Improving End-to-End Bangla Speech Recognition with Semi-supervised Training

Nafis Sadeq^{*2}, **Nafis Tahmid Chowdhury**^{*1}, **Farhan Tanvir Utshaw** [1], **Shafayat Ahmed**[1], and **Muhammad Abdullah Adnan**[1]

[1]Bangladesh University of Engineering and Technology (BUET)
[2]Samsung R&D Institute Bangladesh
{1205008.ns,1505077.ntc,1505105.ftu,1205115.sa}@ugrad.cse.buet.ac.bd
adnan@cse.buet.ac.bd

Abstract

Automatic speech recognition systems usually require large annotated speech corpus for training. The manual annotation of a large corpus is very difficult. It can be very helpful to use unsupervised and semi-supervised learning methods in addition to supervised learning. In this work, we focus on using a semi-supervised training approach for Bangla Speech Recognition that can exploit large unpaired audio and text data. We encode speech and text data in an intermediate domain and propose a novel loss function based on the global encoding distance between encoded data to guide the semi-supervised training. Our proposed method reduces the Word Error Rate (WER) of the system from 37% to 31.9%.

1 Introduction

[1] An annotated speech corpus is an essential component for the development of an automatic speech recognition system (ASR). Speech corpus is a collection of audio files with corresponding text transcriptions. Manually developing a speech corpus of required size is a time consuming and monotonous task. It also requires some prerequisites like a recording environment, clear utterance, and additional information such as gender of speakers, etc. For achieving a large vocabulary continuous speech recognition we need approximately several hundred to few thousands of hours of speech corpus. Semi-supervised training can be a useful solution to tackle the hurdles related to speech corpus development. Semi-supervised training can provide us a way to exploit a huge collection of publicly available text as well as audio resources to improve the performance of an ASR.

In this work, we focus on improving an end-to-end speech recognition system for Bangladeshi Bangla using semi-supervised training. There are very few publicly available large speech corpora for Bangladeshi Bangla. Google released 229 hours of speech corpus for Bangladeshi Bangla (Kjartansson et al., 2018). But there are huge amounts of publicly available news audio files, audiobooks, recordings in Youtube and other media sources. There are a lot of text sources too like news websites, blogs, e-books, etc. Considering the abundance of unpaired audio and text data for Bangla language, a semi-supervised training method that can exploit both unpaired audio and text is very useful. Proper use of the unpaired data along with existing paired speech corpus can boost the performance of the Bangla ASR system.

Different researchers have tried different ways of incorporating this unlabelled, unannotated data for speech recognition. Our approach is similar to the approach used by Karita et al. (2018). We utilize an intermediate representation of speech and text data using a shared encoder network for semi-supervised training of the ASR system. Our contributions in this work are as follows:

- We propose a novel inter-domain loss function based on global encoding distance (GED loss) of speech and text data.
- Our proposed Global Encoding Distance (GED) loss for inter-domain features performs better than both the Gaussian KL-divergence loss proposed in Karita et al. (2018) and Maximum Mean Discrepancy (MMD) loss proposed in Karita et al. (2019). Our loss function is more meaningful and intuitive in the context of unpaired audio and text data. The performance of the GED loss is more robust to minibatch size compared to Gaussian KL-divergence and MMD loss.
- To our best knowledge, this is the first work

[1]* authors contributed equally

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1875–1883
November 16 - 20, 2020.

on Bangla language that incorporates semi-supervised training into deep learning based end-to-end ASR architecture.

- Using our semi-supervised training, we are able to exploit 1000 hours of unpaired audio data and 800K unpaired Bangla sentences. Our experiments show that our semi-supervised training with GED loss achieves WER of 31.9%, outperforming both the baseline end-to-end system with an external language model and semi-supervised method with MMD loss.

The paper is organized in the following manner. We discuss the related works in section 2, the system architecture in section 3, details of our inter-domain loss in section 4, corpus description in section 5, experimental results in section 6, and conclusion in section 7.

