# Worse WER, but Better BLEU? Leveraging Word Embedding as Intermediate in Multitask End-to-End Speech Translation

Shun-Po Chuang[1], Tzu-Wei Sung[2], Alexander H. Liu[1], and Hung-yi Lee[1]

[1]National Taiwan University, Taiwan
[2]University of California San Diego, USA
{f04942141, b03902042, r07922013, hungyilee}@ntu.edu.tw



## Abstract

Speech translation (ST) aims to learn transformations from speech in the source language to the text in the target language. Previous works show that multitask learning improves the ST performance, in which the recognition decoder generates the text of the source language, and the translation decoder obtains the final translations based on the output of the recognition decoder. Because whether the output of the recognition decoder has the correct semantics is more critical than its accuracy, we propose to improve the multitask ST model by utilizing word embedding as the intermediate.

## 1 Introduction

Speech translation (ST) increasingly receives attention from the machine translation (MT) community recently. To learn the transformation between speech in the source language and the text in the target language, conventional models pipeline automatic speech recognition (ASR) and text-to-text MT model (Bérard et al., 2016). However, such pipeline systems suffer from error propagation.

Previous works show that deep end-to-end models can outperform conventional pipeline systems with sufficient training data (Weiss et al., 2017; Inaguma et al., 2019; Sperber et al., 2019). Nevertheless, well-annotated bilingual data is expensive and hard to collect (Bansal et al., 2018a,b; Duong et al., 2016). Multitask learning plays an essential role in leveraging a large amount of monolingual data to improve representation in ST. Multitask ST models have two jointly learned decoding parts, namely the recognition and translation part. The recognition part firstly decodes the speech of source language into the text of source language, and then based on the output of the recognition part, the translation part generates the text in the target language. Variant multitask models have been explored (Anastasopoulos and Chiang, 2018), which shows the improvement in low-resource scenario.

Although applying the text of source language as the intermediate information in multitask end-to-end ST empirically yielded improvement, we argue whether this is the optimal solution. Even though the recognition part does not correctly transcribe the input speech into text, the final translation result would be correct if the output of the recognition part preserves sufficient semantic information for translation. Therefore, we explore to leverage word embedding as the intermediate level instead of text.

In this paper, we apply pre-trained word embedding as the intermediate level in the multitask ST model. We propose to constrain the hidden states of the decoder of the recognition part to be close to the pre-trained word embedding. Prior works on word embedding regression show improved results on MT (Jauregi Unanue et al., 2019; Kumar and Tsvetkov, 2018). Experimental results show that the proposed approach obtains improvement to the ST model. Further analysis also shows that constrained hidden states are approximately isospectral to word embedding space, indicating that the decoder achieves speech-to-semantic mappings.

## 2 Multitask End-to-End ST model

Our method is based on the multitask learning for ST (Anastasopoulos and Chiang, 2018), including speech recognition in the source language and translation in the target language, as shown in Fig. 1(a). The input audio feature sequence is first encoded into the encoder hidden state sequence $h = h_1, h_2, \ldots, h_T$ with length $T$ by the pyramid encoder (Chan et al., 2015). To present speech recognition in the source language, the attention mechanism and a decoder is employed to produce source decoder sequence $\hat{s} = \hat{s}_1, \hat{s}_2, \ldots, \hat{s}_M$, where $M$ is the number of decoding steps in the source language. For each decoding step $m$, the probability $P(\hat{y}_m)$ of predicting the token $\hat{y}_m$ in the source language vocabulary can be computed based on the corresponding decoder state $s_m$.

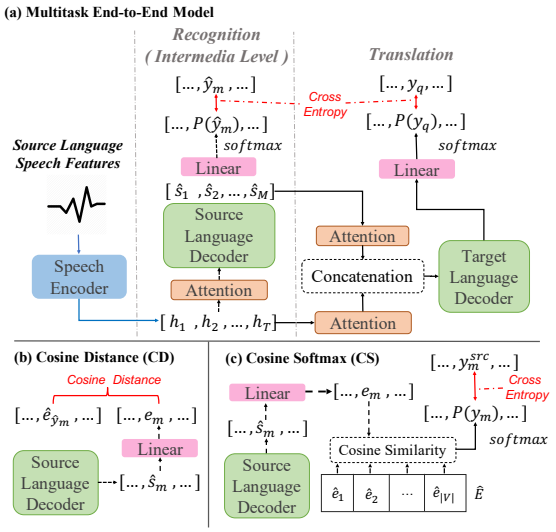


Figure 1: (a) Multitask ST model. Dotted arrows indicate steps in the recognition part. Solid arrows indicate steps in the translation part. (b) Directly learn word embedding via cosine distance. (c) Learn word embedding via cosine softmax function. Both (b)(c) are the recognition part in (a).

