Automatic Speech-to-Text Translation of Hindi

Shivam Kumar Pathak (skp454)
New York University

Richik Jaiswal (rmj324)
New York University

Abstract

We propose a end-to-end system which makes use of a recurrent encoder-decoder deep neural network to translate speech from the Hindi (Fourth most spoken language in the world) directly to the text in English(First most spoken language). We apply a slightly modified sequence-to-sequence with attention architecture that has previously been used for speech recognition and show that it can be repurposed for this more complex task. To address the lack of Hindi Audio to English Text, we create our own dataset using Speech and Text from Hindi media files. We demonstrate that our developed model successfully outperforms the traditional cascade of ASR and MT models. Although the model can learn the utterance of common words, it fails to learn uncommon words and the underlying grammar. Thus, we also propose methods to mitigate this challenge.

1 Introduction

Speech to text translation is the process by which conversational spoken phrases of one language is instantly translated to the text of another language. There are numerous benefits of speech-to-text translations. First, it can help speakers of different language to communicate efficiently. A system like this can be particularly beneficial in crisis relief situations where volunteers work in a different nation for natural disaster management. Upon receiving a request in a foreign language, this system can help to instantly translate it into the native language of volunteer, and assist him/her in adequately addressing the need. Second, this system can help in the documentation of the endangered languages. Lastly, it can help to extend access for local media resources to the global crowd. For example, Youtube auto-caption feature is currently capable of transcribing the audio to text in the same language for different languages of the world. Speech translation system can extend the use of this feature by providing an automatic translation of audio of different languages to text in the native language of the user.

Hindi is the fourth most spoken language in the world and the most spoken language in India. In recent years, the Indian economy has been growing with the fastest rate in the world. With the growing economy of India, the media and market using Hindi as spoken language are also growing fast. There is an obvious need for speech translation systems to translate Hindi audio to text in other languages. Previous attempts to translate Hindi language is limited to machine translation[12][20][19] models which translate text in the Hindi language to text in other languages. One notable implementation of this is Google Translate feature which is capable of translating Hindi language text and is publically available online. To the best of our knowledge, no attempt has been made so far to develop a speech translation system which can translate Hindi language audio.

Traditionally speech translation models have been made by cascading[23][7] separate models of Automatic Speech Recognition[10][9][26] and Machine Translation[2]. The disadvantage of a cascaded speech translation system like this is, the error from both the models get compounded because of cascading. Secondly, training a system of this kind requires extensive resources in the form of audio, its transcription in the same language, and its translation in the text form of the target language. Collecting these data is expensive as low resource languages are most often translated than transcribed. With the advent of the seq-to-seq neural network model, researchers have shifted

.

attention to building end-to-end sequence-to-sequence speech translation model[24][6][21] which are capable of translating the audio from one language to text of other in a single pass.

In this project, taking inspirations from the recent works in automatic speech-to-text translation, we attempt to build an end-to-end speech-to-text translation system for Hindi language, we benchmark our method by its comparison with a cascade of Google Speech(Hindi audio to Hindi text) and Google Translate(Hindi text to English Text). Further, we discuss the achievements and shortcomings of our model, and propose a possible method to solve these in the future.

2 Related Work

2.1 End-to-End Seq-to-Seq Model

Early works on speech-to-text translation[8] used lattices from the Automatic Speech Recognition[10][9] model as input to translation models[15][13]. The first notable attempt to use seq-2-seq model for processing audio data is present in the research work of Listen, Attend, and Spell[9]. Although the objective of this work is Automatic Speech Recognition rather than Automatic Speech Translation, the structure of the end-to-end model used here is also applicable for Speech Translation tasks[24][6]. In this model, audio files are first processed to yield Mel filterbank spectrogram. These spectrograms are input for the RNN based encoder(Listen) which learns the sequence-dependent representations from the speech. The decoder(Spell) uses the output of the encoder and produces text transcription by outputting one character at a time. The attend module comprising of global attention mechanism[22] handles the alignment between the encoder output and decoder input. Use of attention module reduces the burden on the last output of the encoder and helps to use output at all time steps of the encoder.

