Language Translation

Team Mentor: Mr. Tulasi Ram

Team Coordinator: Snehal Tambe

Team Members: Parth Kumar

Rohan Kumar

Rohit Kumar Jha

Introduction

This case study shows how to create a model for text analysis and translation and deploy it as a web service in google colab in order to automatically create an initial english draft of the hindi stories.

Our combined team tried different approaches to tackle this challenge using:

- IndicTRANS model
- Helsinki-NLP/opus-mt-hi-en model
- Salesken/translation-hi-en model
- Custom LSTM model

Problem Statement and Objective

XYZ is a leading Indian publisher who has been in the market for more than 15 years. They publish books, novels, articles in multilingual formats for reading enthusiasts in fiction, nonfiction, sci-fi and all genres of literature. Right now, they are tasked with re-publishing the works of the popular Hindi novelist, Premchand from Hindi to the English language in a special series. However, they must act fast as one of their competitors is also working on the same series, hence they need a Hindi to English translator which will help them to translate the stories and publish them right in time for them to have an edge. The Hindi to English translator should create a rough draft of the fictional work. This rough draft should ease out the task of a human who aims to translate the fictional work and publish it quickly.

User Stories

1. As part of management, I would be interested to see how the language translator will impact the top and bottom lines of the business.

2. As a translator, I would like to understand how the language translator tool can be effectively used to leverage the translation process, improve the efficiency of the job and reduce the time required for translation.

3. As a data science lead, my focus will be to: - a) To understand the literary style of Premchand, words frequently used by Premchand and data analysis of the Premchand literary work b) To design the architecture and process flow for the language translation tool. c) The tool should be designed in a cost-effective manner using the existing open-source datasets and language models minimizing cost and time for development. d) Data preparation pipeline for using the language translation model e) Fine tune the language model f) Measuring the performance of the translation process g) Finalizing the architecture and the models to be used in production

4. As a UI lead, my focus is to design an intuitive UI which can be used by the translators to smoothen the translation process.

5. As a marketing lead, I would like to understand how accurate and effective the translator is to be able to build marketing strategies and create a pre-publishing buzz.

6. As a product lead, I would like to see how time and cost effective the language translator is to make sure the product is rolled out in time with the expected accuracy/ quality.

7. As a data science lead, I would like to know if this model can be reused for any other similar product for translation. Please do let me know what you think.

Expected Solution

Solution will focus on developing a translation tool which will ease the work of humans in the loop in the translation process. The end solution will consist of an UI which can be used to upload the document required for translation. Once the target language is selected, the UI should provide the option to download the document in the target language.

