}_l^{D_{\mathcal{B}}}) \quad (6)
$$

Note that the final prediction, given by equation 6, does not need to be a sequence and can be a categorical class like in spoken language understanding tasks. Next we will show how the *hidden intermediates* become *searchable* during inference.

3.1 Searchable Hidden Intermediates

As stated in section §2.2, approximate search algorithms maximize the likelihood, $P(\mathbf{y} \mid \mathbf{x})$, of the entire sequence by considering different candidates y_l at each step. Candidate-based search, particularly in auto-regressive encoder-decoder models, also affects the decoder hidden representation, $\mathbf{h}^D$, as these are directly dependent on the previous candidate (refer to equations 1 and 3). This implies that by searching for better approximations of the previous predicted tokens, $\mathbf{y}_{l-1} = (\mathbf{y}_{\text{BEAM}})_{l-1}$, we also improve the decoder hidden representations for the next token, $\mathbf{h}_l^D = (\mathbf{h}^D_{\text{BEAM}})_l$. As $\mathbf{y}_{\text{BEAM}} \to \hat{\mathbf{y}}$, the decoder hidden representations tend to the *oracle* decoder representations that have only errors from next word prediction, $\mathbf{h}^D_{\text{BEAM}} \to \hat{\mathbf{h}}^D$. A perfect search is analogous to choosing the ground truth $\hat{y}$ at each step, which would yield $\hat{\mathbf{h}}^D$.

We apply this beam search of hidden intermediates, thereby approximating $\hat{\mathbf{h}}^{D_{\mathcal{B}}}$ with $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}$. This process is illustrated in algorithm 1, which shows beam search for $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}$ that are subsequently passed to the $\textsc{Sub}_{\mathcal{B}\to\mathcal{C}}\textsc{Net}$.[3] In line 7, we show how an external model like an LM or a CTC model can be used to generate an alternate sequence likelihood, $P_{\text{EXT}}(\mathbf{y}^{\mathcal{B}}_{l})$, which can be combined with the $\textsc{Sub}_{\mathcal{A}\to\mathcal{B}}\textsc{Net}$ likelihood, $P_{\mathcal{B}}(\mathbf{y}^{\mathcal{B}}_{l} \mid \mathbf{x})$, with a tunable parameter λ.

Algorithm 1 Beam Search for Hidden Intermediates: We perform beam search to approximate the most likely sequence for the sub-task $\mathcal{A} \to \mathcal{B}$, $\mathbf{y}^{\mathcal{B}}_{\text{BEAM}}$, while collecting the corresponding $\textsc{Decoder}_{\mathcal{B}}$ hidden representations, $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}$. The output $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}$, is passed to the final sub-network to predict final output $\mathcal{C}$ and $\mathbf{y}^{\mathcal{B}}_{\text{BEAM}}$ is used for monitoring performance on predicting $\mathcal{B}$.

1: **Initialize:** $\text{BEAM} \leftarrow \{\text{sos}\}$; k $\leftarrow$ beam size;
2: $\mathbf{h}^{E_{\text{A}}} \leftarrow \textsc{Encoder}_{\mathcal{A}}(\mathbf{x})$
3: **for** l=1 **to** $\max_{\text{STEPS}}$ **do**
4:   **for** $\mathbf{y}^{\mathcal{B}}_{l\text{-}1} \in \text{BEAM}$ **do**
5:     $\mathbf{h}^{D_{\mathcal{B}}}_{l} \leftarrow \textsc{Decoder}_{\mathcal{B}}(\mathbf{h}^{E_{\mathcal{A}}}, \mathbf{y}^{\mathcal{B}}_{l\text{-}1})$
6:     **for** $\mathbf{y}^{\mathcal{B}}_{l} \in \mathbf{y}^{\mathcal{B}}_{l\text{-}1} + \{\mathcal{V}\}$ **do**
7:       $s_l \leftarrow P_{\mathcal{A}\to\mathcal{B}}(\mathbf{y}^{\mathcal{B}}_{l} \mid \mathbf{x})^{1\text{-}\lambda} P_{\text{EXT}}(\mathbf{y}^{\mathcal{B}}_{l})^{\lambda}$
8:       $\mathcal{H} \leftarrow (s_l, \mathbf{y}^{\mathcal{B}}_{l}, \mathbf{h}^{D_{\mathcal{B}}}_{l})$
9:     **end for**
10:   **end for**
11:   $\text{BEAM} \leftarrow \arg^{\text{k}}\max(\mathcal{H})$
12: **end for**
13: $(s^{\mathcal{B}}, \mathbf{y}^{\mathcal{B}}_{\text{BEAM}}, \mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}) \leftarrow \text{argmax}(\text{BEAM})$
14: **Return** $\mathbf{y}^{\mathcal{B}}_{\text{BEAM}} \to \textsc{Sub}_{\mathcal{A}\to\mathcal{B}}\textsc{Net}$ Monitoring
15: **Return** $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}} \to$ Final $\textsc{Sub}_{\mathcal{B}\to\mathcal{C}}\textsc{Net}$

We can monitor the performance of the $\textsc{Sub}_{\mathcal{A}\to\mathcal{B}}\textsc{Net}$ by comparing the decoded intermediate sequence $\mathbf{y}^{\mathcal{B}}_{\text{BEAM}}$ to the ground truth $\hat{\mathbf{y}}^{\mathcal{B}}$. We can also monitor the $\textsc{Sub}_{\mathcal{B}\to\mathcal{C}}\textsc{Net}$ performance by using the aforementioned *oracle* representations of the intermediates, $\hat{\mathbf{h}}^{D_{\mathcal{B}}}$, which can be obtained by feeding the ground truth $\hat{\mathbf{y}}^{\mathcal{B}}$ to $\textsc{Decoder}_{\mathcal{B}}$. By passing $\hat{\mathbf{h}}^{D_{\mathcal{B}}}$ to $\textsc{Sub}_{\mathcal{B}\to\mathcal{C}}\textsc{Net}$, we can observe its performance in a vacuum, i.e. void of search errors in the hidden intermediates.