2 Related Works

Researchers have explored different methods of semi-supervised training for speech recognition. Long et al. (2019) investigate large-scale semi-supervised training to improve acoustic models for automatic speech recognition. They provide an empirical analysis of semi-supervised training with respect to transcription quality, data quality, filtering, etc. Fan et al. (2019) pre-train the encoder-decoder network with unpaired speech and text. They use a large amount of unpaired audio to pre-train the encoder and synthesized audio from the unpaired text to pre-train the decoder. Drugman et al. (2019), Yu et al. (2010) integrate active learning jointly with semi-supervised training in speech recognition system. Thomas et al. (2013) use transcribed multilingual data and semi-supervised training to circumvent the lack of sufficient training data for acoustic modeling. They train deep neural networks as data-driven feature front ends.

Veselỳ et al. (2013) use utterance-level and frame-level confidences for data selection during self-training. They find it beneficial to reduce the disproportion in amounts of paired and unpaired data by including the paired data several times in semi-supervised training. Liu and Kirchhoff (2014) describe the combination of deep neural networks and graph-based semi-supervised learning for acoustic modeling in speech recognition. Dhaka and Salvi (2017) use a sparse auto-encoder to take advantage of both unlabelled and labeled data simultaneously through mini-batch stochastic gradient descent.

Guo et al. (2018) try to improve the performance of a code-switching speech recognition system for Mandarin-English using semi-supervised training. They apply semi-supervised learning for lexicon learning as well as acoustic modeling. Similarly, Veselỳ et al. (2018) & Lileikytė et al. (2016) use untranscribed data for Luxembourgish & Lithuanian ASR respectively. Šmídl et al. (2018) use a two-step training method to generalize the air traffic control speech recognizer. First, a baseline speech recognition system is trained using a paired speech corpus and it is used to transcribe publicly available unlabeled data. The transcribed data is then filtered based on confidence scores and is used to retrain the acoustic model.

Recently, semi-supervised training has been proposed in the context of end-to-end ASR. Karita et al. (2018) propose a shared encoder architecture for speech and text inputs that can encode both data from their respective domain to a common intermediate domain. They combine speech-to-text and text-to-text mapping by using the shared network to improve speech-to-text mapping. They propose an inter-domain loss function based on Gaussian KL-divergence which represents the dissimilarity between the encoded features of speech and text data. They later proposed an inter-domain loss function based on Maximum Mean Discrepancy (Karita et al., 2019). In both cases, they assume that the encoded speech features in the current minibatch are sampled from one distribution and encoded text features in the current minibatch are sampled from a second distribution. The inter-domain loss is calculated based on the discrepancy of these two distributions. This approach has some weaknesses. The performance of this system varies based on the chosen minibatch size. Moreover, this approach does not take into account the variance of the current encoded features in the global context. We solve both problems by introducing a new inter-domain loss function based on global encoding distance.

3 Our System

In this section, we describe our baseline end-to-end architecture as well as semi-supervised architecture.

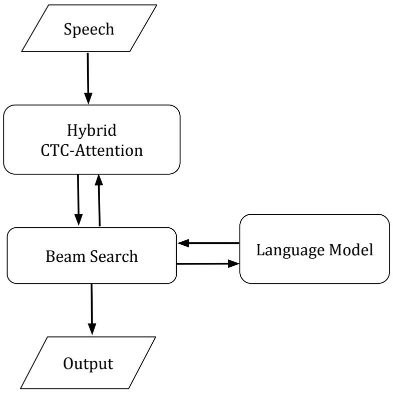

Figure 1: Baseline System

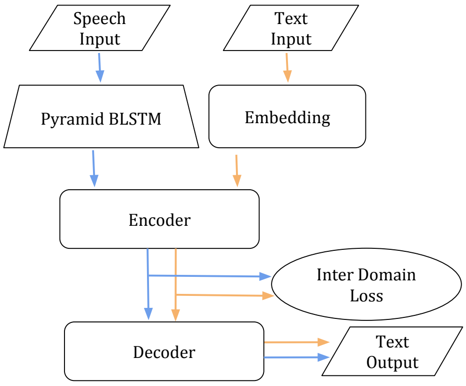

Figure 2: Semi-Supervised System

3.1 Baseline System

Our baseline system is an end-to-end ASR system based on the work of Watanabe et al. (2017). The architecture is shown in Figure 1. CTC and attention networks are combined in this architecture. Both networks share an encoder network. The shared encoder network has 6 layers of Bidirectional Long Short Term Memory (BLSTM) units. Each layer has 320 BLSTM units. A linear projection layer is connected to each BLSTM layer. The linear projection layer consists of 320 units. The decoder has 1 layer of unidirectional LSTM units. The number of LSTM units in this layer is 300. The scores from the attention network and the CTC network are combined during decoding. Let $p(c_t)$ be the probability of output label c_t at position t, given previous output labels and let w be the CTC weight.