To perform speech translation in the target language, both the source language decoder state sequence $\hat{s}$ and the encoder state sequence $h$ will be attended and treated as the target language decoder's input. The hidden state of target language decoder can then be used to derived the probability $P(y_q)$ of predicting token $y_q$ in the target language vocabulary for every decoding step $q$.

Given the ground truth sequence in the source language $\hat{y} = \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_M$ and the target language $y = y_1, y_2, \ldots, y_Q$ with length $Q$, multitask ST can be trained with maximizing log likelihood in *both* domains. Formally, the objective function of multitask ST can be written as:

$$\begin{aligned}\mathcal{L}_{\text{ST}} &= \frac{\alpha}{M}\mathcal{L}_{\text{src}} + \frac{\beta}{Q}\mathcal{L}_{\text{tgt}} \\ &= \frac{\alpha}{M}\sum_{m} -\log P(\hat{y}_m) + \frac{\beta}{Q}\sum_{q} -\log P(y_q),\end{aligned} \tag{1}$$

where $\alpha$ and $\beta$ are the trade-off factors to balance between the two tasks.

## 3 Proposed Methods

We propose two ways to help the multitask end-to-end ST model capture the semantic relation between word tokens by leveraging the source language word embedding as intermediate level. $\hat{E} = \{\hat{e}_1, \hat{e}_2, \ldots \hat{e}_{|V|}\}$, where $V$ is the vocabulary set and $e_v \in \mathcal{R}^D$ is the embedding vector with dimension $D$ for any word $v \in V$, in the recognition task. We choose the source language decoder state (embedding) $\hat{s}$ to reinforce since it is later used in the translation task. To be more specific, we argue that the embedding generated by the source language decoder should be more *semantically* correct in order to benefit the translation task. Given the pre-trained source language word embedding $\hat{E}$, we proposed to constrain the source decoder state $\hat{s}_m$ at step $m$ to be close to its corresponding word embedding $\hat{e}_{\hat{y}_m}$ with the two approaches detailed in the following sections.

### 3.1 Directly Learn Word Embedding

Since semantic-related words would be close in terms of cosine distance (Mikolov et al., 2018), a simple idea is to minimize the cosine distance (CD) between the source language decoder hidden state $\hat{s}_m$ and the corresponding word embedding $\hat{e}_{\hat{y}_m}$ for every decode step $m$,

$$\begin{aligned}\mathcal{L}_{\text{CD}} &= \sum_{m} 1 - \cos(f_\theta(\hat{s}_m), \hat{e}_{\hat{y}_m}) \\ &= \sum_{m} 1 - \frac{f_\theta(\hat{s}_m) \cdot \hat{e}_{\hat{y}_m}}{\|f_\theta(\hat{s}_m)\| \|\hat{e}_{\hat{y}_m}\|},\end{aligned} \tag{2}$$

where $f_\theta(\cdot)$ is a learnable linear projection to match the dimensionality of word embedding and decoder state. With this design, the network architecture of the target language decoder would not be limited by the dimension of word embedding. Fig. 1(b) illustrates this approach. By replacing $\mathcal{L}_{\text{src}}$ in Eq. (1) with $\mathcal{L}_{\text{CD}}$, semantic learning from word embedding for source language recognition can be achieved.

### 3.2 Learn Word Embedding via Probability

Ideally, using word embedding as the learning target via minimizing CD can effectively train the decoder to model the semantic relation existing in the embedding space. However, such an approach suffers from the hubness problem (Faruqui et al., 2016) of word embedding in practice (as we later discuss in Sec. 4.5).