3.2 Multi-Decoder Model

In order to show the applicability of our end-to-end framework we propose our Multi-Decoder model for speech translation. This model predicts a sequence of text translations $\mathbf{y}^{\text{ST}}$ from an input sequence of speech $\mathbf{x}$ and uses a sequence of text transcriptions $\mathbf{y}^{\text{ASR}}$ as an intermediate. In this case, the $\textsc{Sub}_{\mathcal{A}\to\mathcal{B}}\textsc{Net}$ in equation 5 is specified as the ASR sub-net and the $\textsc{Sub}_{\mathcal{B}\to\mathcal{C}}\textsc{Net}$ in equation 6 is specified as the MT sub-net. Since the MT sub-net is also a sequence prediction task, both sub-nets are encoder-decoder models in our architecture (Bahdanau et al., 2015; Vaswani et al., 2017). In Figure 1 we illustrate the schematics of our transformer based Multi-Decoder ST model which can also be summarized as follows:

$$\mathbf{h}^{E_{\text{ASR}}} = \textsc{Encoder}_{\text{ASR}}(\mathbf{x}) \tag{7}$$

$$\hat{\mathbf{h}}^{D_{\text{ASR}}}_{l} = \textsc{Decoder}_{\text{ASR}}(\mathbf{h}^{E_{\text{ASR}}}, \hat{y}^{\text{ASR}}_{1:l\text{-}1}) \tag{8}$$

$$\mathbf{h}^{E_{\text{ST}}} = \textsc{Encoder}_{\text{ST}}(\hat{\mathbf{h}}^{D_{\text{ASR}}}) \tag{9}$$

$$\hat{\mathbf{h}}^{D_{\text{ST}}}_{l} = \textsc{Decoder}_{\text{ST}}(\mathbf{h}^{E_{\text{ST}}}, \hat{y}^{\text{ST}}_{1:l\text{-}1}) \tag{10}$$

As we can see from Equations 9 and 10, the MT sub-network attends only to the decoder representations, $\hat{\mathbf{h}}^{D_{\text{ASR}}}$, of the ASR sub-network, which could lead to the error propagation issues from the ASR sub-network to the MT sub-network similar to the cascade systems, as mentioned in §1. To alleviate this problem, we modify equation 10 such that $\textsc{Decoder}_{\text{ST}}$ attends to both $\mathbf{h}^{E_{\text{ST}}}$ and $\mathbf{h}^{E_{\text{ASR}}}$:

$$\hat{\mathbf{h}}^{D^{\text{SA}}_{\text{ST}}}_{l} = \textsc{Decoder}^{\text{SA}}_{\text{ST}}(\mathbf{h}^{E_{\text{ST}}}, \mathbf{h}^{E_{\text{ASR}}}, \hat{y}^{\text{ST}}_{1:l\text{-}1}) \tag{11}$$

We use the multi-sequence cross-attention discussed by Helcl et al. (2018), shown on the right side of Figure 1, to condition the final outputs generated by $\hat{\mathbf{h}}^{D_{\text{ST}}}_{l}$ on both speech and transcript information in an attempt to allow our network to recover from intermediate mistakes during inference. We call this model the Multi-Decoder w/ Speech-Attention.

4 Baseline Encoder-Decoder Model

For our baseline model, we use an end-to-end encoder-decoder (Enc-Dec) ST model with ASR joint training (Inaguma et al., 2020) as an auxiliarly loss to the speech encoder. In other words, the model consumes speech input using the $\textsc{Encoder}_{\text{ASR}}$, to produce $\mathbf{h}^{E_{\text{ASR}}}$, which is used for cross-attention by $\textsc{Decoder}_{\text{ASR}}$ and the $\textsc{Decoder}_{\text{ST}}$. Using the decomposed ASR task as an auxiliary loss also helps the baseline Enc-Dec model and provide strong baseline performance, as we will see in Section 6.

5 Data and Experimental Setup

Data: We demonstrate the efficacy of our proposed approach on ST in the Fisher-CallHome cor-

[3] The algorithm shown only considers a single top approximation of the search; however, with added time-complexity, the final task prediction improves with the n-best $\mathbf{h}^{D_{\mathcal{B}}}_{\text{BEAM}}$ for selecting the best resultant $\mathcal{C}$.

pus (Post et al., 2013) which contains 170 hours of Spanish conversational telephone speech, transcriptions, and English translations. All punctuations except apostrophes were removed and results are reported in terms of detokenized case-insensitive BLEU (Papineni et al., 2002; Post, 2018). We compute BLEU using the 4 references in Fisher (dev, dev2, and test) and the single reference in CallHome (dev and test) (Post et al., 2013; Kumar et al., 2014; Weiss et al., 2017). We use a joint source and target vocabulary of 1K byte pair encoding (BPE) units (Kudo and Richardson, 2018).

We prepare the corpus using the ESPnet library and we follow the standard data preparation, where inputs are globally mean-variance normalized log-mel filterbank and pitch features from up-sampled 16kHz audio (Watanabe et al., 2018). We also apply speed perturbations of 0.9 and 1.1 and the SS SpecAugment policy (Park et al., 2019).

Baseline Configuration: All of our models are implemented using the ESPnet library and trained on 3 NVIDIA Titan 2080Ti GPUs for ≈12 hours. For the Baseline Enc-Dec baseline, discussed in §4, we use an $\text{ENCODER}_{\text{ASR}}$ consisting of a convolutional sub-sampling by a factor of 4 (Watanabe et al., 2018) and 12 transformer encoder blocks with 2048 feed-forward dimension, 256 attention dimension, and 4 attention heads. The $\text{DECODER}_{\text{ASR}}$ and $\text{DECODER}_{\text{ST}}$ both consist of 6 transformer decoder blocks with the same configuration as $\text{ENCODER}_{\text{ASR}}$. There are 37.9M trainable parameters. We apply dropout of 0.1 for all components, detailed in the Appendix (A.1).

