Tutorial on Speech-to-Speech Machine Translation

ICON2021

Presenters: Akshay Batheja, Rohit Kundu, Vineet Bhat, Shivam Mhaskar, Shyam Thombre, Sourabh Deoghare, Tamali Banerjee, Jyotsana Khatri, and Prof. Pushpak Bhattacharyya

IIT Bombay
Date: 19/12/2021
Contents

• Speech-to-Speech Machine Translation (SSMT)
• Automatic Speech Recognition (ASR)
• Disfluency Correction (DC)
• Machine Translation (MT)
• Automatic Post Editing (APE)
• Text-to-Speech (TTS)
• SSMT Demo
Speech-to-Speech Machine Translation
Problem Statement

• Speech-to-Speech Machine Translation: To translate speech in language A into speech in language B through use of a computer.
When the Moon comes in between the Sun and the Earth, solar eclipse happens.

When the Moon comes in between the Sun and the Earth, solar eclipse happens.

Text to Speech

Automatic Speech Recognition

Manual Editing

Manual Editing

Machine Translation

NLTM: Bahubhashak Project: Speech to Speech Machine Translation - Pipeline
Motivation

• SSMT technologies witnessing rapid growth:
  – Globalization, business needs, frequent travel, tourism industry fueling its demands.

• Huge range of applications, examples:
  – Movie dubbing
  – Movie Subtitling
  – Conversing with foreign language speakers
SSMT Market: USD 330 million in 2020 → USD 600 million in 2026

<table>
<thead>
<tr>
<th>Study Period:</th>
<th>2018 - 2026</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Year:</td>
<td>2020</td>
</tr>
<tr>
<td>Fastest Growing Market:</td>
<td>Asia Pacific</td>
</tr>
<tr>
<td>Largest Market:</td>
<td>Europe</td>
</tr>
<tr>
<td>CAGR:</td>
<td>9.4 %</td>
</tr>
</tbody>
</table>

https://www.mordorintelligence.com/industry-reports/speech-to-speech-translation#:~:text=The%20speech%20to%20speech%20translation%20market%20is%20valued%20at%20USD,period%20(2021%2D2026).
Benefits

• Bridging the language gap in global commerce and cross-cultural communication
  – Aids communication between people speaking different languages
• Education in mother tongue
  – Will dramatically reduce time for making Lectures available in Indian Languages
• Skills development amongst youth
• Technology could be scaled to school education
  – A promising solution for imparting quality school education in rural India
Challenges

• Conversational Spontaneous Speech
  – *It is a ahh umm beautiful day*
  – Ill-formed and incomplete sentences
• Indian English
  – Accent varies across the Nation
  – A problem for ASR
• Word Ordering
  – Significantly different for English and Indian languages
  – A problem for MT
• Code Switching
  – आज I am busy
• Lip Syncing
  – Translated sentences to Indian Languages are usually longer.
Cascaded SSMT Approach

ASR : Transcribing speech into text

MT : Translating text from language A to language B

TTS : Synthesizing speech from text
Well known Systems

US: Janus, DIPLOMAT, Tongues, Nespole! Maxtor
Europe: Verbmobil, Nespole! LC-Star, TC-Star
Japan: MATRIX NEC
China: LodeStar, Digital Olympics
SSMT: a Data Driven S2S problem

- Training data for an end-to-end ST model is very scarce.
- Available currently: only hundreds of hours of speeches, most of which are for
- Japanese–English translation and European languages [102,103]
- For Chinese–English translation, Baidu has released an open dataset containing 70 hours of speeches, including both the corresponding transcriptions and translations
- Through MEITY funded speech consortia led by IIT Madras, speech data in Indian languages is now available
Direct Speech to Speech Machine Translation
Direct Speech to Speech MT

• Problem Statement
  – To translate speech input in one language into speech in another language without relying on the intermediate step of text generation.

• Motivation
  – Lower computational costs and inference latency as compared to the cascaded systems.
  – To provide a translation support for languages that do not have a writing system.
Previous Approaches

• Cascaded S2S MT (2006) [2]
  – The Cascaded S2S pipeline consists of three components
    • ASR -> MT -> TTS

• End-to-End S2T (Speech to Text) (2019) [3]
  – Alleviates the error propagation issue between ASR and MT

  – S2T can be combined with TTS to provide both Speech and Text translation
Recent Developments

• Translatotron by Google AI (2019)
  – An attention-based sequence-to-sequence neural network which can directly translate speech from one language into speech in another language

• Direct S2ST with Discrete Units by Facebook AI and Johns Hopkins University (2021), state-of-the-art in Direct S2ST
Direct S2ST with Discrete Units (1/2)

• Key Contributions:

  – Self-supervised discrete representations of target speech is predicted instead of mel spectrogram features.

  – Model jointly generates speech and text output by combining S2ST and S2T tasks through the shared encoder and a partially shared decoder, for the languages where the transcripts are available at source as well as at target.
Key Contributions:

– In the scenarios where the transcripts for target language is unavailable, Direct S2ST model is trained with multitask learning using discrete representations for the source and target speech.

– The issue of length mismatch between the text and speech output during decoding is resolved using CTC (connectionist temporal classification).
Model Architecture (1/2)

- Transformer-based seq-to-seq model with a speech encoder and a discrete unit decoder and incorporates auxiliary tasks similar to *translatotron* during training to facilitate model learning.

Image source - Discrete Speech to Speech Translation [2]
Model Architecture (2/2)

- For target languages with text transcripts, target text CTC decoding is applied conditioned on the intermediate representations from the discrete unit decoder.

- This CTC decoding is used for jointly training text and speech.

- A **Vocoder** is separately trained to convert discrete units into a waveform.
The weather is cold today.
• Generation of Discrete Units for the target language:
  – HuBERT model is trained on an unlabelled speech corpus of the target language.
  – The trained model is used to encode the target speech into continuous representations at every 20ms frame.
  – A k-means algorithm is applied on the learned representations to generate K cluster centroids.
  – These K cluster centroids are used to encode the target utterances into sequences of cluster indices at every 20 ms.
A target utterance $y$ is represented as $[z_1, z_2, ..., z_T]$, where $z_i$ belongs to $\{0, 1, 2, ..., K-1\}$, $K$ is the number of clusters and $T$ is the number of frames.

S2U model is built by adapting from the transformer model for MT (Machine Translation).

A stack of 1-D convolutional layers, each with stride 2 and followed by a gated linear unit activation function, is prepended to the transformer layers in the encoder for downsampling the speech input.

As the target sequence is discrete, S2U model is trained with cross entropy loss with label smoothing.

Trained the Hi-Fi GAN neural vocoder for unit-to-waveform conversion.
Dataset used (1/2)

• Fisher Spanish-English speech translation corpus
  – The dataset consists of 139k sentences from telephone conversations in Spanish, the corresponding Spanish text transcriptions and their English text translation.

• Data Preparation:
  – A high-quality TTS engine is used to prepare synthetic target speech with a single female voice as the training targets.
Dataset used (2/2)

- Train, dev, test split:

<table>
<thead>
<tr>
<th></th>
<th>train</th>
<th>dev</th>
<th>dev2</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td># samples</td>
<td>126k</td>
<td>4k</td>
<td>4k</td>
<td>3.6k</td>
</tr>
<tr>
<td>source duration (hrs)</td>
<td>162.5</td>
<td>4.6</td>
<td>4.7</td>
<td>4.5</td>
</tr>
<tr>
<td>target duration (hrs)</td>
<td>139.3</td>
<td>4.0</td>
<td>3.8</td>
<td>3.9</td>
</tr>
</tbody>
</table>
• **Source and Target text transcripts are available:**
  – For the systems with dual mode output (like cascaded and S2U + CTC), both, the text output directly from the system and the ASR decoded text from the speech output, are evaluated.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Models</th>
<th>Test (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher Spanish-English Speech Translation Corpus</td>
<td>Translatotron + Pretrained encoder</td>
<td>31.1</td>
</tr>
<tr>
<td></td>
<td>Cascaded (S2T + TTS)</td>
<td>39.5</td>
</tr>
<tr>
<td></td>
<td>S2U reduced + CTC (w/ sc,tc)</td>
<td>37.2</td>
</tr>
</tbody>
</table>
Results (2/3)

- Source text transcripts available but target text transcripts are unavailable:
  - All models are trained without using any target text transcripts.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Models</th>
<th>Test (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher Spanish-English Speech Translation Corpus</td>
<td>Translatotron (w/ sp)</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>S2U reduced (w/ sc)</td>
<td>33.8</td>
</tr>
</tbody>
</table>
Results (3/3)

- Both source and target text transcripts are unavailable:
  - All models are trained without using any text transcripts.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Models</th>
<th>Test (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher Spanish-English Speech Translation</td>
<td>Translatotron no auxiliary task</td>
<td>0.6</td>
</tr>
<tr>
<td>Corpus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S2U reduced (w/ su)</td>
<td>27.1</td>
</tr>
</tbody>
</table>
Summary

• In scenario where text transcripts are available at source as well as target, S2U reduced model with joint speech and text training and auxiliary tasks, has bridged 83% of the gap between transformer-based Translatotron and the S2T+TTS cascaded baseline.

• Also demonstrated the possibility of translating between two unwritten languages by taking advantage of discrete representations of both the source and the target speech for model training.
Comparing Direct S2ST with Cascaded approach

<table>
<thead>
<tr>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascaded system have problem of errors compounding between components e.g.</td>
</tr>
<tr>
<td>Recognition errors leading to larger translational errors. Direct S2ST</td>
</tr>
<tr>
<td>model does not face such issues.</td>
</tr>
<tr>
<td>Reduced computational requirements and lower inference latency.</td>
</tr>
<tr>
<td>Paralinguistic and Non-linguistic information is retained during translation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>A large set of Input/Output speech pairs are required which are more difficult to collect than parallel text pairs for MT.</td>
</tr>
<tr>
<td>Cascaded S2ST is more robust.</td>
</tr>
<tr>
<td>Uncertain alignment between two spectrograms whose underlying spoken content differs also poses a major training challenge.</td>
</tr>
</tbody>
</table>
References


Automatic Speech Recognition
Topics to be covered in ASR

• A general introduction to ASR: Problem statement to mathematical description
• Discussion regarding the Hidden Markov Model based approach to ASR
• Introduction to Deep Learning based ASR
• Techniques for developing good quality speech recognition systems for Indian languages
• Code walkthrough for a SOTA training and evaluation pipeline for English ASR + Demo for Hindi ASR
Automatic Speech Recognition

- ASR is the task of using algorithms and methodologies to enable translation of speech signals to text by computers.
- Research has developed from 1960s to 2020s
- Speech recognition technologies have wide scale use in education, software development, utilities, luxury, military, etc.
- Well known examples: YouTube closed captioning, Voicemail transcription, Dictation Systems, etc.
Simplified pipeline of an ASR System

Image source: Lectures on Automatic Speech Recognition by Prof Preethi Jyothi, Course CS753, IIT Bombay
Motivation

• ASR Research has gained momentum over the years with the advent of faster computation and better resources
• Extensive data collection across widely spoken languages
• Disparity in Data availability for regional languages
• Innovative methods such as Transfer Learning and Knowledge sharing yet to be explored fully
Prior Work (Pre-Deep Learning era)

- Audrey by Bell Labs (1952): Digit Recognizer
- Shoebox by Bell Labs (1962): Isolated word Recognizer
- Harpy by Carnegie Mellon (1970s): 1000 words recognizer
- Introduction of Deep Learning in ASR (2010s)
- Dictate MCA by Dragon Inc (1980s) Typing documents through speech
- HMM in Speech Recognition
Mathematical Description of ASR

• We treat the acoustic input signal as \( X = \{x_1, x_2, x_3, x_4, \ldots \} \) a series of observations and define a sequence of words as the desired output \( W = \{w_1, w_2, w_3, w_4, \ldots \} \)

• Essentially, we are interested in obtaining the following -

\[
\hat{W} = \arg\max_{W \in L} P(W|X)
\]
• We can use Bayes rule to rewrite this as -

\[
\hat{W} = \arg \max_{W \in L} \frac{P(X|W)P(W)}{P(X)}
\]

\[
\hat{W} = \arg \max_{W \in L} P(X|W)P(W)
\]

• \(P(X|W)\): Probability of occurrences \(X\) given words \(W\) (Acoustic Models)
• \(P(W)\): Probability associated with word sequence (Language Models)
Acoustic Models:
• Contains statistical representations of each of the distinct sounds that make up a word
• 44 phonemes in English, each phoneme has its own HMM

Lexicon Models:
• Pronunciation models for speech recognizers; provides discriminating metric between pronunciations of the same word in different context
• Eg, consider “ough” as in through, dough, cough, rough, bough, thorough, enough, etc

Language Models:
• Predicting next word given a sequence of words
• Used to calculate $P(W)$ in previous expressions
Architecture of cascaded ASR System

Image source: Lectures on Automatic Speech Recognition by Prof Preethi Jyothi, Course CS753, IIT Bombay
Evaluation Metrics for ASR

- Characteristics of a good ASR Metric:
  - Direct
  - Objective
  - Interpretable
  - Modular

- Word Error Rate: Popular metric; measures the percentage of corrections needed to transform an incorrect word to a correct word.

\[
WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}
\]
ASR Error rates across the years

Image Source: A Historical Perspective of Speech Recognition, Communications of the ACM, 2014
Hidden Markov Models

• Each spoken word consists of a sequence of ‘l’ pronunciation segments called phonemes
• To compute over all possible pronunciations $Q$ of the word, we model the probability as -

$$P(X|W) = \sum_Q P(X|Q)P(Q|W)$$
• Each base phoneme $q$ is modelled as a HMM as depicted in the figure below -
• For a large vocabulary, including the dependence on all previous words in the sentence becomes computationally difficult task.
• As a result, N-gram models were constructed such that the dependence is limited to the last N words of the sequence -

\[
P(W) = \prod_{k=1}^{K} P\left(w_k|w_1, w_2, \ldots, w_{k-N+1}\right) \quad \quad P(w_k|w_{k-1}, w_{k-2}) = \frac{C(w_k w_{k-1} w_{k-2})}{C(w_k w_{k-1})}
\]
Viterbi Algorithm:

• Identifying the sequence of hidden variables given observed sequence
• Calculates the probability of the various possible outputs of the current time step keeping into account the associated probabilities of the previous time step outputs
• At each timestep we have a probability associated with all the possible paths in the decoding.
• The path with the highest probability is assigned as the Viterbi path and is continued forward to the next time step.
A short digress: Viterbi decoding for Part of Speech (POS) tagging

^ Brown foxes jumped over the fence .

Image source: Lectures on Speech and NLP by Prof Pushpak Bhattacharyya, Course CS626, IIT Bombay
End-to-End Deep Learning Based ASR Systems

Shortcomings of HMM ASR Systems:

- Local optimization of these models does not imply global optimization of the entire pipeline.
- Modules are difficult to train and often require a lot of manual and domain specific feature engineering to achieve good results specific to the domain chosen.
- Demand a large amount of annotated and aligned speech-text corpus.
- Compounding Errors problem.
Demonstrating the power of E2E Models

<table>
<thead>
<tr>
<th></th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN - HMM</td>
<td>5.0</td>
<td>5.8</td>
</tr>
<tr>
<td>E2E (Attention)</td>
<td>14.7</td>
<td>14.7</td>
</tr>
</tbody>
</table>

Results on Librispeech-100 corpus

<table>
<thead>
<tr>
<th></th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN - HMM</td>
<td>4.0</td>
<td>4.4</td>
</tr>
<tr>
<td>E2E (Attention)</td>
<td>4.7</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Results on Librispeech-960 corpus

- Three types of E2E systems - 1) CTC-Based Models, 2) RNN Transducer and 3) Attention-Based models

Results taken from “RWTH ASR Systems for LibriSpeech: Hybrid vs Attention”, Christoph et al, 2019
• A loss function but it solves the alignment problem while computing its loss function.
• CTC attempts to solve the data alignment problem as alignment between segments and audio is no longer needed.
• CTC uses the higher dimensional features learnt through DNNs to directly map audio inputs to hypothesis segments.
RNN - Transducer Model

- RNN encoder, analogous to an acoustic model, maps input audio features $x$ to hidden representations $h^e$
- The prediction network takes as input the previous output label prediction $w^u$ and maps it to a hidden representation $h^p$
- The joint network takes the encoder representation $h^e$ and the prediction network representation $h^p$
- Produces joint logits $z$ which are softmax normalized to provide an output distribution over the output label space and blank symbol
Attention Based ASR Models:

- Encoder model maps the input audio signal to a sequence of vectors instead of one fixed vector.
- The decoder then assigns weights to these sequence of vectors while concatenating them to decode the higher dimensional features.
- Thus at each time step, the previous as well as the future time step features are taken into account while decoding the particular character and alignment.