To address this problem, we introduce cosine softmax (CS) function (Liu et al., 2017a,b) to learn speech-to-semantic embedding mappings. Given the decoder hidden state $\hat{e}_m$ and the word embedding $\hat{E}$, the probability of the target word $\hat{y}_m$ is defined as

$$P_{\text{CS}}(\hat{y}_m) = \frac{\exp(\cos(f_\theta(\hat{s}_m), \hat{e}_{\hat{y}_m})/\tau)}{\sum_{\hat{e}_v \in \hat{E}} \exp(\cos(f_\theta(\hat{s}_m), \hat{e}_v)/\tau)}, \tag{3}$$

where $\cos(\cdot)$ and $f_\theta(\cdot)$ are from Eq. (2), and $\tau$ is the temperature of softmax function. Note that since the temperature $\tau$ re-scales cosine similarity, the hubness problem can be mitigated by selecting a proper value for $\tau$. Fig. 1(c) illustrates the approach. With the probability derived from cosine softmax in Eq. (3), the objective function for source language decoder can be written as

$$\mathcal{L}_{\text{CS}} = \sum_m -\log P_{\text{CS}}(\hat{y}_m). \quad (4)$$

By replacing $\mathcal{L}_{\text{src}}$ in Eq. (1) with $\mathcal{L}_{\text{CS}}$, the decoder hidden state sequence $\hat{s}$ is forced to contain semantic information provided by the word embedding.

## 4 Experiments

### 4.1 Experimental Setup

We used Fisher Spanish corpus (Graff et al., 2010) to perform Spanish speech to English text translation. And we followed previous works (Inaguma et al., 2019) for pre-processing steps, and 40/160 hours of *train* set, standard *dev-test* are used for the experiments. Byte-pair-encoding (BPE) (Kudo and Richardson, 2018) was applied to the target transcriptions to form 10K subwords as the target of the translation part. Spanish word embeddings were obtained from FastText pre-trained on Wikipedia (Bojanowski et al., 2016), and 8000 Spanish words were used in the recognition part.

The encoder is a 3-layer 512-dimensional bidirectional LSTM with additional convolution layers, yielding 8× down-sampling in time. The decoders are 1024-dimensional LSTM, and we used one layer in the recognition part and two layers in the translation part. The models were optimized using Adadelta with $10^{-6}$ as the weight decay rate. Scheduled sampling with probability 0.8 was applied to the decoder in the translation part. Experiments ran 1.5M steps, and models were selected by the highest BLEU on four transcriptions per speech in *dev* set.

### 4.2 Speech Translation Evaluation

**Baseline**: We firstly built the single-task end-to-end model (**SE**) to set a baseline for multitask learning, which resulted in 34.5/34.51 BLEU on *dev* and *test* set respectively, which showed comparable results to Salesky et al. (2019). Multitask end-to-end model (**ME**) mentioned in Sec. 2 is another baseline. By applying multitask learning in addition, we could see that **ME** outperforms **SE** in all conditions.

| | (a) 160 hours | | (b) 40 hours | |
|---|---|---|---|---|
| | *dev* | *test* | *dev* | *test* |
| **SE** | 34.50 | 34.51 | 17.41 | 15.44 |
| **ME** | 35.35 | 35.49 | 23.30 | 20.40 |
| **CD** | 33.06 | 33.65 | 23.53 | 20.87 |
| **CS** | **35.84** | **36.32** | **23.54** | **21.72** |

Table 1: BLEU scores trained on different size of data.

**High-resource**: Column (a) in Table 1 showed the results trained on 160 hours of data. **CD** and **CS** represent the proposed methods mentioned in Sec. 3.1 and 3.2 respectively. We got mixed results on further applying pre-trained word embedding on **ME**. **CD** degraded the performance, which is even worse than **SE**, but **CS** performed the best. Results showed that directly learn word embedding via cosine distance is not a good strategy in the high-resource setting, but integrating similarity with cosine softmax function can significantly improve performance. We leave the discussion in Sec. 4.5.

**Low-resource**: We also experimented on 40 hours subset data for training, as shown in column (b) in Table 1. We could see that **ME**, **CD** and **CS** overwhelmed **SE** in low-resource setting. Although **CD** resulted in degrading performance in high-resource setting, it showed improvements in low-resource scenario. **CS** consistently outperformed **ME** and **CD** on different data size, showing it is robust on improving ST task.