We train our models using an effective batch-size of 384 utterances and use the Adam optimizer (Kingma and Ba, 2015) with inverse square root decay learning rate schedule. We set learning rate to 12.5, warmup steps to 25K, and epochs to 50. We use joint training with hybrid CTC/attention ASR (Watanabe et al., 2017) by setting mtl-alpha to 0.3 and asr-weight to 0.5 as defined by Watanabe et al. (2018). During inference, we perform beam search (Seki et al., 2019) on the ST sequences, using a beam size of 10, length penalty of 0.2, max length ratio of 0.3 (Watanabe et al., 2018).

Multi-Decoder Configuration: For the Multi-Decoder ST model, discussed in §3, we use the same transformer configuration as the baseline for the $\text{ENCODER}_{\text{ASR}}$, $\text{DECODER}_{\text{ASR}}$, and $\text{DECODER}_{\text{ST}}$. Additionally, the Multi-Decoder has an $\text{ENCODER}_{\text{ST}}$ consisting of 2 transformer encoder blocks with the same configuration as $\text{ENCODER}_{\text{ASR}}$, giving a total of 40.5M trainable parameters. The training configuration is also the same as for the baseline. For the Multi-Decoder w/ Speech-Attention model (42.1M trainable parameters), we increase the attention dropout of the ST decoder to 0.4 and dropout on all other components of the ST decoder to 0.2 while keeping dropout on the remaining components at 0.1. We verified that increasing the dropout does not help the vanilla multi-decoder ST model.

During inference, we perform beam search on both the ASR and ST output sequences, as discussed in §3. The ST beam search is identical to that of the baseline. For the intermediate ASR beam search, we use a beam size of 16, length penalty of 0.2, max length ratio of 0.3. In some of our experiments, we also include fusion of a source language LM with a 0.2 weight and CTC with a 0.3 weight to re-score the intermediate ASR beam search (Watanabe et al., 2017). For the Speech-Attention variant, we increase LM weight to 0.4.

Note that the ST beam search configuration remains constant across our baseline and Multi-Decoder experiments as our focus is on improving overall performance through searchable intermediate representations. Thus, the various re-scoring techniques applied to the ASR beam search are options newly enabled by our proposed architecture and are not used in the ST beam search.

6 Results

Table 1 presents the overall ST performance (BLEU) of our proposed Multi-Decoder model. Our model improves by +2.9/+0.3 (Fisher/CallHome) over the best cascaded baseline and by +5.6/+1.5 BLEU over the best published end-to-end baselines. With Speech-Attention, our model improves by +3.4/+1.6 BLEU over the cascaded baselines and +7.1/+2.8 BLEU over encoder-decoder baselines. Both the Multi-Decoder and Multi-Decoder w/ Speech-Attention on average are further improved by +0.9/+0.4 BLEU through ASR re-scoring.[4]

Table 1 also includes our implementation of the Baseline Enc-Dec model discussed in §4. In this way, we are able to make a fair comparison with our framework as we control the model and inference

[4]We also evaluate our models using other MT metrics to supplement these results, as shown in the Appendix (A.2).

Model Type	Model Name	Uses Speech Transcripts	Fisher dev(↑)	Fisher dev2(↑)	Fisher test(↑)	CallHome dev(↑)	CallHome test(↑)
Cascade	Inaguma et al. (2020)	✓	41.5	43.5	42.2	**19.6**	**19.8**
Cascade	ESPnet ASR+MT (2018)	✓	**50.4**	**51.2**	**50.7**	**19.6**	19.2
Enc-Dec	Weiss et al. (2017) ◇	✗	46.5	47.3	47.3	16.4	16.6
Enc-Dec	Weiss et al. (2017) ◇	✓	48.3	49.1	48.7	16.8	17.4
Enc-Dec	Inaguma et al. (2020)	✓	46.6	47.6	46.5	16.8	16.8
Enc-Dec	Guo et al. (2021)	✓	48.7	49.6	47.0	18.5	**18.6**
Enc-Dec	Our Implementation	✓	**49.6**	**50.9**	**49.5**	**19.1**	18.2
Multi-Decoder	Our Proposed Model	✓	52.7	53.3	52.6	20.5	20.1
Multi-Decoder	+ASR Re-scoring	✓	53.3	54.2	53.7	21.1	20.8
Multi-Decoder	+Speech-Attention	✓	**54.6**	**54.6**	**54.1**	**21.7**	**21.4**
Multi-Decoder	+ASR Re-scoring	✓	**55.2**	**55.2**	**55.0**	**21.7**	**21.5**

Table 1: Results presenting the overall performance (BLEU) of our proposed multi-decoder model. Cascade and Enc-Dec results from previous papers and our own implementation of the Enc-Dec are shown for comparison. The best performing models are **highlighted**. ◇Implemented with LSTM, while all others are Transformer-based.

Model	Overall ST(↑)	Sub-Net ASR(↓)	Sub-Net MT(↑)
Multi-Decoder	52.7	22.6	64.9
+Speech-Attention	54.6	22.4	66.6

Table 2: Results presenting the overall ST performance (BLEU) of our Multi-Decoder models, along with their sub-net ASR (% WER) and MT (BLEU) performances. All results are from the Fisher dev set.

