TranQuality

Improving Translation Quality Through Ensemble Translators and Paraphrase Detection

Brittany Dougall, Jai Raju
University of California, Berkeley

~23 seconds

Hi my name is Jai Raju and I have my partner Brittany Dougall from a different section here.

We present TranQuality - a model to improve Translation Quality Through Ensemble Translators and Paraphrase Detection

Objective

What:

- Natural language processing research has focused on high resource languages (English, French etc).
- Lack of labeled data for lower resource languages, or poor-quality data
- Our goal is to attempt to improve the quality of data so we can advance research.

Why:

- India comprises more than 17% of the world's population
- Making technology equitable will advance all mankind.

How:

- Within the last two years, several transformer models fine-tuned for Indian languages have been released.
- However, these models are not always accurate
- In our solution, we introduce a model that chooses the best translation among the several translations and thus provide a better output.

Jai -55 seconds

- India comprises more than 17% of the world's population[1], yet NLP research has focused on English and other high resource languages.
- This is due to the lack of good quality labeled training data for lower resource languages,
- In the last two years, several transformer models fine-tuned for Indian languages have been released. However, these models are not always accurate
- Our goal is taking a small step towards improving quality of data and advance research in the low resource languages

How do we do that:

- In our solution, we introduce 2 models that chooses the best translation among the several translations and thus provide a better output.

PMIndia

Model Analysis Data

- WAT 2021 8th workshop on machine translation dev set - pre-cleaned and filtered curated from the Prime Minister of India's website
- 1k English sentences with parallel Indian language corpora

Sentence 1	Sentence 2	Language
On behalf of the one point three billion people of India, I am delighted to welcome you all to New Delhi.	भारत के तीन बिलियन लोगों की ओर से आप सभी का नई दिल्ली में स्वागत करते हुए मुझे प्रसन्नता हो रही है।	Hi
On behalf of the one point three billion people of India, I am delighted to welcome you all to New Delhi.	இந்தியாவின் 130 லட்சம் மக்கள் சார்பில் உங்கள் அனைவரையும் புதுதில்லிக்கு வரவேற்பதில் மகிழ்ச்சி அடைகிறேன்.	Ta
On behalf of the one point three billion people of India, I am delighted to welcome you all to New Delhi.	130 കോടി ഇന്ത്യക്കാരുടെ പേരിൽ നിങ്ങളെ ന്യൂഡൽഹിയിലേക്കു സ്വാഗതം ചെയ്യാൻ എനിക്കേറെ സന്തോഷമുണ്ട്.	Ml

~20 sec
For our analysis, we needed 2 datasets, one to produce the translations for our ensemble method and a 2nd to tune the models to predict translation quality. For the 1st task, we used the PMIndia dev dataset consisting of 1k English sentences from the Prime Minister of India's website, with parallel Indian language corpora in 10 languages.

Amrita Paraphrase

Model development data

Language	Training	Validation	Test
HI	2.5k	448	448
TA	2.5k	425	426
ML	2.5k	450	450
PB	1.7k	250	250

Sentence 1	**Sentence 2**	Class
भारतीय मुस्लिमों की वजह से नहीं पनप सकता आईएस|	भारत में कभी वर्चस्व कायम नहीं कर सकता आईएस|	P
सुप्रीम कोर्ट के इस आदेश का असर यूपी के 6 पूर्व मुख्यमंत्रियों पर पड़ेगा।	कोर्ट के ऑर्डर के बाद इस सभी को 2 महीने में सरकारी बंगला खाली करना होगा।	NP

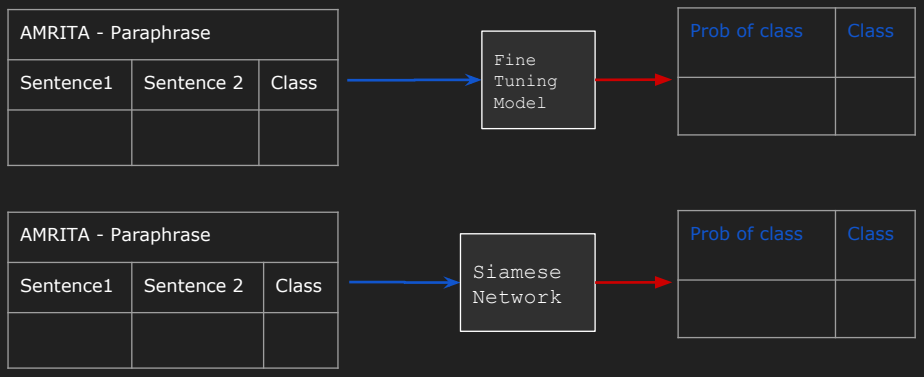
~35 sec To develop our translation evaluation models, we used Amrita University's paraphrase dataset, consisting of pairs of sentences labeled P or NP, for 4 Indian languages (Hi, Ta, Ml, and Pb). As seen here, the distribution of languages is relatively balanced for Hi, Ta, and Ml, with fewer available pairs for Pb. NP pairs comprise roughly 60% of records & P pairs 40% for each language in the training set, with these percentages varying slightly in the dev & test sets. We'll now move on to discuss our pipeline for analysis.