$$\log p(c_t) = w \log p^{ctc}(c_t) + (1 - w) \log p^{att}(c_t) \tag{1}$$

As for the audio feature, we use 40 Mel-frequency cepstral coefficients (MFCC) per audio frame. We also consider their first and second-order temporal derivatives. So, we have 120 speech features per audio frame. These features are fed to the shared encoder and the attention decoder generates the character sequence.

We use a Recurrent Neural Network (RNN), based language model, in shallow fusion (Hori et al., 2017) with the baseline end-to-end architecture. We use both character level and word level RNN in our experiments. The character level RNN has 2 layers of LSTM, with each layer having 650 LSTM units. The word-level RNN has 1 hidden layer and this layer has 1000 LSTM units. For the word level RNN, we use most frequently used 65000 Bangla words as our vocabulary set.

3.2 Semi-supervised System

Our semi-supervised end-to-end speech recognition system for Bangla is based on the work of Karita et al. (2018). The semi-supervised architecture is shown in Figure 2. We use a shared encoder that encodes speech and text input sequences into a common intermediate domain. Speech feature sequences and text character sequences are very different in length. Also, speech features are continuous-valued vectors while text characters are discrete. We use a pyramid BLSTM network that performs sub-sampling on the speech feature sequence. The sub-sampling process shortens the length of the speech feature sequence. We use an embedding layer that converts the text character ids to continuous domain vectors. Thus, the speech and the text inputs become compatible with each other and they are both passed through a shared encoder containing BLSTM units.

Our encoder network has 6 layers of BLSTM cells. The size of each layer is 320 units. The decoder network has 1 layer of LSTM cells. The size of this layer is 300 units. First, this architecture is trained in a supervised manner using the paired speech corpus. Then, we perform retraining using both paired and unpaired corpus. We use 3 different loss to guide semi-supervised retraining. They are the following:

Speech-to-text loss This is a conventional speech-to-text loss during supervised learning, which consists of a negative log-likelihood of the ground-truth text given by the encoded speech features. This loss is the combination of CTC and attention loss similar to the baseline system. We denote this loss as L_{sup}. The calculation of speech-to-text loss is shown in Equation 2. We use CTC weight w_1 to control the relative importance of CTC and attention loss.

Text-to-text auto-encoder loss This is the negative log-likelihood that the encoder-decoder architecture can reconstruct the output text from an unpaired text corpus. We denote this loss as L_{ae}

Inter-domain loss This is the dissimilarity between distributions of the encoded speech features and the encoded text features. We use global encoding distance as a measurement for our inter-domain loss. We denote this loss as L_{id}. More on this is described in section 4.

$$L_{sup} = w_1 L_{ctc} + (1 - w_1) L_{att} \tag{2}$$

$$L_{uns} = w_2 L_{id} + (1 - w_2) L_{ae} \tag{3}$$

$$L_{tot} = w_3 L_{sup} + (1 - w_3) L_{uns} \tag{4}$$

Equation 3 shows how the text auto-encoder loss and the inter-domain loss are combined to generate the unsupervised loss. We use speech text ratio parameter w_2 to control the relative importance of the text auto-encoder loss and the inter-domain loss. Then both the supervised loss L_{sup} and the unsupervised loss L_{uns} are combined to calculate the total loss L_{tot} (shown in Equation 4). Here, w_3 is the supervised loss ratio which controls the relative importance between the supervised and the unsupervised loss.

4 The Inter-Domain Loss

In this section, we describe our proposed inter-domain loss function.

4.1 Encoding Procedure

First, we pre-process the speech and text data in a way that they become compatible with each other. We reduce the length of the speech data by performing sub-sampling with a pyramid BLSTM unit. We also transform the text sequences into a continuous domain vector with an embedding layer. The pre-processed speech and text data are then absorbed by an encoder unit. The output of the encoder unit is considered as the inter-domain representation of the speech and text data. The overview of the encoding process is shown in Figure 3. Figure 4 shows the visualization of the encoded data using t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008).