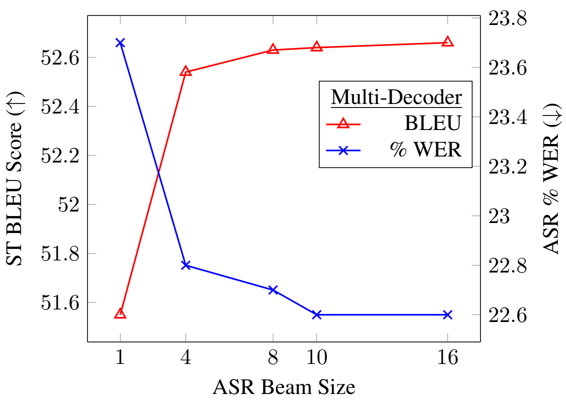

Figure 2: Results studying the effect of the different ASR beam sizes in the intermediate representation search on the overall ST performance (BLEU) and the ASR sub-net performance (% WER) for our multi-decoder model. Beam of 1 is same as greedy search.

configurations to be analagous. For instance, we keep the same search parameters for the final output in the baseline and the Multi-Decoder to demonstrate impact of the intermediate beam search.

6.1 Benefits

6.1.1 Sub-network performance monitoring

An added benefit of our proposed approach over the Baseline Enc-Dec is the ability to monitor the individual performances of the ASR (% WER) and MT (BLEU) sub-nets as shown in Table 2. The Multi-Decoder w/ Speech-Attention shows a greater MT sub-net performance than the Multi-Decoder as well as a slight improvement of the ASR sub-net, suggesting that ST can potentially help ASR.

6.1.2 Beam search for better intermediates

The overall ST performance improves when a higher beam size is used in the intermediate ASR search, and this increase can be attributed to the improved ASR sub-net performance. Figure 1 shows this trend across ASR beam sizes of 1, 4, 8, 10, 16 while fixing the ST decoding beam size to 10. A beam size of 1, which is a greedy search, results in lower ASR sub-net and overall ST performances. As beam sizes become larger, gains taper off as can be seen between beam sizes of 10 and 16.

6.1.3 External models for better search

External models like CTC acoustic models and language models are commonly used for re-scoring encoder-decoder models (Hori et al., 2017), due to the difference in their modeling capabilities. CTC directly models transcripts while being conditionally independent on the other outputs given the input, and LMs predict the next token in a sequence.

Both variants of the Multi-Decoder improve due to improved ASR sub-net performance using exter-

Model	Overall ST(↑)	Sub-Net ASR(↓)
Multi-Decoder	52.7	22.6
+ASR Re-scoring w/ LM	53.2	22.6
+ASR Re-scoring w/ CTC	52.8	22.1
+ASR Re-scoring w/ LM	**53.3**	**21.7**
Multi-Decoder w/ Speech-Attn.	54.6	22.4
+ASR Re-scoring w/ LM	55.1	22.4
+ASR Re-scoring w/ CTC	54.7	22.0
+ASR Re-scoring w/ LM	**55.2**	**21.9**

Table 3: Results presenting the overall ST performance (BLEU) and the sub-net ASR (% WER) of our Multi-Decoder models with external CTC and LM re-scoring in the ASR intermediate representation search. All results are from the Fisher dev set.

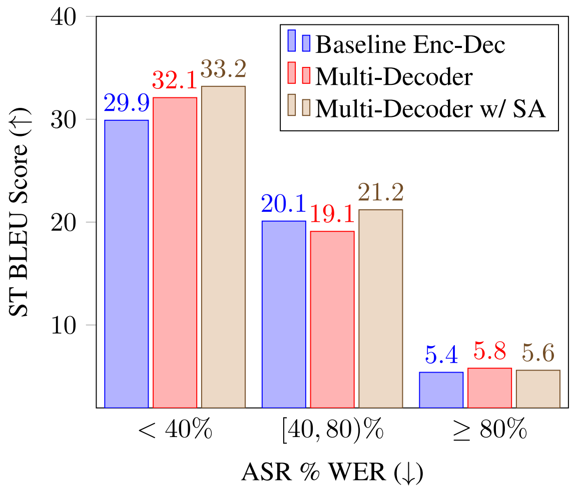

Figure 3: Results comparing the ST performances (BLEU) of our Baseline Enc-Dec, Multi-Decoder, and Multi-Decoder w/ Speech-Attention across different ASR difficulties measured using % WER on the Fisher dev set (1-ref). The buckets on the x-axis are determined using the utterance level % WER using the Multi-Decoder ASR sub-net performance.

nal CTC and LM models for re-scoring, as shown in Table 3. We use a recurrent neural network LM trained on the Fisher-CallHome Spanish transcripts with a dev perplexity of 18.8 and the CTC model from joint loss applied during training. Neither external model incorporates additional data. Although the impact of the LM-only re-scoring is not shown in the ASR % WER, it reduces substitution and deletion rates in the ASR and this is observed to help the overall ST performance.

6.1.4 Error propagation avoidance

As discussed in §3, our Multi-Decoder model inherits the error propagation issue as can be seen in Figure 3. For the easiest bucket of utterances with $< 40\%$ WER in Multi-Decoder's ASR sub-net, our model's ST performance, as measured by the corpus BLEU of the bucket, exceeds that of the Baseline Enc-Dec. The inverse is true for the more difficult bucket of $[40, 80)\%$, showing that error propagation is limiting the performance of our model; however, we show that multi-sequence attention can alleviate this issue. For extremely difficult utterances in the $\geq 80\%$ bucket, ST performance for all three approaches is suppressed. We also provide qualitative examples of error propagation avoidance in the Appendix (A.3).

6.2 Generalizability

In this section, we discuss the generalizability of our framework towards out-of-domain data. We also extend our Multi-Decoder model to other sequence tasks like speech recognition. Finally, we apply our ST models to a larger corpus with more language pairs and a different domain of speech.