Image Source: Deep Context: End-to-End Contextual Speech Recognition, Pundak et al, 2018
Understanding Speech Data

Characteristics of a good ASR dataset:

• Noise Robustness: Presence of sufficiently noisy data for model to learn features with noise
• Diversity: Covering various accents and pronunciations

Bias in ASR:

• ASR unable to recognize and translate the speech impairment and children speech
• Voice assistants perpetuate a racial divide by misrecognising the speech of black speakers more often than of white speakers
Common Pre-processing Steps

**Speech:**
- Using denoising libraries to reduce background noise by isolating vocals

**Text:**
- Indian languages Text data must follow ‘utf-8’ encoding
- Removing punctuation marks
- Numbers written in text format
Low Resource ASR

• Low resource languages lack sufficient data to facilitate research and development of intelligent models. Most Indian languages belong to this category.
• Recent studies have shown that when DNNs are trained with speech signals from multiple languages, the features learnt are of high quality for downstream tasks.
• These features produce better results for speech recognition compared when the DNN is only trained on the data of a low resource language, which is often extremely less to reach optimization.
Citrinet512: Transfer learning in ASR (Part of Nvidia’s NeMo Research) (1/1)

- Citrinet512 is a CTC Based Model
- Contains 1D Convolutional layers with various blocks
- Each block consists of Convolutional layers, Batch Normalization, ReLU and Dropout

Image Source: Citrinet: Closing the Gap between Non-Autoregressive and Autoregressive End-to-End Models for Automatic Speech Recognition, Majumdar et al, 2021
An additional Squeeze and Excite (SE) layer is also present which improves the representational power of a network by modelling the inter-channel dependencies of convolutional networks.

Citrinet512 model can be used as a pretrained checkpoint (training on Eng speech) followed by finetuning for Hindi or any other Indian Language.

Finetuning is performed by replacing the vocabulary in the decoder with Devanagiri characters.
Wav2vec: Learning Speech representations from Audio

• Uses raw audio to learn speech features
• Masks spans of speech representations
• Training objective is to recover the masked audio through the context

Results of wav2vec

Finetuning Wav2Vec 2.0 model for multilingual ASR

- The XLSR Wav2Vec2 model is presented by Facebook AI in “Unsupervised Cross Lingual Representation Learning for Speech Recognition“ - Conneau et al, 2020
- XLSR stands for Cross-Lingual Speech Representations and refers to XLSR-Wav2Vec2’s ability to learn speech representations that are useful across multiple languages
- XLSR-Wav2Vec2 learns powerful speech representations from hundreds of thousands of hours of speech in more than 50 languages (including Hindi) of unlabeled speech
CLSRIL-23: Pretraining wav2vec 2.0 on Indian Languages ASR Data, July 2021

- Authors use wav2vec 2.0 to pretrain on 9k+ hours of unsupervised speech using 23 Indian Languages followed by finetuning in each of these languages.
- Comparison of a monolingual approach and multilingual approach is performed by analysing a model pretrained only on Hindi and a model pretrained on multiple languages.
Comparing both the approaches

• Training time for wav2vec 2.0 is much longer than the Citrinet512 model. For wav2vec 2.0, each epoch took about 1.5 hours for training whereas for Citrinet512, it took about 15 min.
  – Possible Reason: wav2vec 2.0 is a transformer based approach whereas Citrinet512 is a convolutional model, performing faster with a GPU
• Ability of the Citrinet512 is able to output punctuation marks much better than wav2vec 2.0
ASR DEMO
Summary of ASR

• Historical view of the ASR problem statement and the progress made in the last 70 years
• Understanding the traditional HMM models, its strengths and its weaknesses and how it paved the way for E2E systems
• Appreciating the state-of-the-art E2E system for ASR: wav2vec and multilingual training for low resource ASR
• Demo for English and Hindi ASR using this SOTA
Disfluency Correction
Disfluency

Speakers often use filler words, repeat fluent phrases, suddenly change the content of speech, and make corrections in their statement. These are some common disfluencies. Also, speakers use words like ”yeah”, ”well”, ”you know”, ”alright” etc. which do not contribute to the semantic content of the text but are only used to fill pauses or to start a turn, are considered to be disfluent.

Example: Well, this is this is you know a good plan.
Motivation

• Disfluency is a characteristic of spontaneous speech which is not present in written texts

• It reduces the readability of speech transcripts

• It also poses a major challenge to downstream tasks e.g. machine translation

• Since MT models are usually trained on fluent clean corpora, the mismatch between the training data and the actual use case decreases their performance.

• To tackle this challenge, specialized disfluency correction models are developed and applied as a post-processing step to remove disfluencies from the output of speech recognition systems
Disfluency Correction

Well, this is this is you know a good plan

Disfluent text

Disfluency Correction Model

this is a good plan

Fluent text
# Types of Disfluencies (1/2)

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filled Pause</td>
<td>Non lexicalized sounds with no semantic content.</td>
<td><em>but uh</em> we have to go through the same thing.*</td>
</tr>
<tr>
<td>Interjection</td>
<td>A restricted group of non lexicalized sounds indicating affirmation or negation.</td>
<td><em>uh-huh, I can understand the issue.</em></td>
</tr>
<tr>
<td>Discourse Marker</td>
<td>Words that are related to the structure of the discourse in so far that they help beginning or keeping a turn or serve as acknowledgment. They do not contribute to the semantic content of the discourse.</td>
<td><em>Well, this is a good plan.</em></td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Repetition or Correction</td>
<td>Exact repetition or correction of words previously uttered.</td>
<td>If if I <em>can’t</em> don’t know the answer myself, I will find it.</td>
</tr>
<tr>
<td>False Start</td>
<td>An utterance is aborted and restarted with a new idea or train of thought.</td>
<td><em>We’ll never find a day</em> what about next month?</td>
</tr>
<tr>
<td>Edit</td>
<td>Phrases of words which occur after that part of a disfluency which is repeated or corrected afterwards or even abandoned completely, to indicate that the previously uttered words are not intended.</td>
<td>We need two tickets, <em>I’m sorry</em>, three tickets for the flight to Boston.</td>
</tr>
</tbody>
</table>
Disfluency: Surface Structure (1/2)

- **Reparandum**
  - contains those words, which are originally not intended to be in the utterance
- **Interruption point**
  - marks the end of the reparandum
- **Interregnum**
  - consists of an editing term or a filler
- **Repair**
  - words from the reparandum are finally corrected or repeated here or a complete new sentence is started
Disfluency: Surface Structure (2/2)

Image source: Saini et al. [4]
SwitchBoard English Disfluency Correction corpus.

<table>
<thead>
<tr>
<th>Split</th>
<th>No of Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>55482</td>
</tr>
<tr>
<td>Validation</td>
<td>11889</td>
</tr>
<tr>
<td>Test</td>
<td>11889</td>
</tr>
</tbody>
</table>
Dataset (2/2)

SwitchBoard English Disfluency Correction corpus.

<table>
<thead>
<tr>
<th>Disfluent Sentence</th>
<th>Fluent Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>well i i just live in a</strong> i live in an apartment now</td>
<td>i live in an apartment now</td>
</tr>
<tr>
<td><strong>it's a</strong> it's a fairly large community</td>
<td>it's a fairly large community</td>
</tr>
<tr>
<td><strong>and so uh you know</strong> i'm kind of spoiled</td>
<td>i'm kind of spoiled</td>
</tr>
</tbody>
</table>
Data Preprocessing

1. Lower-case
2. Normalization
3. Punctuation removal
4. Apply Byte-Pair Encoding (BPE)
Evaluation

- BLEU score
- Precision, Recall, F1 score (for disfluency detection)
- Human evaluation
  - Disfluent words are dropped
  - Fluent words are retained

\[
\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad \quad \quad \quad \text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

\[
F1 \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]
Sequence to Sequence

Model the problem as a Translation task as if

Disfluent Sentence: Sentence in language 1
Fluent Sentence : Sentence in language 2

Performance on SWBD Testset: 95.01 bleu score
Usage of Joint BPE

- Using different BPE vocabulary at source and target sides, model achieved 94.70 bleu score on test set.
- Use of joint BPE vocabulary (from the concatenated corpus of disfluent sentences and fluent sentences) improved the performance to 95.01.
Style Transfer (1/3)

• Unsupervised: works without any parallel data (fluent, disfluent pairs)
• For every mini-batch of training, soft translations for a domain are first generated
• Subsequently they are translated back into their original domains to reconstruct the mini-batch of input sentences.
• The sum of token-level cross-entropy losses between the input and the reconstructed output serves as the reconstruction loss.
Style Transfer (2/3)
Style Transfer (3/3)

• It consists of a single encoder and a single decoder
• Since we are only operating on the English language in the source (disfluent) and target (fluent), it is important to utilize the benefit of parameter sharing.
• Single encoder and single decoder are used to translate in both directions, i.e., from disfluent to fluent text and vice-versa.
• The decoder is additionally conditioned using a domain embedding to convey the direction of translation, signifying whether the input to the encoder is a fluent or disfluent sentence.
Domain Embedding

Dimensionality reduced word embedding is concatenated with the domain embedding DE at every time-step (t) to form the input for the decoder.
Disfluency Correction using Unsupervised and Semi-supervised Learning: Results

<table>
<thead>
<tr>
<th>#Sentences</th>
<th>Percentage(%)</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(Unsupervised) 0</td>
<td>78.72</td>
<td>79.39</td>
</tr>
<tr>
<td>554</td>
<td>1</td>
<td>83.85</td>
<td>85.28</td>
</tr>
<tr>
<td>2774</td>
<td>5</td>
<td>84.67</td>
<td>86.03</td>
</tr>
<tr>
<td>5548</td>
<td>10</td>
<td>84.98</td>
<td>86.12</td>
</tr>
<tr>
<td>13870</td>
<td>25</td>
<td>85.88</td>
<td>87.04</td>
</tr>
<tr>
<td>27741</td>
<td>50</td>
<td>86.10</td>
<td>87.90</td>
</tr>
<tr>
<td>55482</td>
<td>100</td>
<td>87.16</td>
<td>88.22</td>
</tr>
</tbody>
</table>

Image Source: Saini et al. [3]
Disfluency Detection

• Detect the disfluent words in the sentences
• Remove those disfluent words from the disfluent sentence
• We would get the fluent sentence

Image Source: Saini et al. [4]
Disfluency Removal

• Detect the disfluent words in the sentences
• Remove those disfluent words from the disfluent sentence
• We would get the fluent sentence

Disfluent: this is this is you know a big problem
Fluent: this is a big problem
Sequence Tagging

• We tag each of the words as disfluent or fluent.
• 0 corresponds to fluent and 1 corresponds to disfluent.
• Train model on the sequence tagging task

Disfluent: this is this is a big problem
Fluent: this is a big problem
Tags: 1 1 0 0 0 0 0
Synthetic Data: Motivation

• Disfluency phenomenon is clearly visible in the Indian languages.

• But due to the unavailability of disfluency correction dataset, it is not possible to train models.

• This motivates the work of disfluency correction without any real parallel data.
Rule-based Disfluency Generation

Pronoun phrase repetition

○ Original fluent sentence: *I was saying that we should go for a movie*
○ Disfluent sentence: *I was I was saying that we should go for a movie*

Insertion of filler words

○ Original sentence: *The new year is looking grim*
○ Disfluent sentence: *ah ah the new year is looking grim*
Randomly insert frequent filler words

- **Bn:** বাপু-র নেতৃত্বে মানে পরিচালিত ঐতিহাসিক জন-আন্দোলন 'চম্পারনসত্যাগ্রহ'-এর প্রভাব ছিল সুদূর প্রসারিত।

- **Transliteration:** vApu-ra netRRitve mAne parichAlita aitihAsika jana-Andolana 'champAranasatyAgraha'-era prabhAva Chila sudUra prasArita|

- **En:** The historical mass movement ’Champaran Satyagraha’, I mean conducted under the leadership of Bapu, had a far-reaching effect.
Word Repetition

❖ Bn: সেজন্যই আমরা আপনাদের সহযোগিতায় দেশের সকল জমির জন্য 'মৃত্তিকা স্বাস্থ্য কার্ড' চালু চালু করার অভিযান শুরু করেছি।

❖ Transliteration: sejanyai AmarA ApanAdera sahayogitAyaঃ deshera sakala jamira janya 'mRRittikA svAsthya kAr.Da' chAlu chAlu karAra abhiyAna shuru kareChi|

❖ En: That is why we have started the campaign to introduce 'Soil Health Card' for all the lands of the country with your cooperation.
Few Rules for Bengali (3/6)

Synonym correction

❖ Bn: আজ আমরা খবরের কাগজের সংবাদপত্রের কথায়- সশস্ত্র বাহিনীকে তাদের ইচ্ছামতো কাজের পূর্ণ স্বাধীনতা দিয়েছি।

❖ Transliteration: Aja AmarA khavarera kAgajera saMvAdapatrera kathAyaঃ- sashastra vAhinlke tAdera ichChAmato kAjera pUrNa svAdhInatA diyaঃeChi|

❖ En: Today, in the words of the news paper newspaper, we have given the armed forces full freedom to do whatever they want.
Missing syllables

- Bn: ওডিশার সার্বিক উন্নয়নে কেন্দ্রীয় সরকারের প্রতিশ্রুতিদ্বারা প্রতিশ্রুতিবদ্ধতার কথা পুনর্ব্যক্ত করেন প্রধানমন্ত্রী।

- Transliteration: o.DaишAra sArvika unnaya⁠ne kendrlya⁠ sarakArera pratishrutivaddhatAra pratishrutivaddhatAra kathA punarvyakta karena pradhAnamantrI|

- En: The Prime Minister reiterated the central government's commitment to the overall development of Orissa.
Pronoun Correction

❖ Bn: সন্ত্রাসবাদের হুমকি যেভাবে দিন দিন বেড়ে চলছে, তাকে না তার মোকাবিলায় সম্ভাব্য পদক্ষেপ গ্রহণের বিষয়গুলি সম্পর্কেও আলোচনা করেন দুই প্রধানমন্ত্রী।

❖ Transliteration: santrAsavAdera humaki yebhAve dina dina ve.Daঁe chaleChe, tAke nA tAra mokAvilAyaঁ sambhAvya padakShepa grahaNera viShayaঁguli samparkeo AlochanA karena dui pradhAnamantrI|

❖ En: The two Prime Ministers also discussed the possible steps to be taken to counter the growing threat of terrorism.
Use part of word before the actual word

- **Bn:** কিন্তু প্রোমো প্রোমোটারচক্রের ফাঁদে পড়ে ঠকে যান।
- **Transliteration:** kintu promo promoTArcakra phA.Nde pa.Dae Thake yAna|
- **En:** But fall into the trap of the promo promoter cycle.
Summary

- Disfluency correction is a crucial step in SSMT pipeline
- It removes irregularities from speech transcriptions and make that ready for Machine Translation
- We are able to build DC systems with parallel data or monolingual data (in both domains) along with little amount parallel data
- Model trained on artificially generated data has limitations
- It would be interesting to take help from other language’s disfluency correction data
Disfluency Correction Demo

https://www.cfilt.iitb.ac.in/speech2text/


Machine Translation
Content

• Introduction
• Foundations
  – MT Paradigms
  – Neural Machine Translation
  – LaBSE Filtering
• Latest Developments
  – Pivoting
  – Phrase Table Injection
  – Back-Translation
  – Multilingual NMT
  – Unsupervised NMT
• Demonstration
What is Machine Translation?

• Automatic conversion of text from one language to another
  – Preserve the meaning
  – Fluent output text
History of MT

- 1954: First public demo of MT by IBM
  - Georgetown IBM experiment
- 1956: First MT conference
- 1972: Logos MT system
  - Translating military manuals into Vietnamese
  - Rule based approach
- 1993: Statistical MT
  - IBM models
- 2013: Neural Machine Translation
Why MT is hard?
Why MT is hard?