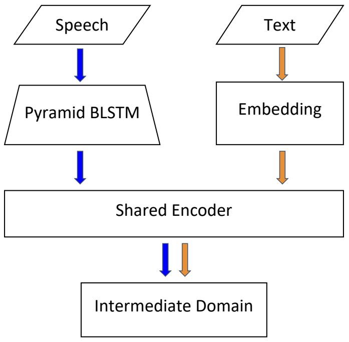

Figure 3: Overview of Encoding

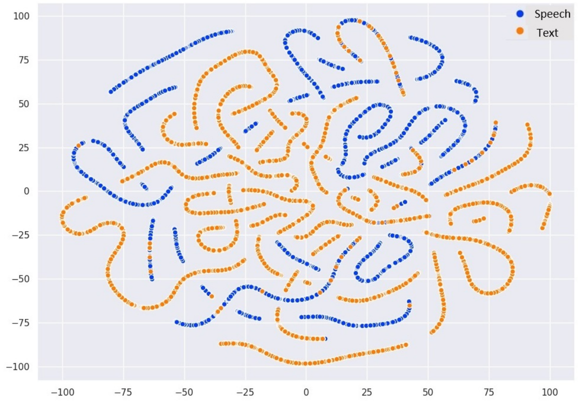

Figure 4: t-SNE Visualization of Encoded Data

4.2 Maximum Mean Discrepancy Loss

Here, we describe the MMD loss proposed by Karita et al. (2019) and some of its limitations. A minibatch is formed by sampling the encoded features from unpaired speech and text data. All encoded speech features in this minibatch are considered to be from one underlying distribution. Similarly, all encoded text features from this minibatch are considered to be from another underlying distribution. Then Maximum Mean Discrepancy between these two distributions is calculated. A similar MMD calculation is repeated for the paired minibatch. Then the inter-domain loss is calculated by combining the MMD loss from the paired and the unpaired set, as shown in Algorithm 1.

This approach has some limitations because the distribution assumption is made only considering the unpaired data in the current minibatch. This loss calculation lacks the knowledge about global distribution, density, and variance of the unpaired data. Also, assuming a distribution based on the current minibatch makes the system unstable with respect to changing batch size. In other words, the system is not guaranteed to converge to the optimal

Algorithm 1 Computation of the MMD loss

```
$N \leftarrow$ Number of samples
$D \leftarrow$ dimension of encoded vector
$H_{SP} \leftarrow$ encoded speech, paired minibatch
$H_{TP} \leftarrow$ encoded text, paired minibatch
$H_{SU} \leftarrow$ encoded speech, unpaired minibatch
$H_{TU} \leftarrow$ encoded text, unpaired minibatch
$H_{SP} \in \mathbb{R}^{N_{sp} \times D}, H_{TP} \in \mathbb{R}^{N_{tp} \times D}$
$H_{SU} \in \mathbb{R}^{N_{su} \times D}, H_{TU} \in \mathbb{R}^{N_{tu} \times D}$
function LOSS($H_{SP}, H_{TP}, H_{SU}, H_{TU}$)
    $lp = MMD(H_{SP}, H_{TP})$
    $lu = MMD(H_{SU}, H_{TU})$
    return $lp + lu$
function MMD($H_S, H_T$)
    $m_s = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \sum_{k=1}^{D} H_{i,k}^S H_{j,k}^S$
    $m_t = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \sum_{k=1}^{D} H_{i,k}^T H_{j,k}^T$
    $k_s = \frac{\sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \exp(\sum_{k=1}^{D} H_{i,k}^S H_{j,k}^S - m_s)}{N_s^2}$
    $k_t = \frac{\sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \exp(\sum_{k=1}^{D} H_{i,k}^T H_{j,k}^T - m_t)}{N_t^2}$
    $k_{s,t} = \frac{\sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \exp(\sum_{k=1}^{D} H_{i,k}^S H_{j,k}^T - \frac{m_s}{2} - \frac{m_t}{2})}{N_s N_t}$
    return $k_s + k_t - 2k_{s,t}$
```

solution for all minibatch sizes.