2.2 End-to-End Automatic Speech Translations

In the paper Listen and Translate[6], the same structure as LAS[9], was used to perform a speech-to-text translation on a corpus of synthetic speech. Unlike LAS[9], the Listen and Translate[6] decoder produces a translation in the text of another language rather than spelling it in the same language. Similar to this, in [24] the same structure was used to perform speech-to-text translation. The significant difference between both these work is the later made specific changes to improve the performance of encoders. These changes were the use of 80-dimensional Mel filterbank, delta, and delta-delta features in the spectrogram, and the use of deep CNN and ConvLSTM layers in encoders to process these multichannel spectrogram inputs. The work of Zhang et al., 2017 which used Very deep convolutional networks for end-to-end speech recognition[25] inspired these changes in the encoder. Also, in [24] a large dataset of real speech was used instead of synthetic speech.

2.3 Augmented approach to End-to-End Automatic Speech Translation

The paucity of training data is the biggest challenge in the end-to-end AST. In a lot of works[3][1][5][4] attempts have been made to combat this challenge. In [3], only 20 hours of speech data was used for training, which is relatively small for this problem. A major difference in the method made here to work with such a small dataset was to perform the decoding at word level rather than grapheme/character level as done in previous works[24][6][21]. In [5], to supplement the need of training data, an augmented corpus of speech-to-text data was developed using the Librispeech Data Corpus. More than 1000 hours of speech data from audiobook was aligned with the translation in the text of another language to create this augmented corpus. In [4], the encoder of the model was pre-trained with audio text pairs from high resource languages, and audio text of the same language to improve the performance. In [1], tied multitask approach was used to train the model for Speech Recognition and Translation at the same time. Thus, improving the performance of the overall model.

3 Data

In the previous works[3][6][1], standard speech-to-text data corpora were used. [3] and [1] used Fisher Spanish Speech dataset[18], and [6] used BTEC dataset[16] with French English Translations. Because we want to make a speech-to-text translation model for the Hindi Language, we made our

data corpus comprising of Hindi Speech audio and English text sentences parallel pairs. To create this data corpus, we used audio and subtitles files of 20+ Hindi movies. Each subtitles file comprises sentences of English text and a start & end timestamp corresponding to the time in the audio clip for which the given sentence is a translation. Using these timestamps from the subtitles files, we split the audio clip and create the Hindi Speech & English Text parallel pairs. After splitting, preprocessing was done on audio clips and text sentences separately.

3.1 Audio Preprocessing

Many audio clips in our data correspond to the music/song present in the Movie from which they were obtained. Thus, as a first step, we removed all the clips corresponding to music/songs. After this elimination, we were left with 28964 clips with a max length of 11 seconds and average length of 7 seconds. The standard practice to use audio data in machine learning is to convert them into spectrogram features by decomposing them in the frequency domain. We used librosa package to extract 80-Dimensional log Mel Filterbank, delta, and delta-delta features to produce these required spectrograms. We use a sliding window of 25ms, and stride of 10ms. All the audio clips with a length smaller than the max length were zero padded at the end to retain consistency in input arrays. Thus, the resulting arrays obtained from these audio clips have the shape of 3x80x1034, where 3 corresponds to the number of channels(Fbank, Delta & Delta-Delta), 80 corresponds to the frequency dimensions of Mel Fbank, and 1034 corresponds to the number of time windows.

3.2 Text Preprocessing

For text pre-processing first, we remove punctuation, unwanted characters and convert each word in the sentence to lower case. Next, all the unique words were extracted to create a dictionary. The vocabulary of our initial dictionary was 9364 words long with 164189 total occurrences in the dataset. Through our initial inspection, we found many words were present multiple types due to inconsistencies such as different verb form or singular and plural noun. These consistencies had artificially inflated the vocabulary size; thus, we use the lemmatize function from the NLTK package to process the variant forms of words present in vocabulary. Doing this reduced the vocabulary size by 2153 words without affecting the number of total occurrences. Post lemmatizing, we checked the number of occurrences of each word and found that around 6K words occurred less than only 10 times in the dataset. Thus, we eliminated these words and replaced them with placeholders. Finally, our vocabulary had 1322 unique words and 135K total occurrences.