Language Divergence
Language divergence

- Languages express meaning in divergent ways
- **Syntactic divergence**
  - Arises because of the difference in structure
- **Lexical semantic divergence**
  - Arises because of semantic properties of languages
Different kinds of syntactic divergence

• Constituent order divergence (Word order)

English: He is waiting for him.
Hindi: वह उसके लिए इंतजार कर रहा है।

<table>
<thead>
<tr>
<th>Subject</th>
<th>He</th>
<th>वह</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb</td>
<td>waiting</td>
<td>इंतजार कर रहा है</td>
</tr>
<tr>
<td>Object</td>
<td>him</td>
<td>उसके</td>
</tr>
</tbody>
</table>

• Adjunction divergence

English: Delhi, the capital of India, has many historical buildings.
Hindi: भारत की राजधानी दिल्ली में बहुत सी ऐतिहासिक इमारतें हैं।

• Null subject divergence

English: I am going.
Hindi: जा रहा हूँ।
Different kinds of lexical semantic divergence

• Conflational divergence

  English: He stabbed him.
  Hindi: उसने उसे छुरे से मारा

• Categorial divergence (Lexical category change)

  English: They are competing.
  Hindi: वे प्रतिस्पर्धा कर रहे हैं

• Head-swapping divergence (Promotion or demotion of logical modifier)

  English: The play is on.
  Hindi: खेल चल रहा है
The Vauquois Triangle

MT Paradigms
Different paradigms of Machine Translation

• Rule based Machine Translation
• Statistical Machine Translation
• Example based Machine Translation
• Neural Machine Translation
Rule based Machine Translation

• Linguists create rules
• Three types
  – Direct
    • Map input to output with basic rules
  – Transfer based
    • Direct + Morphological and Syntactic analysis
    • The level of transfer is dependent on the language pairs
  – Interlingua based
    • Use an abstract meaning
    • Interlingua: Represent meaning of text unambiguously
    • It works at the highest level of transfer
• Performance of system highly dependent on experts who are creating rules
Statistical Machine Translation

• Learning from parallel corpora
• Three important things
  – Word translation
  – Word alignment
  – Word fertility management
• Problem to solve for SMT

\[ \hat{e} = \arg \max_e (P(e|f)) = \arg \max_e (P(e).P(f|e)) \]

e is target language sentence, f is source language sentence, P(e) is language model in target language and P(f|e) is translation model.
Example based Machine Translation

• Focus is on: Analogy
• Based on textual similarity
• Process
  – Analysis
    • Phrasal fragments of the input sentence
  – Transfer
    • Finding the aligned phrases from the database of examples
  – Generation
    • Recombination (Stitch together the aligned phrases)
Example based Machine Translation: Example

- Phrasal fragments: He buys, a book, on, Machine Translation
- Aligned phrases: Identifies the aligned phrases from the database

\[
\begin{align*}
\text{He buys: वह खरीदता है} \\
\text{a book: एक पुस्तक} \\
\text{on: पर} \\
\text{machine translation: मशीन अनुवाद}
\end{align*}
\]

- Recombination: Recombine those phrases to construct a sentence (Adjusting morphology, reordering)

\[
\text{वह मशीन अनुवाद पर एक पुस्तक खरीदता है।}
\]
Phrase based Statistical Machine Translation

• Why?
  – Translation of phrases is more intuitive

• Process involved
  – Two-way alignment
    • Using SMT (eg. IBM model 1)
  – Symmetrization
  – Expansion of aligned words to phrases (Phrase table construction)
Phrase based SMT: English to Hindi alignment

<table>
<thead>
<tr>
<th>वह</th>
<th>आज</th>
<th>शाम</th>
<th>को</th>
<th>केक</th>
<th>बनाने</th>
<th>की</th>
<th>योजना</th>
<th>बना</th>
<th>रहा</th>
<th>है</th>
</tr>
</thead>
<tbody>
<tr>
<td>He</td>
<td>is</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>is</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>planning</td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>make</td>
<td>a</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>cake</td>
<td>in</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>evening</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>वह</td>
<td>आज</td>
<td>हान</td>
<td>को</td>
<td>केक</td>
<td>बनाने</td>
<td>की</td>
<td>योजना</td>
<td>बना</td>
<td>रहा</td>
<td>है</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>----------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>He</td>
<td>is</td>
<td>planning</td>
<td>to</td>
<td>make</td>
<td>a</td>
<td>cake</td>
<td>in</td>
<td>the</td>
<td>evening</td>
<td></td>
</tr>
</tbody>
</table>

He is planning to make a cake in the evening.

Phrase based SMT: Hindi to English alignment
### Phrase based SMT: Phrase generation

- **Principle of coverage**: Every word must be in a phrase
- **Principle of non-vacuousness**: No empty phrases
- **Principle of consistency**: The aligned phrases must be consistent in the sense all words of phrase in source languages
MT Evaluation

- Manual evaluation
- Quality of sentence depends on two factors
  - Adequacy
    - How faithful the meaning of a sentence is transferred
  - Fluency
    - Acceptability of the native speaker

- Automatic evaluation measures
  - Word/phrase matching based
  - Edit distance based
  - Ranking based
BLEU score

- Bilingual Evaluation Understudy
- Word/Phrase matching based

\[
BLEU = BP \cdot \exp \left( \sum_{n=1}^{N} (w_n \cdot \log(p_n)) \right)
\]

- BP is brevity penalty, to penalize based on the length of the generated sentence.

\[
BP = \begin{cases} 
1 & c > r \\
e^{(1-r/c)} & c \leq r 
\end{cases}
\]

c = the length of the candidate translation, r = the effective reference corpus length, \( p_n \) is modified n-gram precision, \( w_n \) is weight (uniform in BLEU)
Example:
- 1-gram precision is 1.
- Modified 1-gram precision is 4/6.

The ratio of the number of phrases of length n present in candidate translation that are also present in reference translation and total number of phrases of length n in candidate translation.

In modified n-gram precision maximum count from reference translation.
Neural Machine Translation
Why Neural Machine Translation?

- In RBMT, EBMT, and SMT, there is no notion of similarity or relationship between symbolic representation of individual words.
- Ability to translate *I go to school* does not make these models capable of translating *I went to college*.
- However, Neural Network techniques work with distributed representations.
- NMT evaluates a single formula that explains all rules of the translation task. (Generalisation)
Paradigms of Machine Translation

- RBMT (Rule based machine translation)
- EBMT (Example based machine translation)
- SMT (Statistical machine translation)
- NMT (Neural machine translation)

Human effort → Data hungry

Rules handmade by human expert → Rules by learning from data
What is NMT?

• The task of MT is a sequence-to-sequence problem.
• It uses an encoder-decoder NN architecture with attention mechanism.
• NMT requires large parallel corpus.
• Here, we will discuss RNN-based and Transformer-based encoder-decoder architectures.
Simple RNN-based Encoder-Decoder [9] architecture overview

Image source - [http://www.iitp.ac.in/~shad.pcs15/data/nmt-rudra.pdf]
Summary vector representation

- John admires Mary
- Mary is in love with John
- Mary respects John
- John respects Mary

- I was given a card by her in the garden
- In the garden, she gave me a card
  - She gave me a card in the garden
- She was given a card by me in the garden
- In the garden, I gave her a card
  - I gave her a card in the garden

Image source- [9]
Problems with simple Encode-Decode paradigm (1/2)

What happens in enc-dec architecture-
1. Encoding transforms the entire sentence into a single vector.
2. Decoding process uses this sentence representation for predicting the output.

Problems-
• Quality of prediction depends upon the quality of sentence embeddings.
• After few time-step, summary vector may lose information of initial words of input sentence.
Problems with simple Encode-Decode paradigm (2/2)

Possible solutions-

• For prediction at each time step, present the representation of the relevant part of the source sentence only.

  the girl goes to school

  लड़की स्कूल जाती है

  – Attention-based encoder-decoder
Annotation vectors and context vectors

Attention weights are calculated from alignment scores which are output of another feed-forward NN which is trained jointly.
Attention-based Encoder-Decoder [10] architecture

Image source: [http://www.iitp.ac.in/~shad.pcs15/data/nmt-rudra.pdf]
Attention-based Encoder-Decoder [10] architecture
Attention-based Encoder-Decoder [10] architecture

Image source: [http://www.iitp.ac.in/~shad.pcs15/data/nmt-rudra.pdf]
Transformer [11]

- Motivations to choose Transformer over RNN:
  - Faster
  - More efficient.
- Architecture:
  - This is an encoder-decoder architecture with Transformers instead of RNNs.

Image source: [jalammar.github.io/illustrated-transformer/]
Transformer: Embedding

- Embedding:
  - Input of the encoder = sum(word_embedding, positional encoding)
  - To set a constant and small vector_size of positional encoding, researchers apply a strategy using sinusoidal function for which model can translate long sentences of the training set.

Image source: [jalammar.github.io/illustrated-transformer/]
Transformer: Encoder

1. Self attention-
   a. For each input token $X$, self attention mechanism generates an output vector $Z$ of same size.
   b. Multi-head attention-
      i. Input vectors are processed for multiple sets (heads) to get an output for each set.
      ii. Outputs are combined and processed then to get final encoder output $Z$.
2. Add and normalise - $\text{LayerNorm}(X+Z)$
3. Feedforward
4. Add and normalise

Image source: [jalammar.github.io/illustrated-transformer/]
Transformer: Multi-head attention

1. Multiply the input \(X\) (or output \(R\) of last encoder) with trainable \(W_i^Q, W_i^K, W_i^V\) to get \(Q_i, K_i, V_i\), for each head \(i\). (8 number of heads used).

2. Prepare \(Z_i\) of \(X\) for \(i\)-th head as \(\sum \text{softmax}((Q.K^T)/\sqrt{d})V^x\), where \(d\) is size of \(Q\).

3. \(Z_i\) to \(Z\) conversion → concatenate then multiply with trainable \(W^o\) to transform into a vector \(Z\) matching size of \(X\).

Q: Query; K: Key; V: Value.

Image source: [alammar.github.io/illustrated-transformer/](alammar.github.io/illustrated-transformer/)
Transformer: Enc-Dec attention

Same as self-attention, except–

- It takes the K and V from the output of the encoder stack and creates its Q from the layer below it.
1. Self-attention: In decoder side the self-attention layer is **only** allowed to attend to earlier positions in the output sequence. This is done by masking future positions.

2. Add and normalize

3. Encoder-decoder attention

4. Add and normalize

5. Feedforward

6. Add and normalize
Transformer: Entire scenario
Some BLEU scores for Indian Language NMT

<table>
<thead>
<tr>
<th>Language pair (src\tgt)</th>
<th>Hi</th>
<th>Pa</th>
<th>Bn</th>
<th>Gu</th>
<th>Mr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi</td>
<td>-</td>
<td>60.77</td>
<td>28.75</td>
<td>52.17</td>
<td>31.66</td>
</tr>
<tr>
<td>Pa</td>
<td>64.67</td>
<td>-</td>
<td>25.32</td>
<td>44.74</td>
<td>27.78</td>
</tr>
<tr>
<td>Bn</td>
<td>31.79</td>
<td>26.96</td>
<td>-</td>
<td>24.82</td>
<td>16.61</td>
</tr>
<tr>
<td>Gu</td>
<td>55.02</td>
<td>46.48</td>
<td>25.33</td>
<td>-</td>
<td>25.62</td>
</tr>
<tr>
<td>Mr</td>
<td>42.97</td>
<td>37.08</td>
<td>21.82</td>
<td>33.29</td>
<td>-</td>
</tr>
</tbody>
</table>

Summary

• Machine translation is a hard problem because of language divergence.
• BLEU is an automatic evaluation metric to measure the quality of MT output.
• RBMT, EBMT, SMT, and NMT are 4 paradigms of MT.
• From RBMT to NMT, need for human effort decreases with the cost of data availability.
• We discussed RNN-based and Transformer-based encoder-decoder architectures.
LaBSE Filtering
Introduction

Techniques to extract good quality parallel data from the Hindi-Marathi Samanantar Corpus to improve the quality of our Hindi-Marathi MT models.
Motivation

- Neural Machine Translation (NMT) models are “data hungry”.

- The comparable corpora have increased tremendously on the World Wide Web, making it an important source for MT task.

- The mined sentence pairs are high in quantity but their quality varies a lot. This affects the quality of the MT systems.

- Hence, there is a need to come up with a preprocessing step to extract only the good quality sentence pairs from the comparable and parallel corpora before passing them to the MT model.
Literature

- Techniques:
  - LaBSE
  - Distilled PML
LaBSE by Google AI

- Language agnostic BERT sentence embedding model is based on a multilingual BERT model.

- Supports 109 languages including some Indic-languages.
LaBSE

- What is Multilingual Embedding Model?
  - that maps text from multiple languages to a shared vector space.
  - Means similar words will be closer and unrelated words will be distant in the vector space as shown in fig:

![Image source- Language agnostic Bert Sentence Embedding [1]](148)
The model architecture is based on Bi-Directional dual encoder with an additive margin loss.
LaBSE Training PIPELINE

• Firstly multilingual BERT model is trained on 109 languages for MLM (Masked Language Modelling) task.

• The obtained BERT encoders is used in parallel at source and target for fine-tuning the Translation Ranking Task.
What is Softmax Loss?

• Confusion? Softmax activation and Softmax loss are different?

• It is a softmax activation followed a Cross-Entropy loss

• It is used for multiclass classification.

• Also known as Categorical Cross-Entropy loss.

\[
f(s)_i = \frac{e^{s_i}}{\sum_j e^{s_j}} \quad CE = -\sum_i t_i \log(f(s)_i)
\]
Additive Margin Softmax loss

• Motivation :
  – In a classification task, we face a problem when output lies near the decision boundary in the vector space.
  – AM-Softmax aims to solve this by adding a margin to the decision boundary in order to increase the separability of the classes and also making the intra-class distance more compact.
Literature

• Techniques:
  – LaBSE
  – Distilled PML
Distilled PML

• Distilled Paraphrase Multilingual Model is a Sentence BERT (SBERT) model extended to multiple languages using multilingual knowledge distillation.

• Knowledge Distillation : Compressing a model by teaching a smaller network exactly what to do at each step using an already bigger trained model.

• A Teacher-Student Model Architecture is use to train Distilled PML model.
Given parallel data (e.g. English and German), train the student model such that the produced vectors for the English and German sentences are close to the teacher English sentence vector.
Model Training

• The English SBERT model is chosen as a teacher model.

• XLM-RoBERTa (XLM-R) model is chosen as a student model.

• So in short student model is trained using XLM-R and further fine-tuned on English NLI (Natural language Inference) and STS (Semantic Text Similarity) task using English SBERT model.
Dataset Used (1/3)

- Samanantar Corpus
  - It is the biggest parallel corpus publically available for Indic languages. In our experiments we used Hindi-Marathi Samantar corpus

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of Parallel Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samanantar Corpus</td>
<td>19L</td>
</tr>
</tbody>
</table>
Dataset Used (2/3)

- Combined Corpus:

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of Parallel Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIB</td>
<td>1,08,063</td>
</tr>
<tr>
<td>PMI</td>
<td>29,973</td>
</tr>
<tr>
<td>Tatoeba</td>
<td>46,277</td>
</tr>
<tr>
<td>ILCI</td>
<td>4,62,777</td>
</tr>
<tr>
<td>Total Combined Corpus</td>
<td>6,07,832</td>
</tr>
</tbody>
</table>
Dataset Used (3/3)

- Test Datasets:

<table>
<thead>
<tr>
<th>Corpus Name</th>
<th># of Test Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAT21</td>
<td>2390</td>
</tr>
<tr>
<td>ILCI</td>
<td>2000</td>
</tr>
</tbody>
</table>
Approach

- LaBSE model is used to generate the sentence embeddings of the Hindi-Marathi Samanantar Corpus.

- These embeddings are used to compute the cosine similarity between the Hindi-Marathi sentence pairs.

- Based on these similarity scores we extract the good quality sentence pairs using a threshold similarity score.

- Then we use these good quality sentence pairs to train the Hindi-Marathi MT systems.
Implementation

• Experiments:
  – Baseline
  – Without LaBSE Filtering
  – LaBSE
We use only the combined corpus to train the Hindi-Marathi Baseline models.