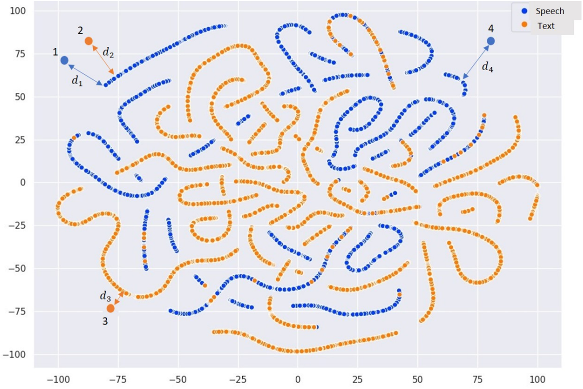

Figure 5: GED Loss

Algorithm 2 Computation of the GED loss

```
$N \leftarrow$ Number of samples
$D \leftarrow$ dimension of encoded vector
$H_{SP} \leftarrow$ encoded speech, paired minibatch
$H_{TP} \leftarrow$ encoded text, paired minibatch
$H_{SU} \leftarrow$ encoded speech, unpaired minibatch
$H_{TU} \leftarrow$ encoded text, unpaired minibatch
$H_{SP} \in \mathbb{R}^{N_{sp} \times D}, H_{TP} \in \mathbb{R}^{N_{tp} \times D}$
$H_{SU} \in \mathbb{R}^{N_{su} \times D}, H_{TU} \in \mathbb{R}^{N_{tu} \times D}$
function LOSS($H_{SP}, H_{TP}, H_{SU}, H_{TU}$)
    $l_{sp} = \sum_{i=1}^{N_{sp}} GED(H_{SP}^i|X)$
    $l_{tp} = \sum_{i=1}^{N_{tp}} GED(H_{TP}^i|X)$
    $l_{su} = \sum_{i=1}^{N_{su}} GED(H_{SU}^i|X)$
    $l_{tu} = \sum_{i=1}^{N_{tu}} GED(H_{TU}^i|X)$
    return $\frac{l_{sp}+l_{tp}+l_{su}+l_{tu}}{N_{sp}+N_{tp}+N_{su}+N_{tu}}$
```

4.3 Global Encoding Distance (GED) Loss

We have found that a significant performance gain can be made by exploiting the global distribution and variance of the encoded unpaired data. We pre-calculate the encoding for our entire unpaired dataset and generate a representative matrix X for our unpaired set. X is calculated as follows. A set of neighboring points are repeatedly sampled from the encoded unpaired data. A representative mean is calculated for these neighboring points. X is the concatenation of all such neighboring means. Here, $X \in \mathbb{R}^{N_x \times D}$ where N_x is the number of representative means and D is the dimension of an encoded feature. The representative mean is used to reduce the size of the matrix X. This matrix X now functions as a global representing matrix for the unpaired set. Now the global encoding distance for an encoded vector v^i with respect to X is defined as follows:

$$d_i = GED(v^i|X) = \min_{j=1}^{N_x} \|e^i - v^i\| \qquad (5)$$

Here, e^i is the i_{th} row of the matrix X ($e^i \in \mathbb{R}^{1 \times D}$) and it represents the i_{th} representing mean of the unpaired set.The global encoding distance for four sample points is shown in Figure 5. For each point, the global encoding distance is the distance from this point to the closest representing mean in matrix X. The pseudocode for calculating inter-domain loss based on global encoding distance is shown in Algorithm 2.

Unlike MMD loss, our proposed loss function captures the dissimilarity between the encoded speech and text features with respect to the global representing matrix X. In addition to capturing the dissimilarity between the data in current minibatch, GED based loss also captures the variance of the encoded data in the global context. This system is less likely to suffer from any potential shortsightedness introduced by the assumption based on a few samples within a minibatch. Also, our system is more likely to converge to the optimal solution for any given minibatch size.

5 Corpus Description

In this section, we describe the corpus used for our experiments.

5.1 Paired Speech Corpus

We use the corpus provided by Kjartansson et al. (2018) as our paired speech corpus. This corpus has around 229 hours of annotated speech data. The total number of utterances is around 217000 and the number of speakers is 505.

5.2 Unpaired Audio Data

The news recordings from a lot of Bangladeshi TV channels are available in the public domain. We mostly use these public domain news recordings as our audio source. After crawling the data, we split the audio files based on silence. We use 0.5 seconds as minimum silence duration and 0.0001 (between 0.0 and 1.0) as silence energy threshold. After silence based segmentation, we discard all audio files shorter than 3 seconds and longer than 9 seconds. Encoding audio files in the intermediate domain becomes easier when all audio files have a similar duration. After this, we have 1000 hours worth of unpaired audio corpus.