After preprocessing both audio and text, we had 26835 parallel pairs. We split this data into Train, Dev, and Test with ratio 60:20:20.

4 Methodology

Inspired by the model structure of [6][24], we have three structural components in our model - Speech Encoder, Attention Alignment & Text Decoder, which we have explained in detail below. Figure 1 shows the architecture design of our model.

4.1 Speech Encoder

The speech encoder takes 80 Mel features frames at different time step (x1,x2...,xt,...,x1034) as input and transforms them into a sequence of hidden states (h1, h2,..., ht,..., hL). Similar to [24] the speech encoder consists of a stack of two convolution layers followed by RNN layers. The CNN layers are activated with ReLU activations[14], and each has 32 filters with shape 3x3 and strides 2x1 along the time and frequency axes. Striding at two steps downsamples the length of sequence along the time axis by a factor of 4, which helps in reducing computation in the subsequent RNN Layers. The RNN layers consist of 3 layers of bidirectional LSTM. Bidirectional LSTM helps to read the sequence both ways and improves the efficiency of hidden state to learn representation from the input sequence. Each Bidirectional LSTM layer has 256 hidden units and a dropout probability of 0.5.

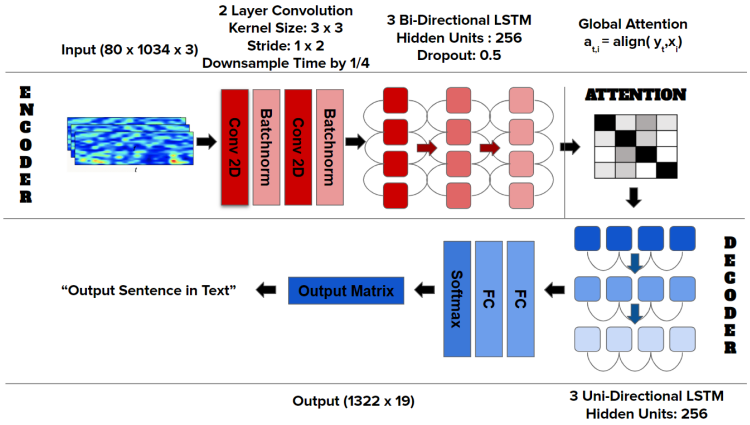

Figure 1: Model Architecture Design

4.2 Text Decoder

The text decoder performs next step prediction, i.e., emitting one output word at a time. The decoder consists of 3 layers of unidirectional LSTM with 256 hidden units and dropout probability of 0.5. A unidirectional LSTM helps to improve performance here as it easily learns the sequence dependency in the text and also helps in prediction of the next word. The decoder takes all the output hidden state of encoder along with the latent state emitted at the previous time step as input. It outputs one-word representation at a time and a latent state, which is used as an input to decode the next word. The output word representation is passed through 2 fully connected layers followed by a softmax layer to determine the corresponding word from the vocabulary.

4.3 Attention Alignment

The attention module is used to learn the alignment between the hidden state output of the encoder at each time step and the input of the decoder at each stage of word decoding. Traditionally in machine Translation models, the last hidden state of the encoder was only used for decoding all the words. The shortcoming of this approach was that it increased the burden on the last hidden state to learn the whole context. In work[2][11] introduced the attention model to determine the weight that should be multiplied to encoder hiddens state and produce the input for each stage of decoding words in Machine Translation. Even in AST, This process is extremely helpful to learn the alignment between the speech and the translated text.