MODEL

We implemented two model approaches, one fine-tuning IndicBert directly and a 2nd using a Siamese neural network with IndicBERT embeddings. While multilingual pre-trained language models and Siamese networks have previously been used to evaluate translation quality, including for Indian languages, these approaches have not implemented the use of IndicBERT.

- The first model is a siamese network based model.
- The second is Fine tuning of an existing language model.

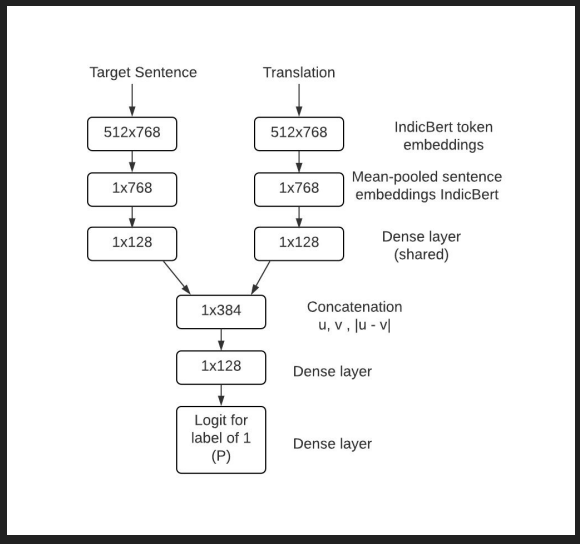
Model Pipeline



~29 sec For each PMIndia English record, IndicTrans and MBart each produce a translation in each of our target languages. These translations and the target texts are then passed to our models trained for paraphrase scoring. The paraphrase scoring models produce a paraphrase label logit(s) that is converted to a probability, and assign a P/NP label. For our direct IndicBert approach, separate models score each language; for the Siamese network, one model scores all languages. We'll now review our model designs and the parameters used for producing these scores.

Siamese Network

Architecture



- Layer normalization after each dense layer + relu
- Training for Best Performing Model:
  - 15 epochs
  - Learning rate 2e-5
  - BCEWithLogitsLoss as loss criterion

~30 sec
For the Siamese model, we first pass each sentence in each pair to IndicBert to get the token embeddings, then use mean-pooling to get their sentence embeddings (P/NP pairs during training and later target sentence and translation for evaluation). The sentence embeddings then separately pass through a shared dense layer (allowing them to share the same weights and biases), where their dimensionality is reduced. The reduced embeddings are then concatenated and pass through 2 more dense layers to reduce their dimensionality to 1 label for each pair of records, a logit for paraphrase status. We'll now discuss our approach using IndicBert models.

Fine tuning IndicBERT

Architecture

- Fine tuned Albert for sequence classification task
- One model for each of the 3 languages [TA, HI, MA]
- Grid Searched the following hyper parameter space
  - Epocs 1-5
  - Max_seq_length : [128, 256, 512]
  - Batch_size [8. 16. 32]
  - learning_rate: [2e-04, 2e-05, 2e-06]
- Inputs are a pair of sequence constructed as : [CLS] A [SEP] B [SEP]
- Number of output labels = 2
- We defaulted with the rest of the model parameters

```
TextClassification(
  (model): AlbertForSequenceClassification(
    (albert): AlbertModel(
      (embeddings): AlbertEmbeddings(
        (word_embeddings): Embedding(200000, 128, padding_idx=0)
        (position_embeddings): Embedding(512, 128)
        (token_type_embeddings): Embedding(2, 128)
        (LayerNorm): LayerNorm((128,), eps=1e-12, elementwise_affine=True)
        (dropout): Dropout(p=0, inplace=False)
      )
      (encoder): AlbertTransformer(
        (embedding_hidden_mapping_in): Linear(in_features=128, out_features=768, bias=True)
        (albert_layer_groups): ModuleList(
          (0): AlbertLayerGroup(
            (albert_layers): ModuleList(
              (0): AlbertLayer(
                (full_layer_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
                (attention): AlbertAttention(
                  (query): Linear(in_features=768, out_features=768, bias=True)
                  (key): Linear(in_features=768, out_features=768, bias=True)
                  (value): Linear(in_features=768, out_features=768, bias=True)
                  (attention_dropout): Dropout(p=0, inplace=False)
                  (output_dropout): Dropout(p=0, inplace=False)
                  (dense): Linear(in_features=768, out_features=768, bias=True)
                  (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
                )
                (ffn): Linear(in_features=768, out_features=3072, bias=True)
                (ffn_output): Linear(in_features=3072, out_features=768, bias=True)
                (dropout): Dropout(p=0, inplace=False)
              )
            )
          )
        )
      )
      (pooler): Linear(in_features=768, out_features=768, bias=True)
      (pooler_activation): Tanh()
    )
    (dropout): Dropout(p=0.1, inplace=False)
    (classifier): Linear(in_features=768, out_features=2, bias=True)
  )
)
```