6.2.1 Robustness through Decomposition

Like cascaded systems, searchable intermediates provide our model adaptability in individual subsystems towards out-of-domain data using external in-domain language model, thereby giving access to more in-domain data. Specifically for speech translation systems, this means we can use in-domain language models in both source and target languages. We test the robustness of our Multi-Decoder model trained on Fisher-CallHome conversational speech dataset on read speech CoVost-2 dataset (Wang et al., 2020b). In Table 4 we show that re-scoring the ASR sub-net with an in-domain LM improves ASR with around 10.0% lower WER, improving the overall ST performance by around +2.5 BLEU. Compared to an in-domain ST baseline (Wang et al., 2020a), our out-of-domain Multi-Decoder with in-domain ASR re-scoring demonstrates the robustness of our approach.

6.2.2 Decomposing Speech Transcripts

We apply our generic framework to another decomposable sequence task, speech recognition, and show the results of various levels of decomposition in Table 5. We show that with phoneme, character, or byte-pair encoding (BPE) sequences as intermediates, the Multi-Decoder presents strong results on both Fisher and CallHome test sets. We also observe that the BPE intermediates perform bet-

Model	Overall ST(↑)	Sub-Net ASR(↓)
IN-DOMAIN ST MODEL		
Baseline (Wang et al., 2020b)	12.0	-
+ASR Pretrain (Wang et al., 2020b) ◇	23.0	16.0
OUT-OF-DOMAIN ST MODEL		
Multi-Decoder	11.8	46.8
+ASR Re-scoring w/ in-domain LM	14.4	**36.7**
Multi-Decoder w/ Speech-Attention	12.6	46.5
+ASR Re-scoring w/ in-domain LM	**15.0**	**36.7**

Table 4: Results presenting the overall ST performance (BLEU) and the sub-net ASR (% WER) of our Multi-Decoder models when tested on out-of-domain data. All models were trained on the Fisher-CallHome Es→En corpus and tested on CoVost2 Es→En corpus. ◇Pretrained with 364 hours of in-domain ASR data.

Model	En→De ST(↑)	En→Fr ST(↑)
NeurST (Zhao et al., 2020)	22.9	33.3
Fairseq S2T (Wang et al., 2020a)	22.7	32.9
ESPnet-ST (Inaguma et al., 2020)	22.9	32.7
Dual-Decoder (Le et al., 2020)	23.6	33.5
Multi-Decoder w/ Speech-Attn.	26.3	37.0
+ASR Re-scoring	**26.4**	**37.4**

Table 6: Results presenting the overall ST performance (BLEU) of our Multi-Decoder w/ Speech-Attention models with ASR re-scoring across two language-pairs, English-German (En→De) and English-French (En→Fr). All results are from the MuST-C tst-COMMON sets. All models use speech transcripts.

Model	Intermediate	Fisher ASR(↓)	CallHome ASR(↓)
Enc-Dec ◇	-	23.2	45.3
Multi-Decoder	Phoneme	20.7	40.0
Multi-Decoder	Character	20.4	39.9
Multi-Decoder	BPE100	**19.7**	**38.9**

Table 5: Results presenting the % WER ASR performance when using the Multi-Decoder model on decomposed ASR task with phoneme, character, and BPE100 as intermediates. All results are from the Fisher-CallHome Spanish corpus. ◇(Weiss et al., 2017)

ter than phoneme/character variants, which could be attributed to the reduced search capabilities of encoder-decoder models using beam search on longer sequences (Sountsov and Sarawagi, 2016) like in phoneme/character sequences.

6.2.3 Extending to MuST-C Language Pairs

In addition to our results using the 170 hours of the Spanish-English Fisher-CallHome corpus, in Table 6 we show that our decompositional framework is also effective on larger ST corpora. In particular, we use 400 hours of English-German and 500 hours of English-French ST from the MuST-C corpus (Di Gangi et al., 2019). Our Multi-Decoder model improves by +2.7 and +1.5 BLEU, in German and French respectively, over end-to-end baselines from prior works that do not use additional training data. We show that ASR re-scoring gives an additional +0.1 and +0.4 BLEU improvement. [5]

By extending our Multi-Decoder models to this MuST-C study, we show the generalizability of our approach across several dimensions of ST tasks. First, our approach consistently improves over baselines across multiple language-pairs. Second, our approach is robust to the distinct domains of telephone conversations from Fisher-CallHome and the TED-Talks from MuST-C. Finally, by scaling from 170 hours of Fisher-CallHome data to 500 hours of MuST-C data, we show that the benefits of decomposing sequence tasks with searchable hidden intermediates persist even with more data.

Furthermore, the performance of our Multi-Decoder models trained with only English-German or English-French ST data from MuST-C is comparable to other methods which incorporate larger external ASR and MT data in various ways. For instance, Zheng et al. (2021) use 4700 hours of ASR data and 2M sentences of MT data for pretraining and multi-task learning. Similarly, Bahar et al. (2021) use 2300 hours of ASR data and 27M sentences of MT data for pretraining. Our competitive performance without the use of any additional data highlights the data-efficient nature of our proposed end-to-end framework as opposed to the baseline encoder-decoder model, as pointed out by Sperber and Paulik (2020).

7 Discussion and Relation to Prior Work

Compositionality: A number of recent works have constructed composable neural network modules for tasks such as visual question answering (Andreas et al., 2016), neural MT (Raunak et al., 2019), and synthetic sequence-to-sequence tasks (Lake, 2019). Modules that are first trained separately can subsequently be tightly integrated into a single end-to-end trainable model by passing differentiable soft decisions instead of discrete decisions

[5]Details of the MuST-C data preparation and model parameters are detailed in Appendix (A.4).

in the intermediate stage (Bahar et al., 2021). Further, even a single encoder-decoder model can be decomposed into modular components where the encoder and decoder modules have explicit functions (Dalmia et al., 2019).