The combined corpus consists of 6L sentences. The train and tune split given below

<table>
<thead>
<tr>
<th>Corpus</th>
<th>#Train</th>
<th>#Tune</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Corpus + Tatoeba</td>
<td>6,07,832</td>
<td>14,390</td>
</tr>
<tr>
<td>(ILCI + PMI + PIB +Bible +Tatoeba)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Without LaBSE Filtering

• In this experiment we trained another Hindi-Marathi MT model using the Combined Corpus and whole Samanantar Corpus.

• The train and tune split is shown below:

<table>
<thead>
<tr>
<th>Corpus</th>
<th># Train</th>
<th># Tune</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Corpus + Tatoeba (ILCI + PMI + PIB + Bible + Tatoeba)</td>
<td>6,07,832</td>
<td>14,390</td>
</tr>
<tr>
<td>Samanantar</td>
<td>19,72,689</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>25,80,677</td>
<td>14,390</td>
</tr>
</tbody>
</table>
LaBSE based Filtering

• Hindi-Marathi MT model is trained using the Combined Corpus and LaBSE filtered Samanantar Corpus.

• We use the LaBSE model provided by the huggingface to generate the LaBSE scores for the whole Samanantar Corpus.

• We also computed the LaBSE scores on the PMI corpus, which is a good quality Hindi-Marathi parallel corpus.

• We computed the average LaBSE score which turned out to be 0.89. So we chose 0.9 as the threshold LaBSE score.

LaBSE model provided by the huggingface: https://huggingface.co/sentence-transformers/LaBSE
## Samanantar LaBSE Data Analysis

<table>
<thead>
<tr>
<th>LaBSE score Range</th>
<th>No. of Parallel Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=0.9</td>
<td>3,54,315</td>
</tr>
<tr>
<td>&gt;=0.91</td>
<td>2,89,802</td>
</tr>
<tr>
<td>&gt;=0.92</td>
<td>2,32,187</td>
</tr>
<tr>
<td>&gt;=0.93</td>
<td>1,80,776</td>
</tr>
<tr>
<td>&gt;=0.94</td>
<td>1,36,200</td>
</tr>
<tr>
<td>&gt;=0.95</td>
<td>97,860</td>
</tr>
<tr>
<td>&gt;=0.96</td>
<td>65,167</td>
</tr>
<tr>
<td>&gt;=0.97</td>
<td>38,699</td>
</tr>
<tr>
<td>&gt;=0.98</td>
<td>17,796</td>
</tr>
<tr>
<td>&gt;=0.99</td>
<td>4,103</td>
</tr>
</tbody>
</table>
LaBSE based Filtering

- We extracted 3.5L sentences from Samanantar Corpus that had a LaBSE score of 0.9 and above.
- The train, tune split for this model is given below

<table>
<thead>
<tr>
<th>Corpus</th>
<th>#Train</th>
<th>#Tune + Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Corpus + Tatoeba</td>
<td>6,07,832</td>
<td>14,390</td>
</tr>
<tr>
<td>(ILCI + PMI + PIB +Bible +Tatoeba)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samanantar_labse (labse&gt;=0.9)</td>
<td>3,54,314</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>9,62,146</td>
<td>14,390</td>
</tr>
</tbody>
</table>
Implementation (1/2)

• Training
  – We have used transformer architecture for all our models.
  – We trained the NMT model with the help of OpenNMT-py library
Implementation (2/2)

• Training

  – The parameters for the transformer model are shown below

<table>
<thead>
<tr>
<th>Encoder Type</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoder Type</td>
<td>Transformer</td>
</tr>
<tr>
<td>Number of layers in encoder/decoder</td>
<td>6</td>
</tr>
<tr>
<td>Number of attention heads</td>
<td>8</td>
</tr>
<tr>
<td>Size of encoder embedding dimensions</td>
<td>512</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.1</td>
</tr>
</tbody>
</table>
• Hindi-Marathi MT model

<table>
<thead>
<tr>
<th>Models</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WAT21</td>
</tr>
<tr>
<td>Baseline</td>
<td>13.8</td>
</tr>
<tr>
<td>Without LaBSE filtering</td>
<td>16.9</td>
</tr>
<tr>
<td>LaBSE filtering</td>
<td>17.8</td>
</tr>
</tbody>
</table>

We used sacrebleu python library to calculate the BLEU scores.
[https://github.com/mjpost/sacrebleu](https://github.com/mjpost/sacrebleu)
Results (2/5)

- Hindi-Marathi MT Model
  - We see an increment of 4 BLEU score points in LaBSE filtered model as compared to Baseline on WAT21 test data.
  - Increment of 1 BLEU score points as compared to the “without LaBSE filtered model” on WAT21 test data.
  - We also see that the BLEU score on ILCI dataset remains the same for Baseline and LaBSE filtered model, while it decreases by 0.2 points for “without LaBSE filtered model”.
Results (3/5)

- Marathi-Hindi MT model

<table>
<thead>
<tr>
<th>Models</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WAT21</td>
</tr>
<tr>
<td>Baseline</td>
<td>22.1</td>
</tr>
<tr>
<td>Without LaBSE filtering</td>
<td>21.6</td>
</tr>
<tr>
<td>LaBSE filtering</td>
<td>25.1</td>
</tr>
</tbody>
</table>
Results (4/5)

- Marathi-Hindi MT Model
  - Increment of 3 BLEU score points in LaBSE filtered model as compared to Baseline on WAT21 test data.

  - Increment of 4 BLEU score points as compared to “without LaBSE filtered model” on WAT21 test data.

  - We also see that the BLEU score on ILCI dataset, increments by 0.5 for LaBSE filtered model as compared to Baseline, while it decreases by 4 points for “without LaBSE filtered model”.

  - This is because the Samanantar corpus doesn’t consist of the in-domain data of ILCI dataset.
Results (5/5)

• We also computed the Spearman’s rank correlation coefficient between LaBSE and Distilled PML scores.

• These scores were computed on a set of 5000 Hindi-Marathi parallel sentences.

• The correlation coefficient turned out to be 0.38

Scatter plot of LaBSE and Distilled PML scores
Summary

• Neural Machine Translation systems are “data-hungry”.

• But the quality of the parallel data is as important as the quantity.

• We presented an approach to extract good quality parallel data from the Samanantar Corpus using the LaBSE score to improve the Hi-Mr MT systems.

• This helped us defeat the Baseline Hi-Mr MT systems.
References


Latest Developments
Phrase Table Injection (1/3)

- In this technique, phrase table is extracted from the Source-Target parallel corpus.
- Finally the Source-Target NMT model is trained using the Source-Target Parallel Corpus and Source-Target Phrases.
Phrase Table Injection (2/3)
Phrase Table Injection (3/3)

• **Dataset**

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test (WAT 2021)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sentences</td>
<td>250,347</td>
<td>2390</td>
</tr>
</tbody>
</table>

• **Results**

<table>
<thead>
<tr>
<th></th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>16.26</td>
</tr>
<tr>
<td>Phrase Table Injection</td>
<td>17.15</td>
</tr>
</tbody>
</table>
Pivoting

- *Pivoting* means utilizing the resources of a related high-resource language for the task of translation between language pairs involving low resource language.

- Example:
  - Utilizing the resources of Hindi for the task of translation between English-Marathi.
  - Utilize the resources of English for the task of translation between Hindi-Marathi or distant language pairs like Russian-Marathi.
Pivoting: Cascade Models

- Cascade Model

Source Sentence → Source-Pivot Model → Pivot Sentence → Pivot-Target Model → Target Sentence
Pivoting: Cascade Models

Advantages

- Translation between language pairs that don’t have sufficient parallel corpus, but each of those language has sufficient parallel corpus with pivot language (for example English).
Pivoting: Cascade Models

Disadvantages

● Double Decoding Time
  ○ As the source sentence is passed through two NMT models the decoding time is doubled.

● Propagating Errors
  ○ As the source sentence is passed through two NMT models, each model introduces its own errors in translation.
Pivoting: Combined Corpus Model

- In this technique, first the Source-Target and Source-Pivot parallel corpus are combined and a NMT model is trained on this combined data.

- This model is then used as an initialization and the final Source-Target NMT model is trained by finetuning on the Source-Target Parallel corpus.
Pivoting: Combined Corpus Model

Diagram:
1. En-Hi, En-Mr
2. En-Hi, En-Mr → Train → NMT Model
3. En-Mr → Finetune → Final En-Mr NMT Model
Pivoting: Combined Corpus Model

- Initially training the model on combined (En-Hi, En-Mr) the model learned some representation and knowledge from the pivot language data (En-Hi).
- This representation and knowledge can be useful for the final task of En-Mr translation.
Pivoting: Combined Corpus Model

• Dataset

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test (WAT 2021)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sentences</td>
<td>250,347</td>
<td>2390</td>
</tr>
</tbody>
</table>

• Results

<table>
<thead>
<tr>
<th></th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>16.26</td>
</tr>
<tr>
<td>Combined Corpus</td>
<td>18.02</td>
</tr>
</tbody>
</table>
Pivoting: Direct Pivoting Model

- In Direct Pivoting technique we initially train 2 models: source-to-pivot and pivot-to-target.
- Then we initialize the encoder and decoder of the source-to-target model using the encoder of source-to-pivot model and decoder of pivot-to-target model.
- Then we finetune the source-to-target model on source-target parallel data.
Pivoting: Direct Pivoting Model

Source-Pivot NMT (Task 1) → Pivot Decoder

Source-Target NMT (Task 3) → Target Decoder

Pivot-Target NMT (Task 2) → Pivot Decoder

Finetune

Initialize

[13]
Pivoting: Direct Pivoting Model

• **Dataset**

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test (WAT 2021)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sentences</td>
<td>250,347</td>
<td>2390</td>
</tr>
</tbody>
</table>

• **Results**

<table>
<thead>
<tr>
<th></th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>16.26</td>
</tr>
<tr>
<td>Direct Pivoting</td>
<td>16.68</td>
</tr>
</tbody>
</table>
Multilingual NMT

● Motivation
  ○ Translation between $N$ languages to $N$ languages will require $O(N^2)$ models.
  ○ A single $N$-to-$N$ multilingual model can translate between all $O(N^2)$ language directions.
  ○ Multilingual Models share knowledge between all languages improving performance for low resource language pairs.
• Parameter sharing: Shared encoder and decoder
• Need to find the right amount of shared parameters

Google’s MNMT System (Johnson et al., 2017)

- A single multilingual model

### Table 1: Many to One: BLEU scores on for single language pair and multilingual models. *: no oversampling

<table>
<thead>
<tr>
<th>Model</th>
<th>Single</th>
<th>Multi</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT De→En</td>
<td>30.43</td>
<td>30.59</td>
<td>+0.16</td>
</tr>
<tr>
<td>WMT Fr→En</td>
<td>35.50</td>
<td>35.73</td>
<td>+0.23</td>
</tr>
<tr>
<td>WMT De→En*</td>
<td>30.43</td>
<td>30.54</td>
<td>+0.11</td>
</tr>
<tr>
<td>WMT Fr→En*</td>
<td>35.50</td>
<td>36.77</td>
<td>+1.27</td>
</tr>
<tr>
<td>Prod Ja→En</td>
<td>23.41</td>
<td>23.87</td>
<td>+0.46</td>
</tr>
<tr>
<td>Prod Ko→En</td>
<td>25.42</td>
<td>25.47</td>
<td>+0.05</td>
</tr>
<tr>
<td>Prod Es→En</td>
<td>38.00</td>
<td>38.73</td>
<td>+0.73</td>
</tr>
<tr>
<td>Prod Pt→En</td>
<td>44.40</td>
<td>45.19</td>
<td>+0.79</td>
</tr>
</tbody>
</table>

### Table 2: One to Many: BLEU scores for single language pair and multilingual models. *: no oversampling

<table>
<thead>
<tr>
<th>Model</th>
<th>Single</th>
<th>Multi</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT En→De</td>
<td>24.67</td>
<td>24.97</td>
<td>+0.30</td>
</tr>
<tr>
<td>WMT En→Fr</td>
<td>38.95</td>
<td>36.84</td>
<td>-2.11</td>
</tr>
<tr>
<td>WMT En→De*</td>
<td>24.67</td>
<td>22.61</td>
<td>-2.06</td>
</tr>
<tr>
<td>WMT En→Fr*</td>
<td>38.95</td>
<td>38.16</td>
<td>-0.79</td>
</tr>
<tr>
<td>Prod En→Ja</td>
<td>23.66</td>
<td>23.73</td>
<td>+0.07</td>
</tr>
<tr>
<td>Prod En→Ko</td>
<td>19.75</td>
<td>19.58</td>
<td>-0.17</td>
</tr>
<tr>
<td>Prod En→Es</td>
<td>34.50</td>
<td>35.40</td>
<td>+0.90</td>
</tr>
<tr>
<td>Prod En→Pt</td>
<td>38.40</td>
<td>38.63</td>
<td>+0.23</td>
</tr>
</tbody>
</table>
Massively Multilingual Neural Machine Translation

- Trained Multilingual model on 59 languages. (Aharoni et al., 2019)

<table>
<thead>
<tr>
<th></th>
<th>En-Az</th>
<th>En-Be</th>
<th>En-Gl</th>
<th>En-Sk</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td># of examples</td>
<td>5.9k</td>
<td>4.5k</td>
<td>10k</td>
<td>61k</td>
<td>20.3k</td>
</tr>
<tr>
<td>baselines</td>
<td>2.16</td>
<td>2.47</td>
<td>3.26</td>
<td>5.8</td>
<td>3.42</td>
</tr>
<tr>
<td>one-to-many</td>
<td>5.06</td>
<td>10.72</td>
<td>26.59</td>
<td>24.52</td>
<td>16.72</td>
</tr>
<tr>
<td>many-to-many</td>
<td>3.9</td>
<td>7.24</td>
<td>23.78</td>
<td>21.83</td>
<td>14.19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>En-Ar</th>
<th>En-De</th>
<th>En-He</th>
<th>En-It</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td># of examples</td>
<td>213k</td>
<td>167k</td>
<td>211k</td>
<td>203k</td>
<td>198.5k</td>
</tr>
<tr>
<td>baselines</td>
<td>12.95</td>
<td>23.31</td>
<td>23.66</td>
<td>30.33</td>
<td>22.56</td>
</tr>
<tr>
<td>one-to-many</td>
<td>16.67</td>
<td>30.54</td>
<td>27.62</td>
<td>35.89</td>
<td>27.68</td>
</tr>
<tr>
<td>many-to-many</td>
<td>14.25</td>
<td>27.95</td>
<td>24.16</td>
<td>33.26</td>
<td>24.9</td>
</tr>
</tbody>
</table>

Table 3: En→X test BLEU on the TED Talks corpus
Unsupervised MT
Unsupervised MT

- No parallel corpus
- Train using only monolingual data
- However, the requirement is:
  - Large monolingual corpus
  - Cross-lingual Word Embeddings
Cross-lingual Word Embeddings

• The geometric relations that hold between words are similar across languages.
  – For instance, numbers and animals in English show a similar (isomorphic) geometric structure as their Spanish counterparts.
• The vector space of a source languages can be transformed to the vector space of the target language $t$ by learning a linear projection with a transformation matrix $W_{s\rightarrow t}$. 
Autoencoder

- Representation learning
- Neural network to learn reconstruction of the data
- Optimize **Reconstruction Error**
- Balance between
  - Accurately build a reconstruction
  - Handle inputs such that the model doesn’t learn to copy the data

\[
\begin{align*}
\mathbf{h}_1 & \rightarrow \mathbf{h}_2 \\
\mathbf{h}_2 & \rightarrow \mathbf{h}_3 \\
\mathbf{h}_3 & \rightarrow \mathbf{x}_5
\end{align*}
\]
Denoising auto-encoder

- Learn to generate original sentence from a noisy version of it
- Eliminates the learning of identity function

Corrupted data
Denoising auto-encoder

- Encoder representation is the representation for noisy sentence.
- Decoder tries to generate the original sentence from the encoder representation of the noisy sentence.
- A sentence can be corrupted using different types of noise:
  - Swapping of words
  - Removal of words
  - Replacement of words with other words.
Back-Translation

- Utilize monolingual data of target language
- Generate pseudo parallel data using MT system in opposite direction (target->source)

Train MT system (L1->L2) using a combination of parallel and generated synthetic data both

Iterative Back-Translation

Train $MT_{L2\rightarrow L1}$ using $D$

Generate synthetic data (SD) for $MT_{L1\rightarrow L2}$ using $MT_{L2\rightarrow L1}$

$D = D \cup SD$

Generate synthetic data (SD) for $MT_{L2\rightarrow L1}$ using $MT_{L1\rightarrow L2}$

$D = D \cup SD$

Train $MT_{L1\rightarrow L2}$ using $D$

$D = Parallel \ corpus$

SD = Synthetic data
Iterative Back-Translation

- Beneficial for Low resource languages

---

Language model pretraining for Unsupervised NMT
General Framework

L1 Monolingual Corpus

Language Model
Pre-Training

Unsupervised
NMT Fine Tuning

L2 Monolingual Corpus

Pre-Training

Fine-Tuning
mBERT (Devlin et al., 2019)

- Unsupervised pre-training
- Transfer learning with language models
- MLM (Masked language modeling objective) + Next sentence prediction
- Fine-tuned for language understanding and question answering tasks

MuRIL (Khanuja et al., 2021)

- Multilingual LM for Indic languages (16 indic languages and english).
- Transliterated data (Native → Latin)
- MLM (Masked language modeling) + TLM (Translation language modeling) objective
- Fine-tuned for language understanding and question answering tasks
XLM Fine Tuning

- Perform fine-tuning using
  - Iterative back-translation
  - Denoising auto-encoding
- Alternate between the two objective
- Denoising auto-encoding helps in better training of the decoder
MASS Pre-Training (Song et al., 2019)

- Perform fine-tuning using: Iterative back-translation

BART Pretraining

- Trained by
  - Corrupting text with an arbitrary noising function
  - Learning a model to reconstruct the original text.
- Denoising full text
- Multi-sentence level

My name is John. I go to school daily.