5.3 Unpaired Text Data

We use Bangla newspaper websites for preparing unpaired text corpus. We crawl around 40 Bangla websites. We use text cleaning on the collected data to remove non-Bangla symbols, punctuation, special characters, etc. We then perform text normalization. We convert all numbers to their textual form, elaborate abbreviations, convert dates, etc. We apply the same text normalization on the text transcription of the paired dataset to maintain homogeneity among paired and unpaired corpus. After text cleaning and normalization, we discard all Bangla sentences that have fewer than 4 or greater than 10 words. Our text corpus has around 800K Bangla sentences.

Parameter	**Value**
Initialization	Uniform Distr
Encoder layers	6
Encoder layer size	320 (BLSTM)
Encoder projection layer size	320
Decoder layers	1
Decoder layer size	300 (LSTM)
Learning Rate	0.5
Batch size	24
CTC weight	0.3
Speech text ratio	0.1
Supervised loss ratio	0.9

Table 1: Hyper-parameter Description

6 Evaluations

In this section, we describe the experimental results.

6.1 Test Set

We separate 2000 utterances from the Google speech corpus as our test set. The test set has 5 speakers and covers various domains.

6.2 Training Details

At first, we train the CTC-attention network with the paired speech corpus. It takes around 10 hours in our setup. Then we retrain the model using the unpaired speech and text corpus along with the paired corpus. It takes around 20 hours. All experiments are performed on a hardware with a Core i7 processor, 16 GB Memory, NVIDIA GeForce GTX 1070 GPU. The important hyper-parameters of our system are shown in Table 1.

The training graph for the initial supervised training is shown in Figure 6. In this step, the system learns to minimize the CTC and the attention loss, effectively minimizing the supervised speech to text loss. The training graph for the retraining stage is shown in Figure 7. In this step, the system learns to minimize the text auto-encoder loss, as shown in Figure 7. The CTC and attention loss do not go through a big change in the retraining step because they have already been minimized. The inter-domain loss is calculated in an unsupervised manner, so the loss graph for the inter-domain loss remains steady throughout retraining.

6.3 Performance Comparison with External Language Model

To maintain fairness, we use the same unpaired text corpus to train the RNN language model in

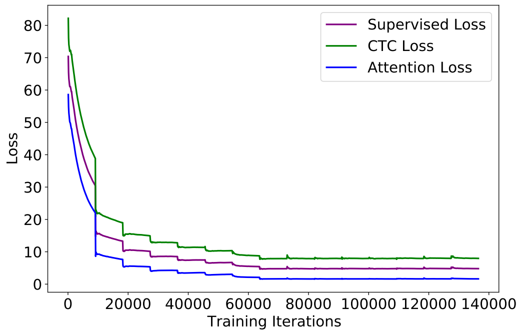

Figure 6: Supervised Training

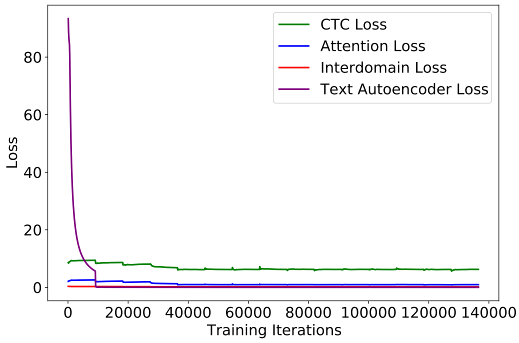

Figure 7: Semi-supervised Retraining

the baseline ASR model and the semi-supervised model. The only difference is, the semi-supervised model exploits the additional unpaired audio corpus. The RNN language model is used in shallow fusion with the baseline end-to-end system. Table 2 compares the Phoneme Error Rate (PER), Word Error Rate (WER), and Sentence Error Rate (SER) of our system with the baseline system with an external language model.