Joint Training with Sub-Tasks: End-to-end sequence models been shown to benefit from introducing joint training with sub-tasks as auxiliary loss functions for a variety of tasks like ASR (Kim et al., 2017), ST (Salesky et al., 2019; Liu et al., 2020a; Dong et al., 2020; Le et al., 2020), SLU (Haghani et al., 2018). They have been shown to induce structure (Belinkov et al., 2020) and improve the model performance (Toshniwal et al., 2017), but this joint training may reduce data efficiency if some sub-nets are not included in the final end-to-end model (Sperber et al., 2019; Wang et al., 2020c). Our framework avoids this sub-net waste at the cost of computational load during inference.

Speech Translation Decoders: Prior works have used ASR/MT decoding to improve the overall ST decoding through synchronous decoding (Liu et al., 2020a), dual decoding (Le et al., 2020), and successive decoding (Dong et al., 2020). These works partially or fully decode ASR transcripts and use discrete intermediates to assist MT decoding. Tu et al. (2017) and Anastasopoulos and Chiang (2018) are closest to our multi-decoder ST model, however the benefits of our proposed framework are not entirely explored in these works.

Two-Pass Decoding: Two-pass decoding involves first predicting with one decoder and then re-evaluating with another decoder (Geng et al., 2018; Sainath et al., 2019; Hu et al., 2020; Rijhwani et al., 2020). The two decoders iterate on the same sequence, so there is no decomposition into sub-tasks in this method. On the other hand, our approach provides the subsequent decoder with a more structured representation than the input by decomposing the complexity of the overall task. Like two-pass decoding, our approach provides a sense of the future to the second decoder which allows it to correct mistakes from the previous first decoder.

Auto-Regressive Decoding: As auto-regressive decoders inherently learn a language model along with the task at hand, they tend to be domain specific (Samarakoon et al., 2018; Müller et al., 2020). This can cause generalizability issues during inference (Murray and Chiang, 2018; Yang et al., 2018), impacting the performance of both the task at hand and any downstream tasks. Our approach alleviates these problems through intermediate search, external models for intermediate re-scoring, and multi-sequence attention.

8 Conclusion and Future Work

We present searchable hidden intermediates for end-to-end models of decomposable sequence tasks. We show the efficacy of our Multi-Decoder model on the Fisher-CallHome Es→En and MuST-C En→De and En→Fr speech translation corpora, achieving state-of-the-art results. We present various benefits in our framework, including sub-net performance monitoring, beam search for better hidden intermediates, external models for better search, and error propagation avoidance. Further, we demonstrate the flexibility of our framework towards out-of-domain tasks with the ability to adapt our sequence model at intermediate stages of decomposition. Finally, we show generalizability by training Multi-Decoder models for the speech recognition task at various levels of decomposition.

We hope insights derived from our study stimulate research on tighter integrations between the benefits of cascaded and end-to-end sequence models. Exploiting searchable intermediates through beam search is just the tip of the iceberg for search algorithms, as numerous approximate search techniques like diverse beam search (Vijayakumar et al., 2018) and best-first beam search (Meister et al., 2020) have been recently proposed to improve diversity and approximation of the most-likely sequence. Incorporating differentiable lattice based search (Hannun et al., 2020) can also allow the subsequent sub-net to digest n-best representations.

9 Acknowledgements

This work started while Vikas Raunak was a student at CMU, he is now working as a Research Scientist at Microsoft. We thank Pengcheng Guo, Hirofumi Inaguma, Elizabeth Salesky, Maria Ryskina, Marta Méndez Simón and Vijay Viswanathan for their helpful discussion during the course of this project. We also thank the anonymous reviewers for their valuable feedback. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014), which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system (Nystrom et al., 2015), which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). The work was supported in part by an AWS Machine Learning Research Award. This research was also supported in part the DARPA KAIROS program from the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

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.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 39–48. IEEE Computer Society.

Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. 2021. Tight integrated end-to-end training for cascaded speech translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 950–957. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Beck, Trevor Cohn, and Gholamreza Haffari. 2019. Neural speech translation using lattice transformations and graph networks. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 26–31, Hong Kong. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52.

Léon Bottou, Yoshua Bengio, and Yann Le Cun. 1997. Global training of document processing systems using graph transformer networks. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489–494. IEEE.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. In *Privacy in Machine Learning and Artificial Intelligence workshop, ICML*.

Siddharth Dalmia, Abdelrahman Mohamed, Mike Lewis, Florian Metze, and Luke Zettlemoyer. 2019. Enforcing encoder-decoder modularity in sequence-to-sequence models. *arXiv preprint arXiv:1911.03782*.

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*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2020. SDST: Successive decoding for speech-to-text translation. *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, et al. 2021. Recent developments on ESPnet toolkit boosted by conformer. In *2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE.

Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. 2018. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE.

Awni Hannun, Vineel Pratap, Jacob Kahn, and Wei-Ning Hsu. 2020. Differentiable weighted finite-state transducers. *arXiv preprint arXiv:2010.01003*.

Jindřich Helcl, Jindřich Libovický, and Dušan Variš. 2018. CUNI system for the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 616–623, Belgium, Brussels. Association for Computational Linguistics.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. 2017. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In *Proc. Interspeech 2017*, pages 949–953.

Ke Hu, Tara N Sainath, Ruoming Pang, and Rohit Prabhavalkar. 2020. Deliberation model based two-pass end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7799–7803. IEEE.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic. Association for Computational Linguistics.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311. Association for Computational Linguistics.

Patricia Johnson. 1992. Cohesion and coherence in compositions in Malay and English. *RELC Journal*, 23(2):1–17.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

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.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, pages 224–227.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Gaurav Kumar, Matt Post, Daniel Povey, and Sanjeev Khudanpur. 2014. Some insights from translating conversational telephone speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pages 3231–3235. IEEE.

Shankar Kumar, Yonggang Deng, and William Byrne. 2006. A weighted finite state transducer translation template model for statistical machine translation. *Natural Language Engineering*, 12(1):35–76.