### 4.3 Analysis of Recognition Decoder Output

In this section, we analyzed hidden states $\mathbf{s}$ by existing methods. For each word $v$ in corpus, we denoted its word embedding $\hat{e}_v$ as *pre-trained embedding*, and $e_v$ as *predicted embedding*. Note that because a single word $v$ could be mapped by multiple audio segments, we took the average of all its predicted embedding. We obtained the top 500 frequent words in the whole Fisher Spanish corpus, and tested on the sentences containing only these words in *test* set.

**Eigenvector Similarity**: To verify our proposed methods can constrain hidden states in the word embedding space, we computed eigenvector similarity between *predicted embedding* and *pre-trained embedding* space. The metric derives from Laplacian eigenvalues and represents how similar be-

| | 160 hours | | 40 hours | |
|---|---|---|---|---|
| | *dev* | *test* | *dev* | *test* |
| **ME** | 16.50 | 18.58 | 13.80 | 15.09 |
| **CD** | **2.60** | **3.44** | **3.95** | **3.63** |
| **CS** | 11.55 | 13.76 | 8.62 | 9.80 |

Table 2: Eigenvector similarity.

| | 160 hours | | 40 hours | |
|---|---|---|---|---|
| | P@1 | P@5 | P@1 | P@5 |
| **ME** | 1.85 | 6.29 | 1.11 | 9.62 |
| **CD** | **61.48** | **77.40** | **56.30** | **69.25** |
| **CS** | 17.78 | 35.19 | 10.37 | 25.19 |

Table 3: Precision@k of semantic alignment on *test* set.

tween two spaces, the *lower* value on the metric, the *more* approximately isospectral between the two spaces. Previous works showed that the metric is correlated to the performance of translation task (Søgaard et al., 2018; Chung et al., 2019). As shown in Table 2, *predicted embedding* is more similar to *pre-trained embedding* when models trained on sufficient data (160 v.s 40 hours). **CD** is the most similar case among the three cases, and **ME** is the most different case. Results indicated that our proposals constrain hidden states in *pre-trained embedding* space.

**Semantic Alignment**: To further verify if *predicted embedding* is semantically aligned to *pre-trained embedding*, we applied Procrustes alignment (Conneau et al., 2017; Lample et al., 2017) method to learn the mapping between *predicted embedding* and *pre-trained embedding*. Top 50 frequent words were selected to be the training dictionary, and we evaluated on the remaining 450 words with cross-domain similarity local scaling (CSLS) method. Precision@k (P@k, k=1,5) were reported as measurements. As shown in Table 3, **CD** performed the best, and **ME** was the worst one. This experiment reinforced that our proposals can constrain hidden states to the similar structure of word embedding space.

### 4.4 Speech Recognition Evaluation

We further analyzed the results of speech recognition for **ME** and **CS**. To obtain the recognition results from Eq (3), simply take $\arg\max_v P_{CS}(v)$. The word error rate (WER) of the source language recognition was reported in Table 4. Combining the results shown in Table 1, we could see that **CS** has worse WER, but higher BLEU compared with **ME**. We concluded that although leveraging word embedding at the intermediate level instead of text results in worse performance in speech recognition (this indicates that the WER of the recognition part does not fully determine the translation performance), the semantic information could somewhat help multitask models generate better translation in terms of BLEU. We do not include the WER of CD in Table 1 because its WER is poor (>100%), but interestingly, the BLEU of CD is still reasonable, which is another evidence that WER of the intermediate level is not the key of translation performance.

| | 160 hours | | 40 hours | |
|---|---|---|---|---|
| | *dev* | *test* | *dev* | *test* |
| **ME** | 43.13 | 38.57 | 53.42 | 54.70 |
| **CS** | 50.15 | 44.43 | 57.63 | 57.21 |

Table 4: Word error rate (%) trained on different size of data.

### 4.5 Cosine Distance (CD) v.s. Softmax (CS)

Based on experimental results, we found that proposals are possible to map speech to semantic space. With optimizing **CS**, BLEU consistently outperformed **ME**, which shows that utilizing semantic information truly helps on ST. Directly minimizing cosine distance made the *predicted embedding* space closest to *pre-trained embedding* space, but performed inconsistently on BLEU in different data sizes. We inferred that the imbalance word frequency training and hubness problem (Faruqui et al., 2016) in word embedding space made hidden states not discriminated enough for the target language decoder while optimizing **CS** can alleviate this issue.