Jai - 35 seconds
In this method, we fine tuned IndicBERT which is a multilingual ALBERT model trained on large-scale corpora, covering 12 major Indian languages:

Unlike Siamese network, we have individual model for each individual language.

We grid searched for 4 hyper parameters listed here.

Inputs are a pair of sequence as our task is sequence classification

Outputs are logits for the 2 classes

And we used all the other default model parameters


RESULTS

Paraphrase Test Results

Fine tuning IndicBERT		
Model	Accuracy	Loss
TA	86.13%	0.445
HI	90.21%	0.341
ML	87.55	0.441

Best Siamese Model	
Accuracy	Loss
81%	0.44

Siamese Model Language Accuracy	
Lang	Accuracy
TA	70%
HI	87%
ML	78%
PB	98%

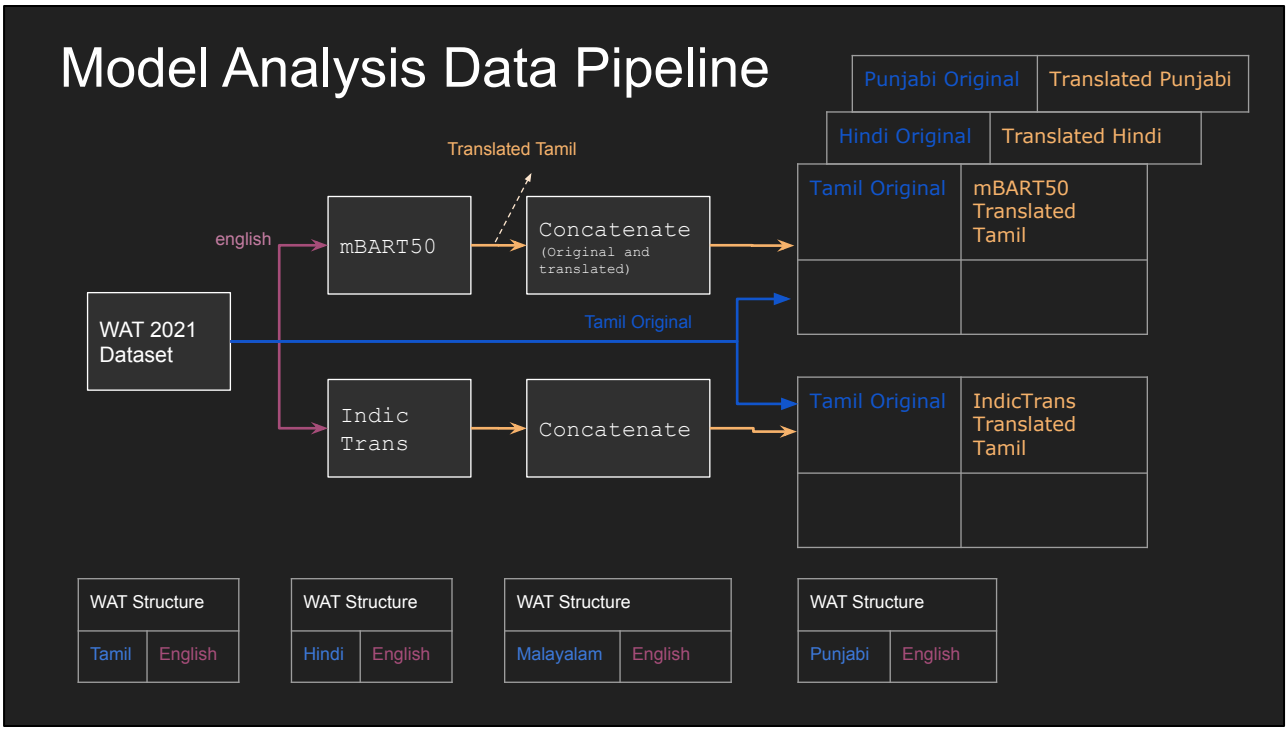
Jai

We experimented several variations like separating dravidian/aryan languages, single/individual models and so on. We chose models that performed the best and the results are presented here.