Chih-Hua Kuo. 1995. Cohesion and coherence in academic writing: From lexical choice to organization. *RELC Journal*, 26(1):47–62.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, pages 9791–9801.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 2879–2888. PMLR.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. *Proceedings of the 28th International Conference on Computational Linguistics*.

Alexander H. Levis, Neville Moray, and Baosheng Hu. 1994. Task decomposition and allocation problems and discrete event systems. *Automatica*, 30(2):203 – 216.

Yuchen Liu, Jiajun Zhang, Hao Xiong, Long Zhou, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2020a. Synchronous speech recognition and speech-to-text translation with interactive decoding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8417–8424.

Zhiyuan Liu, Yankai Lin, and Maosong Sun. 2020b. *Compositional Semantics*, pages 43–57. Springer Singapore, Singapore.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809.

Bernd T Meyer, Sri Harish Mallidi, Angel Mario Castro Martinez, Guillermo Payá-Vayá, Hendrik Kayser, and Hynek Hermansky. 2016. Performance monitoring for automatic speech recognition in noisy multi-channel environments. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 50–56.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium.

Nicholas A. Nystrom, Michael J. Levine, Ralph Z. Roskies, and J. Ray Scott. 2015. Bridges: A uniquely flexible HPC resource for new communities and data analytics. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. Association for Computing Machinery.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, pages 295–302.

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.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617.

Vijayaditya Peddinti, Guoguo Chen, Vimal Manohar, Tom Ko, Daniel Povey, and Sanjeev Khudanpur. 2015. JHU ASpIRE system: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMS. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546.

N. Pham, Thai-Son Nguyen, Thanh-Le Ha, J. Hussain, Felix Schneider, J. Niehues, Sebastian Stüker, and A. Waibel. 2019. The IWSLT 2019 KIT speech translation system. In *International Workshop on Spoken Language Translation (IWSLT)*.

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, Belgium, Brussels. Association for Computational Linguistics.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In *International Workshop on Spoken Language Translation (IWSLT 2013)*.

Vikas Raunak, Vaibhav Kumar, and Florian Metze. 2019. On compositionality in neural machine translation. *NeurIPS Workshop, Context and Compositionality in Biological and Artificial Neural Systems*.

Raj Reddy. 1988. Foundations and grand challenges of artificial intelligence: AAAI presidential address. *AI Mag.*, 9(4):9–21.

Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. 2020. OCR post correction for endangered language texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942, Online. Association for Computational Linguistics.

Tara N Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, et al. 2019. Two-pass end-to-end speech recognition. *Proc. Interspeech 2019*, pages 2773–2777.

Elizabeth Salesky, Matthias Sperber, and Alan W Black. 2019. Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Florence, Italy. Association for Computational Linguistics.

Lahiru Samarakoon, Brian Mak, and Albert YS Lam. 2018. Domain adaptation of end-to-end speech recognition in low-resource settings. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 382–388. IEEE.

Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Niko Moritz, and Jonathan Le Roux. 2019. Vectorized beam search for CTC-attention-based speech recognition. In *Proc. Interspeech 2019*, pages 3825–3829.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Pavel Sountsov and Sunita Sarawagi. 2016. Length bias in encoder decoder models and a case for global conditioning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1525, Austin, Texas. Association for Computational Linguistics.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2019. Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*, 7:313–325.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics*, 29(1):97–133.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. In *Proc. Interspeech 2017*, pages 3532–3536.

John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D Peterson, et al. 2014. XSEDE: accelerating scientific discovery. *Computing in science & engineering*, 16(5):62–74.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017*, pages 3097–3103. AAAI Press.

Evelyne Tzoukermann and Corey Miller. 2018. Evaluating automatic speech recognition in translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 294–302, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 7371–7379. AAAI Press.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. 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 (AACL): System Demonstrations*, pages 33–39. Association for Computational Linguistics.

Changhan Wang, Anne Wu, and Juan Pino. 2020b. CoVoST 2: A massively multilingual speech-to-text translation corpus. *arXiv preprint arXiv:2007.10310*.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020c. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9161–9168.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. Interspeech 2018*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proc. Interspeech 2017*, pages 2625–2629.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Chengqi Zhao, Mingxuan Wang, and Lei Li. 2020. NeurST: Neural speech translation toolkit. *arXiv preprint arXiv:2012.10018*.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. *arXiv preprint arXiv:2102.05766*.

A Appendix

A.1 Training and Inference hyperparameters

We tune training and inference hyperparameters using only the dev sets. We first determined the best hyperparameters for our baseline Enc-Dec implementation and fixed all settings not pertaining to the unique searchable hidden intermediates of our Multi-Decoder. Then, we find the best hyperparameters for our proposed models under these constraints to demonstrate a true comparison against the baseline. For our Speech-Attention variant, we found that increasing attention dropout in the ST sub-net decoder to 0.4 improved performance, which we verified was not true for the vanilla Multi-Decoder model. For our external model re-scoring, we found that a CTC weight of 0.3 is best for all Multi-Decoder and Multi-Decoder w/ Speech-Attention. The best LM weight for the Multi-Decoder was 0.2, while the best LM weight for the Multi-Decoder w/ Speech-Attention was 0.4. For both of these re-scoring hyperparameters, we tried $[0.2, 0.3, 0.4]$. For deciding the beam size, we use the experiment demonstrated in Figure 2 which uses beam sizes of $[1, 4, 8, 10, 16]$.

A.2 Multi-Decoder ST Performance across other automatic MT Metrics

To supplement our overall ST results on the Fisher/CallHome corpus in Table 1, which shows BLEU scores, we also evaluated the same Multi-Decoder and Baseline Enc-Dec (Our Implementation) models on two additional metrics: METEOR (Banerjee and Lavie, 2005) and Translation Edit Rate (TER) (Snover et al., 2006). Performance across all three metrics show consistent trends, with the Multi-Decoder outperforming the Baseline Enc-Dec model on all metrics. We see that both the Multi-Decoder and Multi-Decoder w/ Speech-Attention models are improved through ASR Re-scoring. Further, the models with Speech-Attention perform better than those without.