<table>
<thead>
<tr>
<th>Method</th>
<th>Original Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token Masking</td>
<td>My _ is John. I __ school daily.</td>
</tr>
<tr>
<td>Token deletion</td>
<td>My name John. I go to daily.</td>
</tr>
<tr>
<td>Text infilling</td>
<td>My _ John. I go _.</td>
</tr>
<tr>
<td>Sentence permutation</td>
<td>I go to school daily. My name is John</td>
</tr>
<tr>
<td>Document rotation</td>
<td>name is John. I go to school daily. my</td>
</tr>
</tbody>
</table>
mBART (Liu et al 2020)

- A sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages using the BART objective
- Unsupervised NMT
  - BART pretraining using monolingual corpora of multiple languages + Iterative Back-Translation

mBART (Liu et al 2020)

- Pre-training using BART objective on multiple languages

<table>
<thead>
<tr>
<th>Model</th>
<th>Similar Pairs</th>
<th>Dissimilar Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>En-De</td>
<td>En-Ro</td>
</tr>
<tr>
<td></td>
<td>←</td>
<td>→</td>
</tr>
<tr>
<td>Random</td>
<td>21.0</td>
<td>17.2</td>
</tr>
<tr>
<td>XLM (2019)</td>
<td>34.3</td>
<td>26.4</td>
</tr>
<tr>
<td>MASS (2019)</td>
<td>35.2</td>
<td>28.3</td>
</tr>
<tr>
<td>mBART</td>
<td>34.0</td>
<td>29.8</td>
</tr>
</tbody>
</table>

- En-De and En-ro are only trained using specified source and target languages
- En-Ne and En-Si, the pretraining is performed using mBART on 25 languages.
- mBART also generalizes well for the languages not seen in pretraining.

Results: mBART (only on source and target language) pretraining for unsupervised NMT

Unsupervised NMT for Indic Languages
Motivation

• Unsupervised NMT for Indic languages
• There is lot of cognate overlap between some of the Indic language pairs
• Baseline UNMT (MASS pretraining + Iterative back-translation)
  – low performance for language-pairs with low lexical overlap.
Data

- Monolingual data: AI4Bharat\(^1\)
- Test and validation data: Combination of ILCI\(^2\) and WAT 2020 multi-indic-mt\(^3\) task data (only common sentences are fetched to create validation and test data for indic-indic language pairs)

<table>
<thead>
<tr>
<th>Language</th>
<th>Number of Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengali</td>
<td>7.21 M</td>
</tr>
<tr>
<td>Gujarati</td>
<td>7.89 M</td>
</tr>
<tr>
<td>Hindi</td>
<td>63.00 M</td>
</tr>
<tr>
<td>Malayalam</td>
<td>9.93 M</td>
</tr>
<tr>
<td>Marathi</td>
<td>11.70 M</td>
</tr>
<tr>
<td>Tamil</td>
<td>21.00 M</td>
</tr>
<tr>
<td>Telugu</td>
<td>15.20 M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Language-pair</th>
<th>Size of test data</th>
<th>Size of validation data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ILCI</td>
<td>WAT</td>
</tr>
<tr>
<td>bn - hi</td>
<td>2000</td>
<td>0</td>
</tr>
<tr>
<td>gu - hi</td>
<td>2000</td>
<td>1403</td>
</tr>
<tr>
<td>ml - hi</td>
<td>2000</td>
<td>869</td>
</tr>
<tr>
<td>mr - hi</td>
<td>2000</td>
<td>1098</td>
</tr>
<tr>
<td>ta - hi</td>
<td>2000</td>
<td>1129</td>
</tr>
<tr>
<td>te - hi</td>
<td>2000</td>
<td>851</td>
</tr>
</tbody>
</table>

\(^1\) https://indicnlp.ai4bharat.org/corpora/
\(^3\) http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/
Approaches

- The lexical divergence between source and target languages play a big role in the success of UNMT.
- We explore following approaches
  - Baseline unsupervised NMT (MASS pretraining + iterative back-translation)
  - Script conversion (Transliteration to a common script)
  - Unsupervised bilingual embedding based initialization to bring the vocabulary of the two languages closer
  - Dictionary word substitution using a bilingual dictionary.
    - Randomly replace whole words in the sentence with the corresponding word translation obtained from a ground truth bilingual dictionary as a preprocessing step
## Results: Language-pairs with High Lexical Overlap

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → bn</td>
<td>bn → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>0.94</td>
<td>1.43</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>0.62</td>
<td>0.82</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>0.38</td>
<td>0.78</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>0.35</td>
<td>0.54</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>0.95</td>
<td>1.59</td>
</tr>
<tr>
<td>Transliteration (T)</td>
<td>5.60</td>
<td>7.53</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings + T</td>
<td>7.60</td>
<td>10.65</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training + T</td>
<td>4.78</td>
<td>8.58</td>
</tr>
<tr>
<td>Code Switching Pre-training + T</td>
<td>2.70</td>
<td>3.46</td>
</tr>
<tr>
<td>Dictionary Word Substitution + T</td>
<td>8.34</td>
<td>12.16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → gu</td>
<td>gu → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>11.36</td>
<td>12.99</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>16.5</td>
<td>19.57</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>6.92</td>
<td>9.57</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>2.81</td>
<td>2.31</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>9.83</td>
<td>11.14</td>
</tr>
<tr>
<td>Transliteration (T)</td>
<td>21.72</td>
<td>26.01</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings + T</td>
<td>21.72</td>
<td>26.01</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training + T</td>
<td>17.77</td>
<td>24.33</td>
</tr>
<tr>
<td>Code Switching Pre-training + T</td>
<td>12.92</td>
<td>14.54</td>
</tr>
<tr>
<td>Dictionary Word Substitution + T</td>
<td>26.38</td>
<td>33.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → mr</td>
<td>mr → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>9.49</td>
<td>15.89</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>10.85</td>
<td>18.57</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>7.86</td>
<td>12.79</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>5.91</td>
<td>7.53</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>12.11</td>
<td>20.37</td>
</tr>
</tbody>
</table>
Results: Language-pairs with Low Lexical Overlap

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → ml</td>
<td>ml → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>0.00</td>
<td>0.44</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>0.15</td>
<td>0.79</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>0.00</td>
<td>0.21</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>0.00</td>
<td>0.39</td>
</tr>
<tr>
<td>Transliteration (T)</td>
<td>0.46</td>
<td>2.25</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings + T</td>
<td>0.95</td>
<td>3.66</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training + T</td>
<td>0.2</td>
<td>1.28</td>
</tr>
<tr>
<td>Code Switching Pre-training + T</td>
<td>0.17</td>
<td>0.43</td>
</tr>
<tr>
<td>Dictionary Word Substitution + T</td>
<td>1.04</td>
<td>1.04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → ta</td>
<td>ta → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>0.22</td>
<td>0.59</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>0.47</td>
<td>0.08</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>0.39</td>
<td>0.67</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>0</td>
<td>0.39</td>
</tr>
<tr>
<td>Transliteration (T)</td>
<td>0.43</td>
<td>1.15</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings + T</td>
<td>0.92</td>
<td>2.46</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training + T</td>
<td>1.67</td>
<td>2.54</td>
</tr>
<tr>
<td>Code Switching Pre-training + T</td>
<td>0.24</td>
<td>0.44</td>
</tr>
<tr>
<td>Dictionary Word Substitution + T</td>
<td>0.94</td>
<td>2.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>CHRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hi → te</td>
<td>te → hi</td>
</tr>
<tr>
<td>Original Script</td>
<td>0.46</td>
<td>1.35</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings</td>
<td>1.19</td>
<td>2.50</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training</td>
<td>0.38</td>
<td>0.7</td>
</tr>
<tr>
<td>Code Switching Pre-training</td>
<td>0.21</td>
<td>0.48</td>
</tr>
<tr>
<td>Dictionary Word Substitution</td>
<td>0.38</td>
<td>0.68</td>
</tr>
<tr>
<td>Transliteration (T)</td>
<td>1.67</td>
<td>3.45</td>
</tr>
<tr>
<td>Bilingual BPE Embeddings + T</td>
<td>3.11</td>
<td>6.9</td>
</tr>
<tr>
<td>Anchored Cross-lingual Pre-training + T</td>
<td>1.71</td>
<td>2.38</td>
</tr>
<tr>
<td>Code Switching Pre-training + T</td>
<td>0.87</td>
<td>1.43</td>
</tr>
<tr>
<td>Dictionary Word Substitution + T</td>
<td>3.38</td>
<td>7.8</td>
</tr>
</tbody>
</table>
Lexical Similarity versus BLEU score

- The correlation between lexical similarity between source and reference sentences and BLEU score for Gujarati → Hindi.
Lexical Similarity versus BLEU score

- The correlation between lexical similarity between source and reference sentences and BLEU score for Hindi → Malayalam.
- r value with original script is very low.
## Shared Vocabulary

<table>
<thead>
<tr>
<th>Lang-pair (src - tgt)</th>
<th>% source tokens present in target</th>
<th>% target tokens present in source</th>
<th>% source tokens present in target</th>
<th>% target tokens present in source</th>
</tr>
</thead>
<tbody>
<tr>
<td>bn - hi</td>
<td>4.04</td>
<td>1.21</td>
<td>13.65</td>
<td>4.07</td>
</tr>
<tr>
<td>gu - hi</td>
<td>2.86</td>
<td>1.31</td>
<td>19.43</td>
<td>8.88</td>
</tr>
<tr>
<td>ml - hi</td>
<td>0.0</td>
<td>0.0</td>
<td>1.1</td>
<td>1.82</td>
</tr>
<tr>
<td>mr - hi</td>
<td>14.15</td>
<td>7.11</td>
<td>14.15</td>
<td>7.11</td>
</tr>
<tr>
<td>ta - hi</td>
<td>0.0</td>
<td>0.0</td>
<td>0.62</td>
<td>1.09</td>
</tr>
<tr>
<td>te - hi</td>
<td>0.45</td>
<td>0.36</td>
<td>3.65</td>
<td>2.87</td>
</tr>
</tbody>
</table>

Statistics on Lexical overlap between the two languages

223
### Qualitative Analysis

- Fluent but inadequate translations

<table>
<thead>
<tr>
<th>Malayalam (Devanagari)</th>
<th>Source (English meaning)</th>
<th>कश्चिं कस्मिस्तं भिख्या भिक्षणं कथिकाकावृत्तेऽविलम्पपू। Take as much as you can eat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>हम उतना ही लें, जितना खाना है</td>
<td></td>
</tr>
<tr>
<td>Translation by Original Script (Gloss)</td>
<td>पहलवान ने बेदर को निकाल डाला। wrestler has waiter removed</td>
<td></td>
</tr>
<tr>
<td>Translation by Transliteration (Gloss)</td>
<td>करने जा रहे हैं, ये चिंता बारीकी। to-do go are is these thinking closely</td>
<td></td>
</tr>
<tr>
<td>Translation by Bilingual BPE Embeddings + T (Gloss)</td>
<td>भोजन करने लगे, ये बरतन food to-do these utensils</td>
<td></td>
</tr>
<tr>
<td>Translation by Dictionary Word Substitution + T (Gloss)</td>
<td>खाने के लिए ये फल eating of for these fruits</td>
<td></td>
</tr>
</tbody>
</table>
Qualitative Analysis

- Meaning drift due to transliteration

| Bengali | Source (Devanagari) | হেল্ডি লাইফ স্টাইল মেনে চলুন।
|         | (English meaning)   | হেল্ডি লাইফ স্টাইল মেনে চলুন।
|         |                     | Adopt a healthy life-style
|         | Reference           | হেল্ডি লাইফ স্টাইল অপনাএ ।
| Hindi   | Translation by Original Script (Gloss) | লাউটসবার্গ ফ্রেম কেয়ার শুরু, তহলকা
|         | lotsburg frame care start, panic       |
|         | Translation by Transliteration (Gloss) | হেল্ডি লাইফ স্টাইল কে লিএ চল রহে হেঁ:
|         | healthy life-style for walking are     |
|         | Translation by Bilingual BPE Embeddings + T (Gloss) | হেল্ডি লাইফ স্টাইল ভি চল রহে হেঁ ...
|         | healthy life-style too walk doing is   |
|         | Translation by Dictionary Word Substitution + T (Gloss) | হেল্ডি লাইফ স্টাইল কে লিএ চল রহে হেঁ
|         | healthy life style for walking are     |

- **চলুন** (Chaluna) → to go, **মন চলুন** (mene chaluna) → maintain
- The Hindi translation for the verb chaluna is **चलो** (chalo).
  [This happens because of transliteration and bilingual embeddings]
Summary

• Current state of the art approaches in unsupervised NMT
• Analysis of Unsupervised NMT for Indic languages
  – The lexical divergence between source and target language plays an important role.
  – 3 approaches to bridge lexical divergence between source and target languages
    • Script conversion
    • Initialization using bilingual embeddings
    • Dictionary word substitution
Machine Translation Demonstration
Automatic Post Editing
Outline

1. Motivation
2. Problem Statement
3. Challenges
4. Categorization of APE Systems
5. APE Paradigms
6. WMT APE Shared Tasks
7. HW-TSC’s APE System
8. Experiments and Results
Motivation

• Machine Translation (MT) systems: far from perfect
• Requirement of post-processing through human intervention
  – Generation of parallel data (mt_op <--> post-edited mt_op)
• Can we automate the post-processing phase using this data?
• Use cases:
  – Ideal: to eliminate the need of human involvement.
  – Black-box scenario: To further improve translations by identifying and correcting recurring MT errors
  – Adapt terminologies for a specific domain
Problem Statement

• Automatic Post Editing (APE): Given the translations generated by a machine translation system, generate corrected versions of them which are publishable.
  – The edits should be minimal.

• In a supervised setting, training data contains triplets:
  • Source sentence: People can get COVID-19 even after vaccination.
  • MT translation: लसीकरणानंतरही लोकांना कोविड - 19 मिळू शकतो .
  • Human post-edited version: लसीकरणानंतरही लोकांना कोविड - 19 होऊ शकतो .