Siamese model is a single model and used an extra language dataset.

Testing our models on real data

Jai



Jai -45 seconds

Our objective was to learn a model that can choose between translations. This slide along with the next slide depicts the pipeline to accomplish that.

Here we show how we build the data for scoring

and in the next slide we show the scoring pipeline:

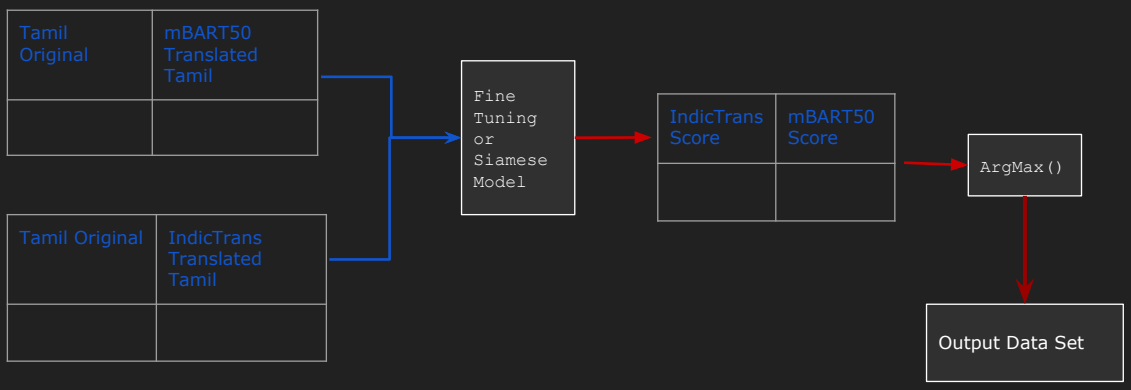
We start with WAT dataset as discussed earlier.

It has english and its translated Indian language sentences as shown in the bottom. We will use Tamil for example here.

We convert english sentences from this dataset to Tamil using 2 translators (mBART and IndicTrans here), but they could be any number of translators.

The output from the translator, Tamil in this case is merged with the original sentence so that we have a pair of sentences from each translator that will be used for scoring.

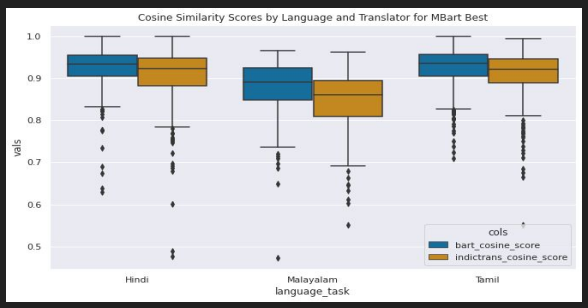
Analysis Data Scoring Pipeline



Jai - 15 seconds
We use the translated and the original as the inputs to our models. Our model outputs logits and we finally use a ArgMax function to choose the best translation.

Translation Results Analysis

Lang	Indic Trans	MBART	Siamese Ensemble	IndicBert Ensemble
HI	39.1	28.1	38.7	30.4
TA	15.4	13.0	15.5	15.9
ML	8.3	1.3	7.2	7.5



- Siamese model: ↑ cos scores, despite ↓ SacreBleu scores, edit distance generally ↓
- IndicBert models: variable changes in cos similarity and edit distance by language
  - TA + ML: cos scores ↑, edit distances ↓
  - HI: cos ↓, edit ↑
- Edit distances for MBART sentences tend to be higher on average than those from IndicTrans, even when considered more similar
- Evidence that our models detect transliteration and score transliterations poorly

On the left are the resulting SacreBleu scores for both of our ensemble methods. Despite higher accuracy rates, training IndicBert does not result in significantly higher SacreBleu scores than those of the Siamese model and both methods result in decreased scores for both Hindi and Malayalam. Our analysis reveals that this latter finding may not always be due to model error. For the Siamese network and for the IndicBert Tamil and Malayalam models, the sentences chosen tended to be more semantically similar to the target text than the sentences not chosen, regardless of translator source. In general, character edit distances for the chosen translation were lower than the translation not chosen. However, MBART translations have higher edit distances on average; when these sentences are chosen, it has the effect of decreasing Sacre Bleu scores, even if the majority of translations were chosen from IndicTrans. For the Hi IndicBERT model, cosine similarity scores tended to be lower and edit distances higher than the translation not selected, with translations being selected with little difference in their probability of paraphrase. This may be due to differences in Hi dialect between training and translation datasets - the translation dataset contained an older dialect than did the training dataset. Additional findings are detailed in our paper. Thank you for listening. Are there any questions?

Thank You

Questions ?