4.4 Experimental Settings

All experiments were performed on an HPC Cluster provided by New York University using the CentOS operating system. The system consisted of 1 nodes powered by 125GB memory with 4 cores and a single gpu. The neural network model was developed on pytorch framework. The training was done in batches of size 16 for 120 epochs. The learning rate was set low at 0.003 with adam optimization as the optimizer. Cross entropy loss was used to find the accuracy of fit between the true and the predicted words.

4.5 Evaluation

To benchmark, the results of the translations produced by the developed system, a baseline consisting of separate modules for Automatic Speech Recognition and Machine Translation was used. Google cloud speech API was used to create the transcription of Hindi Speech in Hindi text(Automatic Speech Recognition), and the Google Translate API was used to translate these texts into the text of English language, i.e., the target language(Machine Translation). Uni-gram and sentence BLEU score[17] was used to evaluate the accuracy of produced translations by both the baseline and developed neural network model. Both these scores were calculated using the sentence_bleu function from the NLTK package in python.

Table 1: Uni-gram and sentence BLEU scores

Model	Mean Uni-Gram BLEU Score	Mean Sentence BLEU Score
Google ASR + MT	0.094	0.026
End-to-End AST	0.182	0.114

Short and Frequent:	True: [i, am, hungry]	**Good Uni-BLEU**
	Pred: [i, am, hungry]	**Good Sent-BLEU**
Short and infrequent:	True: [she, is, orphan]	**Bad Uni-BLEU**
	Pred: [he, else, the, is]	**Bad Sent-BLEU**
Short and Complex:	True: [where, is, munni]	**Good Uni-BLEU**
	Pred: [munni, where, is]	**Bad Sent-BLEU**
Long and Frequent:	True: [if, we, dont, find, munni, parent, will, she, stay, with, us]	**Moderate Uni-BLEU**
	Pred: [if, we, munni, be, no, the, find, what, she, us, be, stay]	**Bad Sent-BLEU**

Figure 2: Translations by End-to-End Automatic Speech Translation Model

5 Results

Table 1 shows the uni-gram and sentence BLEU scores for both the end-to-end automatic speech translation model we developed and the baseline model consisting of google ASR and MT. The end-to-end model was able to successfully outperform the baseline model on both the parameters, i.e. uni-gram and sentence BLEU scores. The possible reasons for this performance can be first, the error gets compounded in the cascaded model of ASR and MT. Second, as the BLEU score does not take into account the synonyms for similar words, the baseline model may have suffered due to this. In other works[24][3][6], the BLEU score on the translated text was considerably higher than what was yielded by our model. In [24], a sentence BLEU score of 0.483 was obtained on the Fisher Spanish English dataset, and a score of 0.487 was obtained on the Call Home dataset. In [6] a score of 0.436 was captured on Synthetic French English dataset. There are multiple possible reasons for the considerably low performance of our model in comparison to these works. First, we used a comparatively small size of the training dataset. Second, we did not use the Beam Search decoder as used in these works. A realistic comparison of our results can be made with the results of [3] which used a small size of the training dataset. In the work [3] for 50hours of the training dataset, a sentence BLEU score of approximately 0.15 was obtained, which is comparable to our results. To further understand the shortcoming of our model, we analyzed the translations produced by our model. These translations are present in Figure 2. The model can successfully map the utterance in a speech to text for common words, whereas it fails to do the same with uncommon words. Second, the model is not able to successfully reproduce the grammar as is present in the original sentence. In figure 3, we can see the uni-gram score distribution produced by the model is considerably higher than that of the baseline model. Hence, we can conclude that although the model can successfully learn the utterance of common words, it fails to learn the utterance of uncommon word and the underlying grammar in the sentence. To combat this, we plan to use an external language model with the decoder in our future work.