• Input: MT translation, Output: Human post-edited version
Categorization of APE systems

- APE Systems can be categorized as follows:
  - Accessibility of MT System: Black-box or Glass-box
  - Type of Post-editing Data: Real or Synthetic
  - Domain of the Data: General or Specific

- We focus on:
  - Black-box scenario
  - Real as well as Synthetic Data
  - Domain Specific APE systems
Challenges (1/2)

• Data :
  – Deep Learning based Methods: data-hungry
  – Data Sparsity
  – Increased complexity: Same error can be corrected in more than one way
  – Coverage of error-correction patterns
  – Requirement of new datasets
Challenges (2/2)

• Technology:
  – Neural APE systems: Follow similar trend as MT
  – Joint modelling of SRC and MT_OP helps, but increases data sparsity
  – Poses a risk of unnecessary edits
  – Issue of overfitting
  – Requirement: Techniques resilient to problem of over-correction, and can work in low-resource settings
APE Paradigms (1/2)

• APE task: a monolingual translation task
  – The same MT technology has been used for APE
• Rule-based APE:
  – Not much work done
  – Uses precise PE rules
  – The rules might not be capturing all possible scenarios
  – Not portable across domains
APE Paradigms (2/2)

• Phrase-based APE:
  – Dominated the APE field for a few years
  – Showed significant improvements when underlying MT system was rule-based
  – Limited improvements when underlying MT system was SMT

• Neural APE:
  – Current-state-of-the-art
  – Showed significant improvements when underlying MT system is SMT
Terminologies

- SRC: source language sentence
- MT_OP: translation of SRC generated using a MT system
- MT_REF: reference target language sentence for the SRC
- PE_REF: Human post-edited version of MT_OP
- PE_OP: Output generated by the APE system
- Triplet: (SRC, MT_OP, PE_REF)
- Example:
  - SRC: People can get COVID-19 even after vaccination.
  - MT_OP: लसीकरणानंतरही लोकांना कोविड - 19 मिळू शकतो .
  - PE_REF: लसीकरणानंतरही लोकांना कोविड - 19 होऊ शकतो .
WMT APE Shared Tasks
WMT APE Shared Task

- WMT APE Shared Task is hosted every year since 2015
- Considers the black-box scenario
- APE data is shared across participants
  - Train, development data: (SRC, MT_OP, PE_REF)
  - Test data: (SRC, MT_OP)
- Participants submit PE_OP for the test set
- The submitted systems are evaluated over the benchmark datasets
- Evaluation metrics used for rankings
  - TER and BLEU
WMT APE Shared Task

• Baseline APE systems:
  – “do nothing” APE system
    • Does not make any modification to MT_OP,
      i.e. PE_OP = MT_OP
• Participants are allowed to use any external data.
• Optional resources - Synthetic corpora:
  – Artificial Corpus (4 Million triplets) [6]
  – eSCAPE corpus (14.4 Million triplet) [7]
WMT15 APE Shared Task

- Pilot round
- Language pair: English - Spanish
- Domain: News
- Data: Train (12,000), Development (1,000), Test (2,000) triplets
- PE_REF was collected through crowd-sourcing
- Number of submissions: 7
- All the submissions were based on phrase-based technology
- None of the submissions beat the baseline
WMT15 APE Shared Task

- Difference between
  - the baseline and top ranked system: -0.315 TER points
  - the phrase based APE and top ranked system: 0.926

- It was hypothesized that the poor results are due to origin, quantity and domain of the data.

Results:

<table>
<thead>
<tr>
<th>ID</th>
<th>Avg. TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>22.913</td>
</tr>
<tr>
<td>FBK Primary</td>
<td>23.228</td>
</tr>
<tr>
<td>LIMSI Primary</td>
<td>23.331</td>
</tr>
<tr>
<td>USAAR-SAPE</td>
<td>23.426</td>
</tr>
<tr>
<td>LIMSI Contrastive</td>
<td>23.573</td>
</tr>
<tr>
<td>Abu-MaTran Primary</td>
<td>23.639</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>23.649</td>
</tr>
<tr>
<td>(Simard et. al) [124]</td>
<td>23.839</td>
</tr>
<tr>
<td>Abu-MaTran Contrastive</td>
<td>24.715</td>
</tr>
</tbody>
</table>
WMT16 APE Shared Task

- Language pair: English - German
- Domain: IT domain
- Data: Train (12,000), Development (1,000), Test(2,000) triplets
- PE_REF was collected through professional post-editors
- Number of submissions: 11
- Except two, rest of the submissions were based on phrase-based technology
- 7 teams crossed the baseline.
- The top ranked system used
  - RNN based encoder-decoder model with attention
  - Artificially generated Data of around 4 million triplets (Using Back Translation)
WMT16 APE Shared Task

• Difference between
  – the baseline and top ranked system: 3.24 TER points
  – the phrase based APE and top ranked system: 3.12
• Have the results improved due to change of data or due to technology shift?

Results:

<table>
<thead>
<tr>
<th>ID</th>
<th>Avg. TER (↓)</th>
<th>BLEU (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMU Primary</td>
<td>21.52</td>
<td>67.65</td>
</tr>
<tr>
<td>AMU Contrastive</td>
<td>23.06</td>
<td>66.09</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>23.92</td>
<td>64.75</td>
</tr>
<tr>
<td>FBK Primary</td>
<td>23.94</td>
<td>64.75</td>
</tr>
<tr>
<td>USAAR Primary</td>
<td>24.14</td>
<td>64.10</td>
</tr>
<tr>
<td>USAAR Contrastive</td>
<td>24.14</td>
<td>64.00</td>
</tr>
<tr>
<td>CUNI Primary</td>
<td>24.31</td>
<td>63.32</td>
</tr>
<tr>
<td>(Simard et al.)[124]</td>
<td>24.64</td>
<td>63.47</td>
</tr>
<tr>
<td>Baseline</td>
<td>24.76</td>
<td>62.11</td>
</tr>
<tr>
<td>DCU Contrastive</td>
<td>26.79</td>
<td>58.60</td>
</tr>
<tr>
<td>JUSAAR Primary</td>
<td>26.92</td>
<td>59.44</td>
</tr>
<tr>
<td>JUSAAR Contrastive</td>
<td>26.97</td>
<td>59.18</td>
</tr>
<tr>
<td>DCU Primary</td>
<td>28.97</td>
<td>55.19</td>
</tr>
</tbody>
</table>
WMT17 APE Shared Task

• Language pairs and domain:
  – English - German (IT)
  – German - English (Medical)
• Data for English-German: Train (11,000), Test(2,000) triplets
• Data for German-English: Train(25,000), Dev(1,000), Test(2,000)
• Number of submissions: 15 (English-German), 5(German-English)
• Most of the submissions were based on neural approaches, and beat the baseline except one system.
• The top ranked system followed neural APE approach:
  – Used multi-source APE System
  – Trained the system over synthetic corpus and fine-tuned on the provided in-domain data
WMT17 APE Shared Task

- Difference between
  - the baseline and top ranked system: 4.88 TER points
  - the phrase based APE and top ranked system: 5.09 TER points

- Results (English-German):

<table>
<thead>
<tr>
<th>ID</th>
<th>Avg. TER (↓)</th>
<th>BLEU (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBK Primary</td>
<td>19.6</td>
<td>70.07</td>
</tr>
<tr>
<td>AMU Primary</td>
<td>19.77</td>
<td>69.5</td>
</tr>
<tr>
<td>AMU Contrastive</td>
<td>19.83</td>
<td>69.38</td>
</tr>
<tr>
<td>DCU Primary</td>
<td>20.11</td>
<td>69.19</td>
</tr>
<tr>
<td>DCU Contrastive</td>
<td>20.25</td>
<td>69.33</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>20.3</td>
<td>69.11</td>
</tr>
<tr>
<td>FBK_USAAR Contr.</td>
<td>21.55</td>
<td>67.28</td>
</tr>
<tr>
<td>USAAR Primary</td>
<td>23.05</td>
<td>65.01</td>
</tr>
<tr>
<td>LIG Primary</td>
<td>23.22</td>
<td>65.12</td>
</tr>
<tr>
<td>JXNU Primary</td>
<td>23.31</td>
<td>65.66</td>
</tr>
<tr>
<td>LIG Contrastive-Forced</td>
<td>23.51</td>
<td>64.52</td>
</tr>
<tr>
<td>LIG Contrastive-Chained</td>
<td>23.66</td>
<td>64.46</td>
</tr>
<tr>
<td>CUNI Primary</td>
<td>24.03</td>
<td>64.28</td>
</tr>
<tr>
<td>USAAR Contrastive</td>
<td>24.17</td>
<td>63.55</td>
</tr>
<tr>
<td>Baseline</td>
<td>24.48</td>
<td>62.49</td>
</tr>
<tr>
<td>(Simard et al.)[124]</td>
<td>24.69</td>
<td>62.97</td>
</tr>
<tr>
<td>CUNI Contrastive</td>
<td>25.94</td>
<td>61.65</td>
</tr>
</tbody>
</table>

Image Source: [2]
WMT17 APE Shared Task

- Difference between
  - the baseline and top ranked system: 0.26 TER points
  - the phrase based APE and top ranked system: 0.45 TER points
- Difference between the baseline and phrase based APE can be seen as an indicator of task difficulty level.

Results (German-English):

<table>
<thead>
<tr>
<th>ID</th>
<th>Avg. TER (↓)</th>
<th>BLEU (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBK Primary</td>
<td>15.29</td>
<td>79.82</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>15.31</td>
<td>79.64</td>
</tr>
<tr>
<td>LIG Primary</td>
<td>15.53</td>
<td>79.49</td>
</tr>
<tr>
<td>Baseline</td>
<td>15.55</td>
<td>79.54</td>
</tr>
<tr>
<td>LIG Contrastive-Forced</td>
<td>15.62</td>
<td>79.48</td>
</tr>
<tr>
<td>LIG Contrastive-Chained</td>
<td>15.68</td>
<td>79.35</td>
</tr>
<tr>
<td>(Simard et al.)[124]</td>
<td>15.74</td>
<td>79.28</td>
</tr>
</tbody>
</table>

Image Source: [2]
WMT18 APE Shared Task

- Language pairs and domain: English - German (IT)
- Two different MT systems: PBMT, NMT
- Data for PBMT: datasets released in earlier rounds
- Data for NMT: Train (13,442), Dev (1,000), Test (1,023)
- Number of submissions: 11 (PBMT), 10 (NMT)
- All the submissions were based on neural approaches
- This allowed to compare the effectiveness of neural APE systems
- The top ranked system:
  - Used multi-source APE System and transformer architecture
  - Trained the system over large synthetic corpus and fine-tuned on the provided in-domain data
WMT18 APE Shared Task

- Difference between the baseline and top ranked system: 6.24 TER points

- Results (PBMT):

<table>
<thead>
<tr>
<th>ID</th>
<th>TER (pe)</th>
<th>BLEU (pe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS_UEdin Primary</td>
<td>18.0</td>
<td>72.52</td>
</tr>
<tr>
<td>FBK Contrastive (MRT+MLE)</td>
<td>18.62</td>
<td>71.04</td>
</tr>
<tr>
<td>FBK Primary (MRT)</td>
<td>18.94</td>
<td>71.22</td>
</tr>
<tr>
<td>POSTECH Contrastive (fix5)</td>
<td>19.63</td>
<td>69.87</td>
</tr>
<tr>
<td>POSTECH Primary</td>
<td>19.72</td>
<td>69.8</td>
</tr>
<tr>
<td>POSTECH Contrastive (var5)</td>
<td>19.74</td>
<td>69.7</td>
</tr>
<tr>
<td>USAAR_DFKI Primary</td>
<td>22.69</td>
<td>66.16</td>
</tr>
<tr>
<td>USAAR_DFKI*</td>
<td>22.88</td>
<td>66.05</td>
</tr>
<tr>
<td>DFKI-MLT Primary (Transf.large)</td>
<td>24.19†</td>
<td>63.4</td>
</tr>
<tr>
<td>Baseline</td>
<td>24.24</td>
<td>62.99</td>
</tr>
<tr>
<td>DFKI-MLT Contrastive (Transf.base)</td>
<td>24.5†</td>
<td>62.78†</td>
</tr>
<tr>
<td>DFKI-MLT Contrastive (LSTM)</td>
<td>25.3</td>
<td>62.1</td>
</tr>
</tbody>
</table>

Image Source: [3]
WMT18 APE Shared Task

• Difference between the baseline and top ranked system: 0.38 TER points

• Results (NMT):

<table>
<thead>
<tr>
<th>ID</th>
<th>TER (pe)</th>
<th>BLEU (pe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBK Primary (MRT)</td>
<td>16.46</td>
<td>75.53</td>
</tr>
<tr>
<td>MS._UEdin Primary</td>
<td>16.5</td>
<td>75.44</td>
</tr>
<tr>
<td>FBK Contrastive (MRT+MLE)</td>
<td>16.55</td>
<td>75.38</td>
</tr>
<tr>
<td>POSTECH Contrastive (top1)</td>
<td>16.7†</td>
<td>75.14</td>
</tr>
<tr>
<td>POSTECH Primary (fix5)</td>
<td>16.71†</td>
<td>75.13</td>
</tr>
<tr>
<td>POSTECH Contrastive (var5)</td>
<td>16.71†</td>
<td>75.2</td>
</tr>
<tr>
<td>Baseline</td>
<td>16.84</td>
<td>74.73</td>
</tr>
<tr>
<td>USAAR._DFKI Primary</td>
<td>17.23</td>
<td>74.22</td>
</tr>
<tr>
<td>DFKI-MLT Contrastive (Transf.base)</td>
<td>18.84</td>
<td>70.87</td>
</tr>
<tr>
<td>DFKI-MLT Primary (Transf.large)</td>
<td>18.86</td>
<td>70.98</td>
</tr>
<tr>
<td>DFKI-MLT Contrastive (LSTM)</td>
<td>19.88</td>
<td>69.35</td>
</tr>
</tbody>
</table>

Image Source: [3]
Findings:

- The difficulty of the APE task: proportional to quality of machine translation system.
- It is easier to improve translations generated from a PBMT system, using neural APE. But, this is not the case when underlying MT is the NMT system.
- Systems used the eSCAPE and Artificial corpora. The good improvements in results can be attributed to use of huge amount of synthetic data, along with the technology developments.
WMT19 APE Shared Task

• Language pairs and domain: English-German, English-Russian (IT)
• Underlying MT system: Neural
• Data for English-German: datasets released in earlier rounds
• Data for English-Russian: Train (15,089), Dev (1,000), Test (1,023)
• Number of submissions: 18 (English-German), 4 (English-Russian)
• The quality of English-Russian MT system was much higher: 76.20 BLEU score. None of the systems beat the baseline.
• The top ranked system in the English-german task:
  – Explored transfer learning: Adapted BERT-based encoder-decoder model to APE
  – Used the shared encoder
WMT19 APE Shared Task

- Difference between
  - the baseline and top ranked system: 0.78 TER points

- Results (English-German):

<table>
<thead>
<tr>
<th>ID</th>
<th>TER (pe)</th>
<th>BLEU (pe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNBABEL Primary</td>
<td>16.06*</td>
<td>75.96</td>
</tr>
<tr>
<td>POSTECH Primary</td>
<td>16.11*</td>
<td>76.22</td>
</tr>
<tr>
<td>POSTECH Contrastive (var2Ens8)</td>
<td>16.13*</td>
<td>76.21</td>
</tr>
<tr>
<td>USAAR_DFKI Primary</td>
<td>16.15*</td>
<td>75.75</td>
</tr>
<tr>
<td>POSTECH Contrastive (top1Ens4)</td>
<td>16.17*</td>
<td>76.15</td>
</tr>
<tr>
<td>UNBABEL Contrastive (2)</td>
<td>16.21*</td>
<td>75.7</td>
</tr>
<tr>
<td>UNBABEL Contrastive (1)</td>
<td>16.24*</td>
<td>75.7</td>
</tr>
<tr>
<td>FBK Primary</td>
<td>16.37*</td>
<td>75.71</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>16.61†</td>
<td>75.28</td>
</tr>
<tr>
<td>UDS Primary</td>
<td>16.77†</td>
<td>75.03</td>
</tr>
<tr>
<td>IC_USFD Contrastive</td>
<td>16.78†</td>
<td>74.88</td>
</tr>
<tr>
<td>UDS Contrastive (Gaus)</td>
<td>16.79†</td>
<td>75.03</td>
</tr>
<tr>
<td>UDS Contrastive (Uni)</td>
<td>16.80†</td>
<td>75.03</td>
</tr>
<tr>
<td>IC_USFD Primary</td>
<td>16.84†</td>
<td>74.8‡</td>
</tr>
<tr>
<td>Baseline</td>
<td>16.84</td>
<td>74.73</td>
</tr>
<tr>
<td>ADAPT_DCU Contrastive (SMT)</td>
<td>17.07</td>
<td>74.3</td>
</tr>
<tr>
<td>ADAPT_DCU Primary</td>
<td>17.29</td>
<td>74.29</td>
</tr>
<tr>
<td>USAAR_DFKI Contrastive</td>
<td>17.31</td>
<td>73.97</td>
</tr>
<tr>
<td>ADAPT_DCU Contrastive (LEN)</td>
<td>17.41</td>
<td>74.01</td>
</tr>
</tbody>
</table>