Work Done

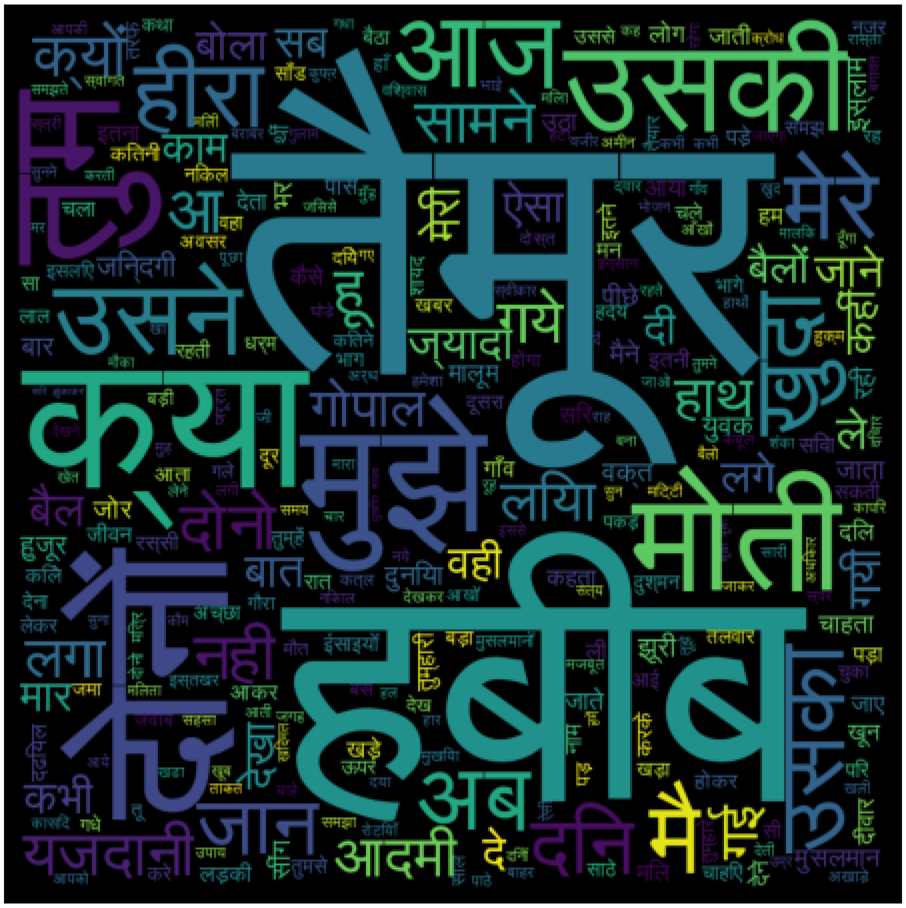
What will you find inside:

- How to clean and prepare text data and featurize it to make it valuable for machine learning scenarios
- How to create a translation model using NLP tools.
- How to create a web UI using streamlit
- How to deploy a trained model on google colab.

DataSet

PremChand Hindi Stories : http://premchand.co.in/
PremChand Human Translated English Stories :
https://www.arvindguptatoys.com/arvindgupta/premchand11.pdf

Exploratory Data Analysis (EDA)



Word Cloud of PremChand Stories

- Author frequently used so many words from Urdu literature. Even characters' names are inspired by urdu literature.
- The writing style is not purely in hindi, it is somewhat a mixture of both hindi and urdu.

Data Cleaning

Data cleaning is the art of extracting meaningful portions from data by eliminating unnecessary details.

It is the process of preparing raw text for NLP (Natural Language Processing) so that machines can understand human language. Text cleaning can be performed using simple Python code that eliminates stopwords, unicode normalization, and simplifies complex words to their root form.

Stop words, such as "am," "the," and "are," occur frequently in text data. Although they help us construct sentences properly, we can find the meaning even if we remove them. This means that the meaning of text can be inferred even without them. So, removing stop words from text is one of the preprocessing steps in NLP tasks. In Python, nltk, and textblob, text can be used to remove stop words from text. We have used nltk to remove the stopwords and punctuations.

We performed the two most basic text cleaning techniques on our text data:

1. Normalizing Text
2. Removing Stopwords

Feature Extraction from Texts

Machine learning algorithms do not understand textual data directly. We need to represent the text data in numerical form or vectors. To convert each textual sentence into a vector, we need to represent it as a set of features. This set of features should uniquely represent the text, though, individually, some of the features may be common across many textual sentences. Features can be classified into two different categories:

- **General features**: These features are statistical calculations and do not depend on the content of the text. Some examples of general features could be the number of tokens in the text, the number of characters in the text, and so on.

- **Specific features**: These features are dependent on the inherent meaning of the text and represent the semantics of the text. For example, the frequency of unique words in the text is a specific feature.

Baseline Model

We have tried to evaluate three different models :

1. IndicTRANS model
2. Helsinki-NLP/opus-mt-hi-en model
3. Salesken/translation-hi-en model

We have selected one language translation model from AI4 Bharat indicnlp : IndicTrans. And we have selected two language translation models from Hugging Face : Salesken and Helsinki as models.

We have only taken text data from PremChand's stories to calculate the features and evaluate the models.