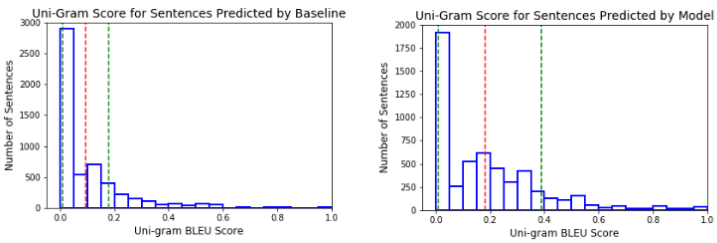

Figure 3: Uni-Gram BLEU score of Baseline and End-to-End Automatic Speech Translation Model

6 Conclusion

We proposed a model for end-to-end speech translation for translating Hindi Speech to English text. We demonstrated with the help of Encoder-Attention-Decoder structure, End-to-End automatic speech translation model can successfully outperform traditional ASR+MT model. Although with sufficient training data, the model is capable of learning utterances of common words from audio, it lacks in learning the grammar of sentences. The uni-gram and sentence BLEU score of the model is still not good enough to be used in applications. To mitigate this in the future, we intend to improve our decoder by using beam decoding and an external language model.

References

[1] Antonios Anastasopoulos and David Chiang. "Tied multitask learning for neural speech translation". In: *arXiv preprint arXiv:1802.06655* (2018).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[3] Sameer Bansal et al. "Low-resource speech-to-text translation". In: *arXiv preprint arXiv:1803.09164* (2018).

[4] Sameer Bansal et al. "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation". In: *arXiv preprint arXiv:1809.01431* (2018).

[5] Alexandre Bérard et al. "End-to-end automatic speech translation of audiobooks". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 6224–6228.

[6] Alexandre Bérard et al. "Listen and translate: A proof of concept for end-to-end speech-to-text translation". In: *arXiv preprint arXiv:1612.01744* (2016).

[7] Laurent Besacier, Bowen Zhou, and Yuqing Gao. "Towards speech translation of non written languages". In: *2006 IEEE Spoken Language Technology Workshop*. IEEE. 2006, pp. 222–225.

[8] Francisco Casacuberta et al. "Recent efforts in spoken language translation". In: *IEEE Signal Processing Magazine* 25.3 (2008), pp. 80–88.

[9] William Chan et al. "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 4960–4964.

[10] Jan K Chorowski et al. "Attention-based models for speech recognition". In: *Advances in neural information processing systems*. 2015, pp. 577–585.

[11] Jan Chorowski and Navdeep Jaitly. "Towards better decoding and language model integration in sequence to sequence models". In: *arXiv preprint arXiv:1612.02695* (2016).

[12] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. "Interlingua-based English–Hindi machine translation and language divergence". In: *Machine Translation* 16.4 (2001), pp. 251–304.

[13] Evgeny Matusov, Stephan Kanthak, and Hermann Ney. "On the integration of speech recognition and statistical machine translation". In: *Ninth European Conference on Speech Communication and Technology*. 2005.

[14] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[15] Hermann Ney. "Speech translation: Coupling of recognition and translation". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 1. IEEE. 1999, pp. 517–520.

[16] Sashi Novitasari et al. "Construction of English-French Multimodal Affective Conversational Corpus from TV Dramas". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. 2018.

[17] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.

[18] Matt Post et al. "Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus". In: *Proc. IWSLT*. 2013.

[19] Ananthakrishnan Ramanathan et al. “Simple syntactic and morphological processing can help English-Hindi statistical machine translation”. In: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*. 2008.

[20] RMK Sinha and A Jain. “AnglaHindi: an English to Hindi machine-aided translation system”. In: *MT Summit IX, New Orleans, USA* (2003), pp. 494–497.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[22] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[23] Alex Waibel and Christian Fugen. “Spoken language translation”. In: *IEEE Signal Processing Magazine* 25.3 (2008), pp. 70–79.

[24] Ron J Weiss et al. “Sequence-to-sequence models can directly translate foreign speech”. In: *arXiv preprint arXiv:1703.08581* (2017).

[25] Yu Zhang, William Chan, and Navdeep Jaitly. “Very deep convolutional networks for end-to-end speech recognition”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 4845–4849.

[26] Geoffrey Zweig et al. “Advances in all-neural speech recognition”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 4805–4809.