Image Source: [4]
WMT19 APE Shared Task

- Difference between the baseline and top ranked system: -0.43 TER points

- Results (English-Russian):

<table>
<thead>
<tr>
<th>ID</th>
<th>TER (pe)</th>
<th>BLEU (pe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>16.16</td>
<td>76.2</td>
</tr>
<tr>
<td>ADAPT_DCU Contrastive</td>
<td>16.59</td>
<td>75.27</td>
</tr>
<tr>
<td>ADAPT_DCU Primary</td>
<td>18.31</td>
<td>72.9</td>
</tr>
<tr>
<td>FBK Primary</td>
<td>19.34</td>
<td>72.42</td>
</tr>
<tr>
<td>FBK Contrastive</td>
<td>19.48</td>
<td>72.91</td>
</tr>
</tbody>
</table>
WMT19 APE Shared Task

• Findings:
  – The same test as previous year was used for the English-German subtask. Four systems were able to beat the top system of the last year.
  – The quality of translations provided for the English-Russian task was very high. Also, around 60% of the sentences required no edit. This made the problem more challenging.
  – This highlights the problem of over-corrections.
WMT20 APE Shared Task

- Language pairs and domain: English-German, English-Chinese (Wikipedia)
- Data sizes: Train (7,000), Dev (1,000), Test (1,000)
- Number of submissions: 11 (English-German), 4 (English-Chinese)
- Tradeoff: general domain vs lower quality translations
- The top ranked system:
  - Followed the architecture of the last year’s winning team
  - Instead of using models like BERT, pretrained a NMT model
  - Used bottleneck adapter layers to prevent overfitting
  - Used external MT candidates. So, the input to APE looked like (SRC, MT_OP, EXT_MT_OP)
WMT20 APE Shared Task

- Difference between the baseline and top ranked system: 11.35 TER points

<table>
<thead>
<tr>
<th>Language</th>
<th>System Name</th>
<th>TER</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-de</td>
<td>HW-TSC_DIRECT_CONTRASTIVE.pe</td>
<td>20.21</td>
<td>66.89</td>
</tr>
<tr>
<td></td>
<td>HW-TSC_CONCAT_PRIMARY.pe</td>
<td>20.52</td>
<td>66.16</td>
</tr>
<tr>
<td></td>
<td>MinD-mem.enc_dec_post_CONTRASTIVE</td>
<td>26.99</td>
<td>55.77</td>
</tr>
<tr>
<td></td>
<td>POSTECH-ETRL_XLM-Top4Ens_CONTRASTIVE</td>
<td>27.02</td>
<td>56.37</td>
</tr>
<tr>
<td></td>
<td>MinD-mem.enc_dec_PRIMARY</td>
<td>27.03</td>
<td>55.58</td>
</tr>
<tr>
<td></td>
<td>POSTECH-ETRL_XLM-Top3Ens_PRIMARY</td>
<td>27.37</td>
<td>55.83</td>
</tr>
<tr>
<td></td>
<td>BeringLab_model1_PRIMARY</td>
<td>27.61</td>
<td>54.71</td>
</tr>
<tr>
<td></td>
<td>BeringLab_model2_CONTRASTIVE</td>
<td>27.96</td>
<td>54.60</td>
</tr>
<tr>
<td></td>
<td>POSTECH_TERNoi-7Fold-Ens8_CONTRASTIVE</td>
<td>28.22</td>
<td>54.51</td>
</tr>
<tr>
<td></td>
<td>POSTECH_TERNoi-8Fold-Ens8_PRIMARY</td>
<td>28.41</td>
<td>54.22</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>31.56</td>
<td>50.21</td>
</tr>
<tr>
<td></td>
<td>KAISTxPAPAGO_EMT_PRIMARY</td>
<td>32.00</td>
<td>49.21</td>
</tr>
</tbody>
</table>
WMT20 APE Shared Task

• Difference between
  – the baseline and top ranked system: 12.13 TER points

• Results (English-Chinese):

<table>
<thead>
<tr>
<th>Model Name</th>
<th>TER</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW-TSC_CONCAT_PRIMARY.pe</td>
<td>47.36</td>
<td>37.69</td>
</tr>
<tr>
<td>HW-TSC_DIRECT_CONTRASTIVE.pe</td>
<td>48.01</td>
<td>37.32</td>
</tr>
<tr>
<td>POSTECH-ETRI_XLM-Top3Ens_PRIMARY</td>
<td>54.92</td>
<td>28.90</td>
</tr>
<tr>
<td>POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE</td>
<td>55.08</td>
<td>28.97</td>
</tr>
<tr>
<td>Baseline</td>
<td>59.49</td>
<td>23.12</td>
</tr>
</tbody>
</table>

Image Source: [5]
WMT21 APE Shared Task

- Language pairs and domain: English-German, English-Chinese (Wikipedia)
- Data sizes: Train (7,000), Dev (1,000), Test (1,000)
- The data is re-translated and so re-post-edited to improve the quality.
- Number of submissions: 4 (English-German)
- Results:

<table>
<thead>
<tr>
<th>ID</th>
<th>TER</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netmarble_Contrastive</td>
<td>17.28</td>
<td>71.55</td>
</tr>
<tr>
<td>PVIE_Contrastive</td>
<td>17.74</td>
<td>70.54</td>
</tr>
<tr>
<td>PVIE_Primary</td>
<td>17.85</td>
<td>70.50</td>
</tr>
<tr>
<td>Netmarble_Primary</td>
<td>17.97</td>
<td>70.53</td>
</tr>
<tr>
<td>Baseline</td>
<td>18.05</td>
<td>71.07</td>
</tr>
</tbody>
</table>
WMT APE Shared Tasks: Summary

- Requirement of post-edits from professional post-editors
- Domain of the data
- Type of underlying MT system and Quality of translations
- Technology Development: Utilization of more and more data
- Uncertainty about effectiveness of current neural approaches

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>En-Es</td>
<td>En-De</td>
<td>En-De</td>
<td>De-En</td>
<td>En-De</td>
<td>En-De</td>
<td>En-De</td>
<td>En-Ru</td>
<td>En-De</td>
<td>En-Zh</td>
<td>En-De</td>
<td>En-Zh</td>
</tr>
<tr>
<td>Domain</td>
<td>News</td>
<td>IT</td>
<td>IT</td>
<td>Medical</td>
<td>IT</td>
<td>IT</td>
<td>IT</td>
<td>IT</td>
<td>IT</td>
<td>IT</td>
<td>Wiki</td>
<td>Wiki</td>
</tr>
<tr>
<td>MT Type</td>
<td>PBSMT</td>
<td>PBSMT</td>
<td>PBSMT</td>
<td>PBSMT</td>
<td>PBSMT</td>
<td>NMT</td>
<td>NMT</td>
<td>NMT</td>
<td>NMT</td>
<td>NMT</td>
<td>NMT</td>
<td>NMT</td>
</tr>
<tr>
<td>Baseline TER</td>
<td>22.91</td>
<td>24.76</td>
<td>24.48</td>
<td>15.55</td>
<td>24.24</td>
<td>16.84</td>
<td>16.84</td>
<td>16.16</td>
<td>31.56</td>
<td>59.49</td>
<td>18.05</td>
<td>-</td>
</tr>
<tr>
<td>ΔTER</td>
<td>-0.32</td>
<td>3.24</td>
<td>4.88</td>
<td>0.26</td>
<td>6.24</td>
<td>0.38</td>
<td>0.78</td>
<td>-0.43</td>
<td>11.35</td>
<td>12.13</td>
<td>0.77</td>
<td>-</td>
</tr>
</tbody>
</table>

ΔTER = Baseline TER - TER of the top-ranked system
HW-TSC’s APE System

- Winner of WMT20 APE Shared Task [8].
- Used transfer learning: Fine-tuned a pre-trained NMT model on the in-domain APE data
- Used data augmentation: translated the source sentences in the APE data using Google’s MT system in order to increase diversity of features.
- To control the issue of over-fitting, ‘Bottleneck Adapter Layers’ are used.
  - It is a low-dimensional FNN layer.
HW-TSC’s APE System

- Data used for training NMT model: WMT19 news translation dataset for English-German, and WMT20 news translation dataset
- APE Data: 7000 triplets (wikipedia domain)
- Results on the development set (1000 triplets):

<table>
<thead>
<tr>
<th>System</th>
<th>En-De BLEU</th>
<th>En-De TER</th>
<th>En-Zh BLEU</th>
<th>En-Zh TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>50.37</td>
<td>31.374</td>
<td>22.62</td>
<td>60.417</td>
</tr>
<tr>
<td>+ Fine-tuning</td>
<td>59.51</td>
<td>25.941</td>
<td>31.74</td>
<td>49.257</td>
</tr>
<tr>
<td>+ External MT</td>
<td>65.72</td>
<td>20.959</td>
<td>37.37</td>
<td>47.830</td>
</tr>
<tr>
<td>+ Ensemble</td>
<td>66.96</td>
<td>20.222</td>
<td>37.83</td>
<td>46.918</td>
</tr>
<tr>
<td>Submission</td>
<td>66.89</td>
<td>20.21</td>
<td>37.69</td>
<td>47.36</td>
</tr>
</tbody>
</table>

Image Source: [8]
Experiments and Results
Experiment 1

- Goal: To compare different Neural APE approaches.
- We have an English-Marathi parallel corpora but do not have corresponding human post-edits. So we generate an artificial data which can be used to train and evaluate the APE models.
- APE Data Generation: We used two different methods to generate the artificial triplets (SRC, MT_OP, PE_REF) using the Legal domain data from the Anuvaad corpora.
  - Using MT_REF as PE_REF
    - Triplet: (SRC, MT_OP, MT_REF)
  - Using round-trip translation
    - Used Marathi-English MT system to translate MT_REF to SRC’
    - Used English-Marathi MT system to translate SRC’ to MT_OP’
    - New triplet is formed as (SRC’, MT_OP’, MT_REF)
    - Found that the translations generated using this method are noisy.
- Data size: Training: 1.5 lac, Validation: 15k, Testing: 15k
Experiment 1

- **APE Systems:**
  - Single-source:
    - Treats the APE task as a monolingual translation task.
    - Ignores the SRC. Input to the APE system is MT_OP, and produces PE_REF.
  - Multi-Source:
    - Exploits dependency of errors in translation on the source sentence and the corresponding translation.
    - Input to the system is a (SRC, MT_OP) pair, and outputs the PE_REF.
    - The method uses two separate encoders: one encodes the SRC and the other encodes the MT_OP. And then, a single decoder produces the PE_REF.
Experiment 1

- **APE Systems:**
  - Multi-source (with shared encoder)
    - This setting uses a single encoder to encode both the SRC and MT_OP.
    - It is beneficial when both languages share the vocabulary.
    - SRC and MT_OP are concatenated and passed to the encoder as a single sequence.
  - Neural Programmer Interpreter
    - Instead of following an end-to-end approach that directly generates a edited sentence from the translation, the method follows a two step approach.
      - In the first phase, the translation is mapped to a sequence of edit operations (insert, keep, delete), and then this sequence along with the translation is used to generate a post-edited sentence.
Experiment 1

<table>
<thead>
<tr>
<th>APE System</th>
<th>BLEU</th>
<th>TER</th>
<th>N-modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (No APE)</td>
<td>38.54</td>
<td>41.90</td>
<td>-</td>
</tr>
<tr>
<td>Single Source</td>
<td>34.86</td>
<td>45.55</td>
<td>50</td>
</tr>
<tr>
<td>Multi-source</td>
<td>37.25</td>
<td>42.37</td>
<td>42</td>
</tr>
<tr>
<td>Multi-source (Shared Encoder)</td>
<td>34.50</td>
<td>46.23</td>
<td>53</td>
</tr>
<tr>
<td>Neural Programmer-Interpreter</td>
<td>27.22</td>
<td>53.57</td>
<td>67</td>
</tr>
</tbody>
</table>

Note:
- Training data: 1.5L triplets
- No human post-edited data
- Underlying NMT system is trained on multi-domain data
- Used 1000 PE-REF segments to get ‘N-modified’
Experiment 1

• Analysis:
  – Not able to find any APE system-specific patterns
  – General observation: Unnecessary edits are performed by the APE systems.
  – Example (Using the Multi-source APE model)
    • SRC: Special powers in case of urgency.
    • MT_REF (and PE_REF): निकडीच्या बाबतीत विशेष अधिकार.
    • MT_OP: तातडीच्या बाबतीत विशेष शक्ती.
    • PE_OP: तातडीच्या परिस्थितीत विशेष शक्ती.
Experiment 2

- **Language Pair:** English - German
- **Data (IT Domain):**
  - Combined Training datasets from WMT16, 17, 18 APE shared task
  - Training Data: 36442 Triplets (SRC, MT_OP, PE_REF)
  - Test Data: 1023 Triplets (WMT18 APE Shared Task - NMT)
  - Synthetic Data: 1M Triplets from eSCAPE corpus
- **Models:**
  - Single Source APE: trained using the synthetic data
  - Multi Source APE: trained using the synthetic data
  - Multi Source APE: trained using the synthetic data + fine-tuned on the Combined Training data
## Experiment 2

<table>
<thead>
<tr>
<th>Model</th>
<th>TER</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (No APE)</td>
<td>16.84</td>
<td>74.73</td>
</tr>
<tr>
<td>Single-source (Synthetic)</td>
<td>20.02</td>
<td>69.38</td>
</tr>
<tr>
<td>Multi-source (Synthetic)</td>
<td>18.89</td>
<td>71.39</td>
</tr>
<tr>
<td>Multi-source (Fine-tuned)</td>
<td>16.83</td>
<td>74.73</td>
</tr>
<tr>
<td>Neural Program-Interpreter (NPI) (Synthetic)</td>
<td>19.07</td>
<td>71.10</td>
</tr>
<tr>
<td>NPI (Fine-tuned)</td>
<td>18.45</td>
<td>72.82</td>
</tr>
</tbody>
</table>
Summary

- From the review of WMT APE Shared, it is clear that the APE can be used in the black-box scenario to further improve quality of translations.
- Recent advancements show a paradigm shift in the field of APE from statistical based APE approaches to neural approaches. This has also given a push to novel ways of synthetic data generation.
- New APE techniques: utilize more and more knowledge.
  - Single-source -- Multi-source -- Pretrained Models -- Data Augmentation
- Whether APE can be used to further improve quality of translations obtained using a high quality NMT system is still unclear.
- The problem of over-correction is prominent when the underlying MT system is of high quality.
References


References


References


Speech Synthesis
Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Previous Approaches
  – Latest Developments
• Demonstration
Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Introduction
  – Previous Approaches
  – Latest Developments
• Demonstration
In this presentation, we focus on Indian languages, specifically the Marathi language.

Also we allow the system to be non-causal, if required.
Speech synthesis has the potential to improve the daily lives of many people around the world.

- It can give voice to people with speaking disabilities.
  E.g. People with damaged vocal tract

The speech synthesiser housing of famous physicist Stephen Hawking

Img source: https://en.wikipedia.org/wiki/Speech_synthesis
Motivation (2/2)

• Machines reading text would enable people with visual impairments or reading disabilities to comprehend any book they wish to read. E.g. Audiobooks

• People who are able to understand spoken language, but cannot read text, will get a chance to gain knowledge from the books by listening them. E.g. Regions in India with high illiteracy rate.
Challenges

• Availability of Data:
The best English TTS systems are trained on 24 hours of data. The currently available data for Marathi and Hindi is 4.8 hours.

• Text processing:
Marathi and Hindi are almost phonemic languages. However, currently there do not exist many tools for accurate grapheme-to-phoneme conversion.