When we do not use any language model, the baseline end-to-end system achieves WER of 37%. Adding a word-level RNN language model improves the WER to 33.8%. The best accuracy in the baseline setup is achieved by the character level RNN where the WER is 32.5%. The character level RNN performs better than the word level RNN probably due to the presence of out-of-vocabulary words in the test set. The semi-supervised end-to-end system that exploits unpaired audio and text data outperforms all baseline setup and achieves WER of 31.9%. It is important to note that we do not use any separate language model with the semi-supervised system. The semi-supervised system already exploits the unpaired text data to some extent using text-to-text auto-encoder. But the performance of the semi-supervised system can be further improved by combining a language model during decoding.

Model Type	Language Model	PER (%)	WER (%)	SER (%)
Baseline	None	12.6	37.0	64.6
	Word	12	33.8	60.2
	Char	11.4	32.5	58.5
Semi-Supervised	None	11.3	31.9	58

Table 2: Performance Comparison with Baseline

Inter-Domain Loss	PER (%)	WER (%)	SER (%)
Guassian KL	11.9	34.0	60.8
MMD	11.4	32.7	59.1
GED	11.3	31.9	58

Table 3: Performance of Inter-Domain Loss

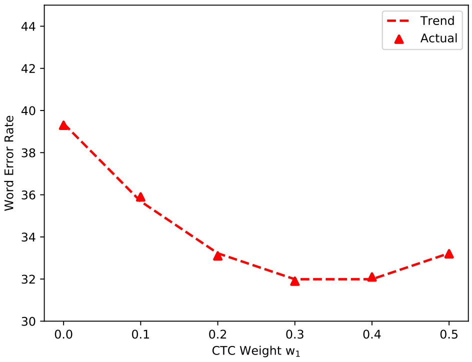

Figure 8: Effect of CTC Weight w_1

6.4 Performance Comparison of Inter-domain Loss

Table 3 shows the performance of the semi-supervised system for different inter-domain loss. Our proposed inter-domain loss based on global encoding distance achieves WER of 31.9% and SER of 58%, outperforming both Gaussian KL and MMD loss.

6.5 Effect of CTC Weight

Figure 8 shows the effect of the CTC weight w_1 (Equation 2) on the performance of our system. We found the best results when using CTC weight of 0.3. The tuning of the hyper-parameters is performed on a separate validation set.

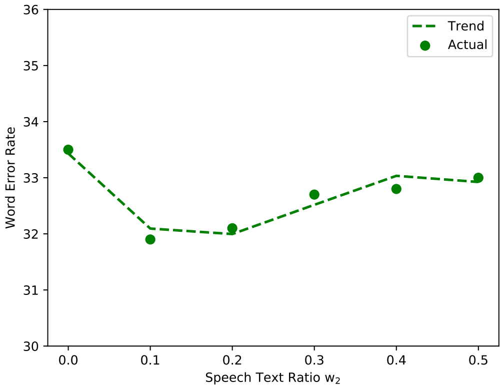

Figure 9: Effect of Speech Text Ratio w_2

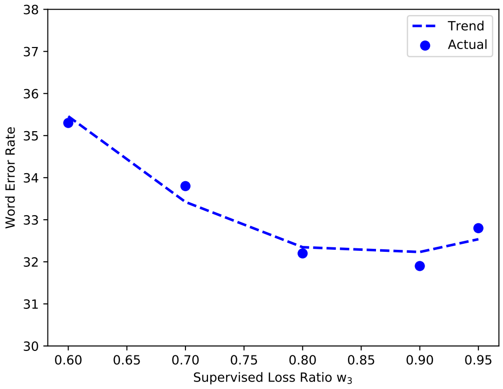

Figure 10: Effect of Supervised Loss Ratio w_3

6.6 Effect of Speech Text Ratio

Figure 9 shows the effect of the speech text ratio w_2 (Equation 3) on the performance of our system. We found the best results when using speech text ratio of 0.1.

6.7 Effect of Supervised Loss Ratio

Figure 10 shows the effect of the supervised loss ratio w_3 (Equation 4) on the performance of our system. We found the best results when using supervised loss ratio of 0.9.

6.8 Effect of Batch Size

Figure 11 shows the performance of the semi-supervised system with respect to batch size. The performance of the semi-supervised system with MMD loss decreases with smaller batch sizes. Our proposed GED loss is more robust to batch size and more likely to converge to the optimal solution even for small batch size.