Indic Trans model:

- Dataset trained on : samanantar
- Model: transformer-XL

IndicTrans is a Transformer-XL model trained on samanantar dataset. Two models are available which can translate from Indic to English and English to Indic. The model can perform translations for 11 languages : Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu.
After evaluating our baseline model, the results obtained are:
We got a BLEU score of 0.63.

Helsinki model:

- Dataset trained on : opus
- Model: transformer
- pre-processing: normalization + tokenization + BPE

After evaluating our baseline model, the results obtained are:
We got a BLEU score of 0.55.

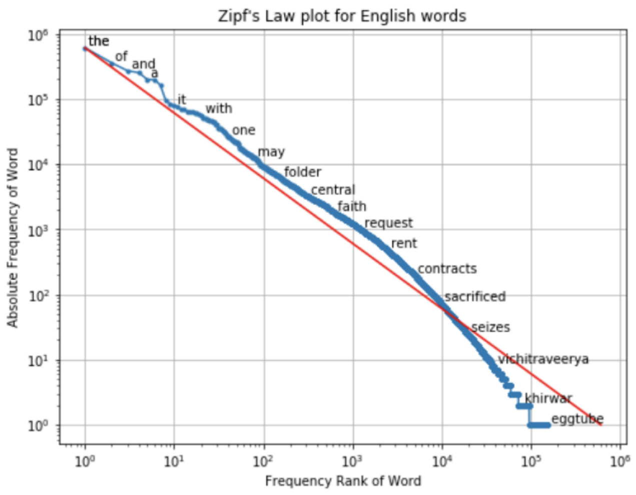
Salesken model:

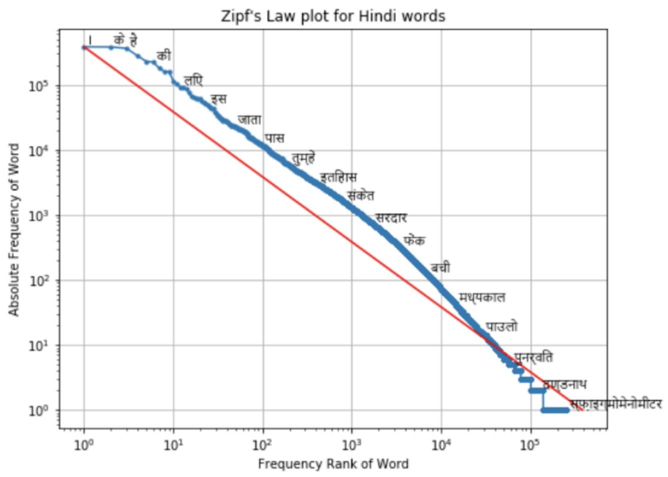
- Dataset trained on : samanantar
- Model: opus-mt

After evaluating our baseline model, the results obtained are:
We got a BLEU score of 0.408.

Pre-processing

The languages chosen for building these translation models were *Hindi* and English, and the parallel corpora was obtained from the *IIT Bombay Hindi-English Parallel Corpus*. This is a Hindi-English parallel corpus containing **1,492,827** pairs of sentences. To understand the word distributions in both languages, respective ***Zipf's law*** plots are shown below :





Pre processed tasks performed on dataset are::