• Handling words from different languages:
Word sounds from other language do not have much representation in the data, making it difficult to predict.
Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Previous Approaches
  – Latest Developments
• Demonstration
Basics of Signal Processing (1/5)

• Sampling and Quantization
  – Storing analog signal in bits would require infinite memory
  – We discretize the signal along two dimensions - time and value
  – The discretization along the time axis is called as *sampling*. This requires a sampling rate to specified
  – The discretization along the value axis is called as *quantization*. This requires deciding on how many bits we want to use to store one sample of the signal.
Digital system
  – Impulse Response:
    ○ The impulse response of a system is the output that is generated when a delta signal is given as the input
    ○ It is characteristic of any digital system and provides information about how the system reacts to change in input
  – Output of system:
    ○ Consider a digital signal $x[n]$, a digital system with impulse response $h[n]$, and the output signal $y[n]$
    ○ Then we have,
      \[ y[n] = x[n] \ast h[n] = \sum x[k] h[n-k] \]
      where, $\ast$ is the convolution operator
Basics of Signal Processing (3/5)

- Fourier Transform
  - Provides the frequency domain representation of any signal
  - Captures the relative strength of different frequencies in the signal
  - It is an invertible transformation

Image source:
https://www.researchgate.net/figure/Fourier-transform-of-a-sum-of-sinusoids-and-filtering-the-highest-frequency_fig3_237061998
Basics of Signal Processing (4/5)

• Discrete Fourier Transform (DFT)
  – Converts a digital signal input of finite length into same-length sequence which represents the samples of fourier transform of the input signal

• Short-time Fourier Transform
  – DFT is applied to small segments of input signal

• Spectrograms
  – The STFT of each segment of the input signal are vertically placed next to each other to form a matrix
Basics of Signal Processing (5/5)

Normalized Spectrogram

One way to model human speech is via the source-filter approach. Here, the speech is considered to be formed by passing a source signal through a filter:

- **Source**: Periodic vibrations from the glottis (vocal cord)
- **Filter**: Vocal tract, including the tongue to change its shape

**Underlying Hypothesis:**
All sounds of any language can be produced by passing a periodic signal through some specific Linear Time-Invariant filter.
Source-Filter Model (2/3)

- Linear Time-Invariant System
  - Linearity: A linear combination of any two signals as input would result in the same linear combination of corresponding outputs.
  - Time-invariant: Shifting the input signal along the time axis, will correspondingly shift the output signal.
Source-Filter Model (3/3)

Image source: https://www.vocalsonstage.com/vocals-on-stage-blog/resonance-and-articulation
Human Speech (1/2)

• Voiced and Unvoiced
  – When the generation of a sound requires vibration of glottis, then we say that it is a *voiced* sound.
  Note: All vowels are voiced sounds.
  – When the vibration of vocal cords are not used for articulation, the generated sound is said to be *unvoiced*.

• Plosives and Fricatives
  – *Plosives* are consonants which require a burst of air source. This is achieved via closure of mouth followed by sudden compression and burst of air (E.g. ब, फ)
  – *Fricatives* are consonants which involve a constriction in the vocal tract to create aspiration (E.g. /s/ in sick, /z/ in zebra)
Human Speech (2/2)

Air from lungs → Glottis → Impulse Train → Vocal Tract → Speech

- White noise (aspiration/fricatation)
- Impulse (plosion)
Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Previous Approaches
  – Latest Developments
• Demonstration
Perceiving Sound

• Human Hearing (What makes speech sound natural?)
  – Loudness: Proportional to the intensity of the sound
  – Pitch: Human sensation of various frequencies in sound
  – Prosody: Refers to rhythm, stress, and intonation in sound

• Absolute threshold of hearing
  – Minimum sound intensity of a pure tone that can be heard with no other sound in the environment
  – It is frequency dependent (Best range: 2kHz - 5kHz)
  – Measure in dB SPL, which is the pressure relative to 20 micropascal (quietest audible sound pressure level for normal hearing)
Logarithmic Scale and Hearing

- Weber–Fechner law:
  - Above a minimum threshold of perception $S_0$, the perceived intensity $P$ is logarithmic to the stimulus intensity $S$
    \[ P = K \log \left( \frac{S}{S_0} \right) \]

- Empirical evidences support the concept that logarithmic mapping in brain minimizes relative error in perception

- So, a change in intensity at a low intensity is more clearly audible compared to a change in intensity at a high intensity

Mel Frequency Cepstral Coefficient

• Taking the logarithmic hearing into consideration, we can extract MFCC features from any speech signal. The components are:
  – Sampling and Windowing
  – Discrete Fourier Transform
  – Mel Filter Bank
    ○ Mel frequency scale: \( m = 2595 \log_{10}(1 + f / 700) \)
  – Discrete Cosine Transform

• Mel Filter Bank:
  Pre-decided number of triangular bandpass filters separated according to the mel frequency scale within specified minimum and maximum frequencies
MFCC Extraction

Image source: https://www.researchgate.net/figure/Mel-frequency-cepstrum-coefficient-MFCC-calculation_fig4_277553387
Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Previous Approaches
  – Latest Developments
• Demonstration
History

• 1779 – Mechanical model was built that modelled the human vocal tract, and produced the five vowel sounds in English

• 1791 – Models of tongue and lips were added which allowed production of consonants along with vowels

• 1939 – Keyboard operated voice-synthesizer (Bell Labs)

• Late 1950s – First computer-based speech-synthesis systems
Diphone-based Synthesis (1/3)

• What is a phoneme?
  – The simplest, perceptually distinct units of sound in any language that clearly distinguish one word from another.

• What is a diphone?
  – A combination of two half-phones is called as a diphone.

• The core idea of diphone-based approaches is to decompose speech at the level of diphones and then combine these units based on the characters of input text.
Diphone-based Synthesis (2/3)

• Pitch Synchronous Overlap and Add:
  – Speech is divided into pitch-synchronous waveforms
  – These are then varied in time or spectral domain to obtain synthetic versions of same speech unit
  – Next, the newly formed speech units are overlapped and added to generate the new speech
  – Ideally, there would be no information loss

Image Source:
https://wiki.aalto.fi/pages/viewpage.action?pageId=155477136
Diphone-based Synthesis (3/3)

- **TD-PSOLA (Time Domain - PSOLA):**
  - The speed and pitch of a sound is manipulated in time domain
  - Pitch periods are extracted from the sound signal
    i. Pitch change: Segments are brought closer or separated
    ii. Speed change: Segments are repeated or deleted

- **FD-PSOLA (Frequency Domain - PSOLA):**
  - Spectral envelope is computed using linear predictive analysis
  - Pitch is modified via linear interpolation to obtain synthetic versions of same sound unit
  - Unnatural discontinuities at concatenation boundaries
Unit-Selection Synthesis (1/3)

- Selects & concatenates units (phonemes) from large database.
- Text can have additional annotations containing prosodic and phonetic context information.
- Database is transformed into a state transition network, with phonemes as states.
- The network is fully connected, since any sequence of phonemes is possible.

Unit-Selection Synthesis (2/3)

- **Cost Functions**
  - Weighted sum of difference between target and unit feature vectors → weights for each feature need to be learned
  - **Target cost:**
    - Measures the difference in selected unit and target
    - Feature vectors: pitch, power, duration, voicing, vowel/consonant, consonant type, point of articulation
  - **Concatenation cost:**
    - Measures the quality of a join between selected units
    - Feature vectors: cepstral distance, difference in log power, pitch
Unit-Selection Synthesis (3/3)

- Learning weights for cost functions
  - Weight Space Search
  - Regression Training

- Generating final speech
  - Using the fully connected graph and the cost functions we can use viterbi decoding for selecting final units
  - The search space for viterbi decoding is pruned based on
    - Phonetic context
    - Target cost
    - Concatenation cost
WaveNet

Wavenet (Google Deepmind)

- Dataset: North American English dataset (24.6 hours)
- First deep neural architecture trained to generate raw audio waveforms that surpassed many approaches of the time.
- Autoregressive generative model
- Model consisted of dilated causal convolutional layers to increase the receptive field of CNN

Content

• Introduction
• Foundations
  – Production of Human Speech
  – Science behind Human Hearing
• TTS Synthesis
  – Previous Approaches
  – Latest Developments
• Demonstration
Framework of TTS Systems

All models described ahead follow this pattern for speech synthesis
Text to Mel – Tacotron 2

Tacotron2 (Google) → MOS: 3.86

- Dataset: North American English (internal dataset)
- Autoregressive model
- Pre-Net module to bottleneck information flow
- Post-Net module for minor corrections in generated mel spectrogram

Perfect attention alignments
Text to Mel – FastSpeech

FastSpeech (Microsoft)

→ MOS: 3.84

• Dataset: LJSpeech (24 hours)
• Non-autoregressive model
• Feed-Forward Transformer architecture (memory intensive)
• Length regulator to control the length of generated phonemes.

Text to Mel – ForwardTacotron

ForwardTacotron (NVIDIA)

- Dataset: LJSpeech (24 hours)
- Non-autoregressive model
- Tweaked FastSpeech to remove quadratic bottleneck due to self-attention
- Tacotron2 purely used for phoneme duration extraction
- Trained to predict energy and pitch information from ground-truth.


Note: There is no paper for ForwardTacotron yet. Above details are taken from a blog released by NVIDIA.
Vocoder – WaveGlow

WaveGlow (NVIDIA)  
→ MOS: 4.00

- Dataset: LJSpeech (24 hours)
- Flow-based, non-autoregressive generative model
- In flow-based networks, the function approximated by the neural network is invertible
- So if \( z \) & \( x \) are the input & output respectively, then
  \[
  x = f_0 \circ f_1 \circ ... \circ f_k(z) \quad \text{&} \quad z = f_k^{-1} \circ ... \circ f_0^{-1}(x)
  \]
  where, \( f_1, f_2, ..., f_k \) are the layers of the model

**Fig. 1: WaveGlow network**

Generative Adversarial Networks

- GAN architectures consist of two different models - generator and discriminator
- Generator: Generates the desired data from a noise vector
- Discriminator: Performs the binary classification task of discriminating between the real data and the data generated by generator

**Loss Function:**

\[
E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]
\]
Vocoder – MelGAN

MelGAN

→ MOS: 3.72

• Dataset: LJSpeech (24 hours)
• First GAN model to produce high quality speech waveforms
• Non-autoregressive and fully-convolutional model
• Multiple discriminators are used to evaluate the input at various resolutions
• Number of parameters 20 times less than Waveglow

GAN-based Vocoders

- Recent vocoders have focused on 2 major components
  - GAN architectures
  - Multi-resolution Loss

- Many GAN architectures were introduced in 2020-2021 period
  - ParallelWaveGAN, HiFi-GAN, VocGAN

- All the architectures produced high-quality speech outputs

- A recent paper[11] (Aug 2021) hypothesizes that the performance can be majorly attributed to the multi-resolution discriminative framework, rather than the details of the actual architecture
Content

• Foundations
  – Production of Human Speech
  – Science behind Human Hearing

• TTS Synthesis
  – Introduction
  – Previous Approaches
  – Latest Developments

• Demonstration
# Data Description

- Marathi data from IndicTTS Database (developed by IIT Madras) of a female speaker.
- Sampling rate of speech: 22050 Hz

<table>
<thead>
<tr>
<th>Speech</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration</td>
<td>Total length</td>
</tr>
<tr>
<td>4.8 hours</td>
<td>24756 words</td>
</tr>
<tr>
<td>Mean duration</td>
<td>Mean length</td>
</tr>
<tr>
<td>7 sec</td>
<td>10 words</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>2.3 sec</td>
<td>2.7 words</td>
</tr>
<tr>
<td>Minimum duration</td>
<td>Minimum length</td>
</tr>
<tr>
<td>2 sec</td>
<td>4 words</td>
</tr>
<tr>
<td>Maximum duration</td>
<td>Maximum length</td>
</tr>
<tr>
<td>21 sec</td>
<td>22 words</td>
</tr>
</tbody>
</table>
Data Preparation (1/2)

- **Text Cleaners (TC):**
  - Simple transliteration cleaner: Converts to ASCII using unidecode library
    E.g. माझे नाव श्याम आहे → maajhe naav shyaam aahe
  - Indic transliteration cleaner: Uses indic transliteration library (for Indian languages)
    E.g. माझे नाव श्याम आहे → mAjhe nAva shyAma Ahe

- **Phonemizer:** Converts graphemes to phonemes (IPA). Can be used with TC.
  E.g. माझे नाव श्याम आहे → maːjeː naːv ʃjaːm aːheː
### Data Preparation (2/2)

**Mel Spectrogram Generation:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of mel filters</td>
<td>80</td>
</tr>
<tr>
<td>Minimum mel frequency</td>
<td>0 Hz</td>
</tr>
<tr>
<td>Maximum mel frequency</td>
<td>8000 Hz</td>
</tr>
<tr>
<td>Filter length</td>
<td>1024</td>
</tr>
<tr>
<td>Hop length</td>
<td>256</td>
</tr>
<tr>
<td>Window length</td>
<td>1024</td>
</tr>
<tr>
<td>Window type</td>
<td>Hanning</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>22050 Hz</td>
</tr>
</tbody>
</table>
Experiments and Analysis (1/3)

Less amount of data motivates the use of transfer learning.

- Tacotron2 + WaveGlow (pretrained vocoder):
  - Fine-tuned on Marathi data.
  - Experimented with both text cleaners and phonemizer.
Experiments and Analysis (2/3)

- Tacotron2 fails on some text inputs – 2 major issues

Māma ucyā vyavsthāpākālā
mala diṭelēni rakkam jāma
carṇyaśāṭi bāṅkēt gēlo hōlo,
pāṛtu tśāṃnī sāṅgitālē kē
gelyā 5 ṛṭivasāt mēṭhālēlē
eṣe tvarit jāma carṇyaṭchī
parvāṅgī nāāhī.

Hazy Alignment and Garbage Output
Experiments and Analysis (3/3)

- **ForwardTacotron + WaveGlow (pretrained vocoder):**
  - Trained from scratch on Marathi data
  - Experimented with phonemizer
नमस्कार, मेरा नाम श्याम है और मैं आईआईटी बॉम्बे का छात्र हूँ जो डेटा साइंस के क्षेत्र में अध्ययन कर रहा हूँ।

बैंक विवरण

आज मेस में दोपहर का भोजन वास्तव में स्वादिष्ट था, इसलिए अब मुझे अपने घर के पास खाने के लिए नई जगह ढूंढने की आवश्यकता नहीं है।

उड़ने वाले विवरण

गर्मियों में, कोरोना महामारी समाप्त होने के बाद, मैं उत्तर में हिमालय, दुनिया के सबसे महान पर्वत पर जाना चाहूंगा।
Summary (1/2)

• Text-to-Speech synthesis attempts to generate intelligible and natural sounding speech for any given text

• Many intricacies are involved during human speech production and source-filter model provides great insights into it

• Human hearing which is more sensitive to logarithmic frequency scale

• MFCC features align with the nature of human hearing and are great at capturing the characteristics of human speech
Summary (2/2)

• Earlier approaches created database and smartly concatenated individual phoneme-based units to generate the output speech.

• Deep Neural architectures first generate the mel spectrograms from the text which is then converted into waveforms by vocoders.

• Recently, GAN approaches have dominated as vocoders, and the adversarial training with multi-resolution training have ensured very high-quality output speech.
References


References


References


SSMT Demonstration
Conclusions
Conclusions (1/2)

- SSMT is the task to translate speech in language A into speech in language B through use of a computer.
- Applications: movie dubbing, movie Subtitling, conversing with foreign-language speakers, etc.
- Accent, hesitation, disfluency, and grammatically-flexible nature of spoken languages make SSMT more challenging than text-to-text MT.
- The task can either be trained -
  - as Direct Speech to Speech conversion, or
  - Cascaded SSMT. It can be divided into 3 subtasks:
    - Automatic Speech Recognition (ASR)
    - Machine Translation (MT)
    - Text-to-Speech synthesis (TTS)
Conclusions (2/2)

• We illustrated Direct SSMT approaches which work on signal level avoiding intermediate text generation.
• We discussed different ASR and TTS techniques and showed demonstrations.
• We also discussed Disfluency correction, Automatic post-editing techniques which are used to remove irregularities from speech transcriptions to make that ready for MT and vice versa.
• We discussed different paradigms of MT along with recent advancements (i.e. LaBSE filtering, Phrase table injection, Pivoting, Multilingual MT, Unsupervised MT). We also demonstrated the MT system.
• We demonstrated the entire SSMT pipeline.
Future directions

• Improving ASR results in Indian languages in a conversational setting and speaker diarization
• Checking usability of current neural APE approaches on improving translations obtained from a high-quality NMT system
• Restricting APE systems from performing unnecessary edits
• Improving disfluency correction with the help of other language data.
• Improving the naturalness of output speech through text processing for Indian languages TTS synthesis
Resources

All the resources will be made available here:

https://github.com/sourabhd13/ssmt_tutorial_icon2021
Thank You!