## 5 Conclusions

Our proposals showed that utilizing word embedding as intermediate helps with the ST task, and it is possible to map speech to the semantic space. We also observed that lower WER in source language recognition not imply higher BLEU in target language translation.

This work is the first attempt to utilize word embedding in the ST task, and further techniques can be applied upon this idea. For example, cross-lingual word embedding mapping methods can be considered within the ST model to shorten the distance between MT and ST tasks.

## 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.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018a. Low-resource speech-to-text translation. *arXiv preprint arXiv:1803.09164*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018b. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. *arXiv preprint arXiv:1809.01431*.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. 2015. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. 2019. Towards unsupervised speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7170–7174. IEEE.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 949–959.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

David Graff, Shudong Huang, Ingrid Cartagena, Kevin Walker, , and Christopher Cieri. 2010. Fisher spanish speech (ldc2010s01). *https://catalog.ldc.upenn.edu/LDC2010S01*.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation. *arXiv preprint arXiv:1910.00254*.

Inigo Jauregi Unanue, Ehsan Zare Borzeshi, Nazanin Esmaili, and Massimo Piccardi. 2019. ReWE: Regressing word embeddings for regularization of neural machine translation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 430–436, Minneapolis, Minnesota. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Sachin Kumar and Yulia Tsvetkov. 2018. Von mises-fisher loss for training sequence to sequence models with continuous outputs. *arXiv preprint arXiv:1812.04616*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Yu Liu, Hongyang Li, and Xiaogang Wang. 2017a. Learning deep features via congenerous cosine loss for person recognition. *arXiv preprint arXiv:1702.06890*.

Yu Liu, Hongyang Li, and Xiaogang Wang. 2017b. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Elizabeth Salesky, Matthias Sperber, and Alan W Black. 2019. Exploring phoneme-level speech representations for end-to-end speech translation. *arXiv preprint arXiv:1906.01199*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. *arXiv preprint arXiv:1805.03620*.

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.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

# 6 Appendix

## 6.1 Single-task end-to-end model

One of our baseline models is a single-task end-to-end model, which is abbreviated as **SE** in the previous section. **SE** was trained using the source language speech and the target language text. It shares the same architecture with the multitask model but without the source language text decoding (without the recognition part in Fig. 1(a)). And its objective function can be written as:

$$\mathcal{L}_{\text{SE}} = \mathcal{L}_{\text{tgt}} = \sum_{q} -\log P(y_q). \tag{5}$$

Further details can be referred to (Anastasopoulos and Chiang, 2018).

## 6.2 Using different Word Embeddings

Our proposed model benefits from publicly available pre-trained word embedding, which is easy-to-obtain yet probably coming from the domains different from testing data. It can bring to ST models in a simple plug-in manner.

In Sec. 4.2, we used word embedding trained on Wikipedia. To demonstrate the improvement of using different word embeddings, we additionally provide results of ST models using word embeddings trained on Fisher Spanish corpus (*train* and *dev* set) in Table 5. Here we use the abbreviation of word embedding trained on Wikipedia as W-emb and word embedding trained on Fisher Spanish corpus as F-emb.

In **CD/CS** method, using F-emb obtained 0.27/0.61 improvement from using W-emb on *dev* set. And, **CD** got 0.15 improvement but **CS** got 0.51 degrading performance on *test* set.

The improvements show that using word embeddings trained in the related domain helps on the performance. In **CD** method, although using F-emb improves the performance, it still under-performed **ME** method. It indicates that the selection of adopting methods is critical. In **CS** method, it got a great improvement on *dev* set but not on *test* set. It shows that using F-emb does help with the performance, but using word embedding trained on rich data (W-emb) could provide additional information that can generally extend to the *test* set.

| | Word Embedding Source | 160 hours | |
|---|---|---|---|
| | | *dev* | *test* |
| **ME** | - | 35.35 | 35.49 |
| **CD** | Wikipedia | 33.06 | 33.65 |
| | Fisher Spanish | 33.33 | 33.80 |
| **CS** | Wikipedia | 35.84 | 36.32 |
| | Fisher Spanish | 36.45 | 35.81 |

Table 5: BLEU scores on using different pre-trained word embeddings.

In general, whether using F-emb or W-emb as the training target, the experimental results show consistency to the discussion in Sec. 4.2.