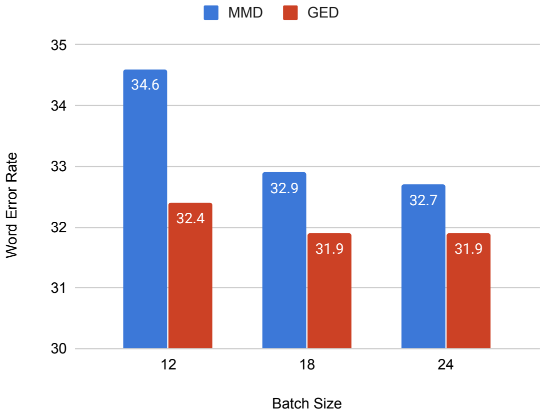

Figure 11: Effect of Batch Size

7 Conclusions

In this paper, we present a semi-supervised approach for the incorporation of unpaired audio data to boost the performance of a Bangla ASR system. Our proposed inter-domain loss function based on global encoding distance performs better than the Gaussian KL divergence and MMD loss proposed previously. We exploit 1000 hours worth of unpaired audio and a similar amount of text data in our semi-supervised training to optimize our speech recognition system. Our ASR which is trained on publicly available paired speech corpus and unpaired data resources outperforms the ASR trained only on the paired speech corpus with language models. In the future, we will try to improve the performance of the semi-supervised system further by fusing an additional language model during decoding.

Acknowledgements

This research work is conducted at the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET) and is supported by Samsung R&D Institute Bangladesh.

References

Akash Kumar Dhaka and Giampiero Salvi. 2017. Sparse autoencoder based semi-supervised learning for phone classification with limited annotations. In *Proc. GLU 2017 International Workshop on Grounding Language Understanding*, pages 22–26.

Thomas Drugman, Janne Pylkkonen, and Reinhard Kneser. 2019. Active and semi-supervised learning in asr: Benefits on the acoustic and language models. *arXiv preprint arXiv:1903.02852*.

Zhiyun Fan, Shiyu Zhou, and Bo Xu. 2019. Unsupervised pre-traing for sequence to sequence speech recognition. *arXiv preprint arXiv:1910.12418*.

Pengcheng Guo, Haihua Xu, Lei Xie, and Eng Siong Chng. 2018. Study of semi-supervised approaches to improving english-mandarin code-switching speech recognition. *arXiv preprint arXiv:1806.06200*.

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. *arXiv preprint arXiv:1706.02737*.

Shigeki Karita, Shinji Watanabe, Tomoharu Iwata, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. 2019. Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6166–6170. IEEE.

Shigeki Karita, Shinji Watanabe, Tomoharu Iwata, Atsunori Ogawa, and Marc Delcroix. 2018. Semi-supervised end-to-end speech recognition. In *Interspeech*, pages 2–6.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-sourced speech corpora for javanese, sundanese, sinhala, nepali, and bangladeshi bengali.

Rasa Lileikytė, Arseniy Gorin, Lori Lamel, Jean-Luc Gauvain, and Thiago Fraga-Silva. 2016. Lithuanian broadcast speech transcription using semi-supervised acoustic model training. *Procedia Computer Science*, 81:107–113.

Yuzong Liu and Katrin Kirchhoff. 2014. Graph-based semi-supervised acoustic modeling in dnn-based speech recognition. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 177–182. IEEE.

Yanhua Long, Yijie Li, Shuang Wei, Qiaozheng Zhang, and Chunxia Yang. 2019. Large-scale semi-supervised training in deep learning acoustic model for asr. *IEEE Access*, 7:133615–133627.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Luboš Šmídl, Jan Švec, Aleš Pražák, and Jan Trmal. 2018. Semi-supervised training of dnn-based acoustic model for atc speech recognition. In *International Conference on Speech and Computer*, pages 646–655. Springer.

Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky. 2013. Deep neural network features and semi-supervised training for low resource speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6704–6708. IEEE.

Karel Veselỳ, Mirko Hannemann, and Lukáš Burget. 2013. Semi-supervised training of deep neural networks. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 267–272. IEEE.

Karel Veselỳ, Carlos Segura, Igor Szöke, Jordi Luque, and Jan Cernockỳ. 2018. Lightly supervised vs. semi-supervised training of acoustic model on luxembourgish for low-resource automatic speech recognition. In *Interspeech*, pages 2883–2887.

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.

Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex Acero. 2010. Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech & Language*, 24(3):433–444.