A.3 Qualitative Examples of Error Propagation Avoidance

To supplement our qualitative analysis of the error propagation avoidance of the Multi-Decoder with Speech-Attention model in §6.1.4, we also show four qualitative examples in Table 7. In the first three examples, the Multi-Decoder and Multi-Decoder with Speech-Attention models both make the same mistakes in the ASR portion of Spanish-English translation, but the model with Speech-Attention recovers by producing correct English translations despite mistakes in the Spanish transcription. On the other hand, the model without Speech-Attention propagates the Spanish transcription errors into English translation errors. In the fourth example only the Multi-Decoder w/ Speech-Attention makes a mistake in Spanish transcription, but the English translation still recovers.

A.4 MuST-C Data Setup and Model Details

Data: We extend our approach to other language pairs from the MuST-C speech translation corpus (Di Gangi et al., 2019). These are recordings of TED talks in English with translations in various target languages. In our experiments we show results on two language pairs, namely, English-German and English-French. We use the provided dev set for deciding the training and inference hyperparameters, as mentioned in Appendix (A.1). We report detokenized case-sensitive BLEU (Post, 2018) on the tst-COMMON set. We apply the same text processing as done in (Inaguma et al., 2020) and use a joint source and target vocabulary of 8K byte pair encoding (BPE) units (Kudo and Richardson, 2018). Similar to §5, we use the ESPnet library to prepare the corpus, and apply the same data preparation and augmentations.

Multi-Decoder Configuration: For the MuST-C experiments, we scaled our Multi-Decoder w/ Speech-Attention config from the Fisher-CallHome experiments by increasing the $\text{ENCODER}_{\text{ST}}$ to contain 4 transformer encoder blocks. We increased the attention dim and attention heads of the $\text{ENCODER}_{\text{ASR}}$ and $\text{DECODER}_{\text{ASR}}$ to 512 dimension and 8 heads respectively, while only increasing the attention dimension to 512 for $\text{ENCODER}_{\text{ST}}$ and $\text{DECODER}_{\text{ST}}$. This increased the total trainable parameters to 135M, which we trained on 4 NVIDIA V-100 GPUs for $\approx$3 days. We also found that increasing the attention dropout of ASR decoder to 0.2 helped with the increased parameters. We kept the remaining dropout parameters the same as our previous experiments. We also keep the remaining training configurations the same like the effective batch-size, learning rate and warmup steps, loss weighting and SpecAugment policy.

During inference, we use the same beam sizes from our Fisher-CallHome experiments and we perform a search across the length penalty and max length ratio settings using the MuST-C dev sets.

Model / Source	ASR Output	ST Output
Ground-Truth	… porque tengo a mis dos hijos acá	… because i have my two children here
Multi-Decoder	… porque tengo mis dos hijos acá	… because i have two kids here
+Speech-Attention	… porque tengo mis dos hijos acá	… because i have my two children here
Ground-Truth	puedes ayudar para que se haga justicia más rápido	you can help so that justice is served quickly
Multi-Decoder	puedes ayudar para que sea justicia más rápido	you can help so it's faster
+Speech-Attention	puedes ayudar para que sea justicia más rápido	you can help so that it's faster justice
Ground-Truth	pero tiene muchas cosas muy bonitas	but there are many beautiful things
Multi-Decoder	pero tienen muchas cosas muy bonitas	but they have a lot of nice things
+Speech-Attention	pero tienen muchas cosas muy bonitas	but there are many very beautiful things
Ground-Truth	acampar ir a pescar y ir a las montañas a esquiar	camping and fishing and going to the mountains to ski
Multi-Decoder	acampar y a pescar y y de las montañas esquiar	camping and fishing and and the mountains skiing
+Speech-Attention	a campar y ir a pescar y ir a las montañas a esquiar	camping and go fishing and go to the mountains to ski

Table 7: Examples where the Multi-Decoder and Multi-Decoder w/ Speech-Attention models make errors in the ASR portion of Spanish-English ST. In these cases the Speech-Attention component alleviates ASR error propagation, producing correct translations despite mistakes in transcription. Words that are transcribed/translated correctly are highlighted in green and those that are incorrect are in pink.

	Fisher test			CallHome test		
Model	BLEU (↑)	METEOR(↑)	TER(↓)	BLEU (↑)	METEOR(↑)	TER(↓)
Baseline Enc-Dec	49.5	37.9	42.7	18.2	22.9	68.7
Multi-Decoder	52.6	39.7	40.5	20.1	24.6	66.5
+ASR Re-scoring	53.7	40.0	39.6	20.8	24.9	65.3
+Speech-Attention	54.1	40.2	39.2	21.4	25.2	65.3
+ASR Re-scoring	**55.0**	**40.4**	**38.5**	**21.5**	**25.4**	**64.2**

Table 8: Results presenting the performance of our Baseline Enc-Dec implementation and our Multi-Decoder models as evaluated by three metrics: BLEU, METEOR, and Translation Edit Rate (TER). These are the same models as in Table 1, which uses BLEU. All results are from the Fisher-CallHome Spanish-English test corpus.

In the intermediate ASR beam search we use a length penalty of 0.1 and 0.2 for English-German and English-French respectively. In the ST beam search we use a max length ratio of 0.3 and length penalties of 0.6 and 0.5 for English-German and English-French respectively. For our experiments with ASR re-scoring, we use a LM weight of 0.1 and a CTC weight of 0.1. In these re-scoring experiments we also set the ASR length penalty to 0.6 and the ST length penalty to 0.5, while increasing the ST max length ratio to 0.5. The LMs used were trained on the English transcripts of the MuST-C English-German and English-French corpora, with dev perplexities of 32.7 and 23.2 respectively.