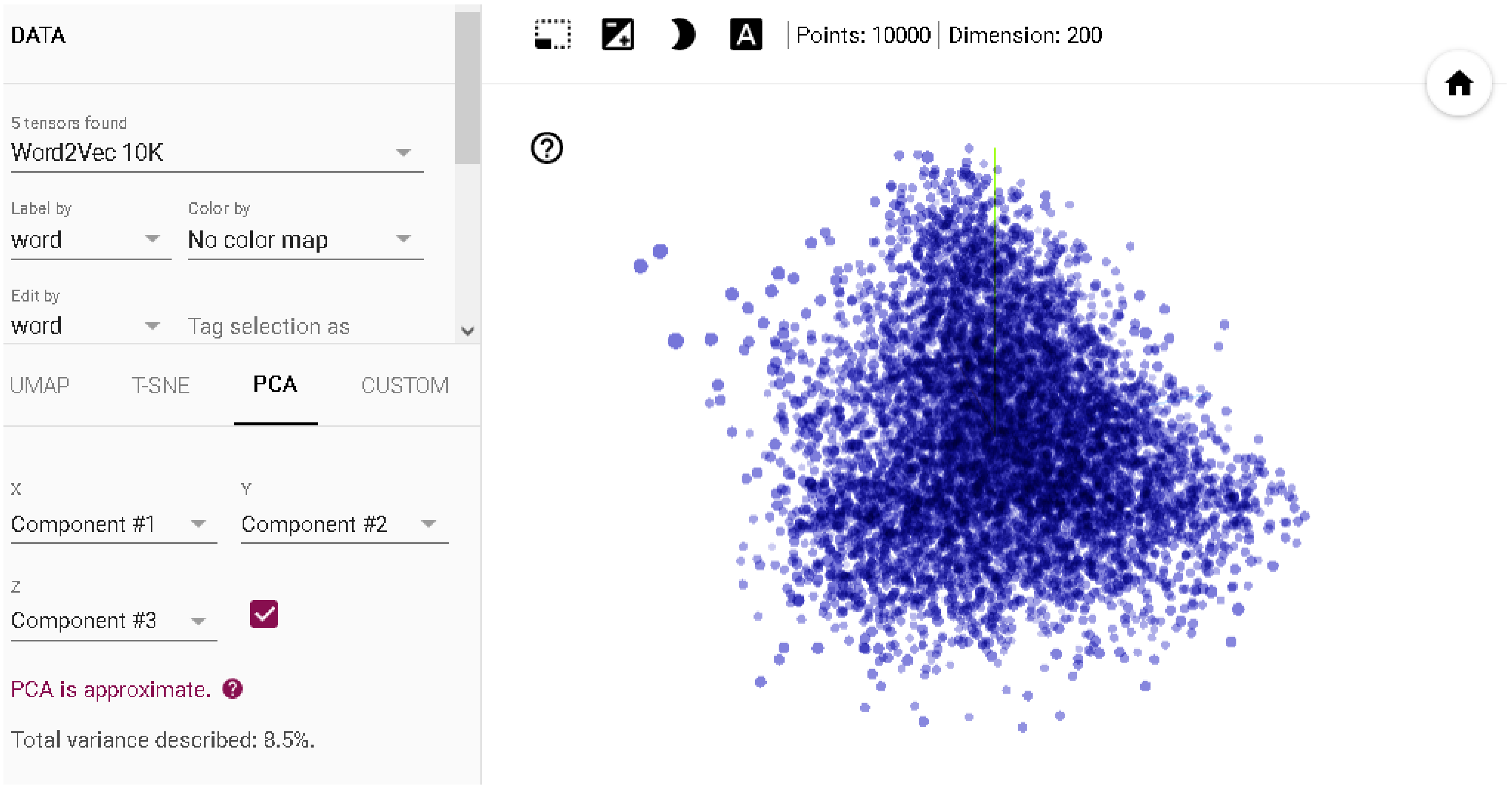
- Dealing with sentence length outliers (some sentences are clubbed together and result in the whole sequence length being 2000, which is grammatically illogical! It is always better to get rid of those, specially in this case, there were 4 such sentences.)
- Basic text cleaning: bad data in files (quotes from other languages), copyright statements and e-mail addresses (need no translation), punctuation, digits, converting to lowercase etc..

Tokenizing and Indexing

For tokenization, we find all the unique words in both languages. This will determine the dimensionality of the index arrays.
We are breaking the sentences and assigning an integer to each unique word, mostly creating a dictionary.
Sentence: This is my home
Tokenization: ['This','is','my','home']

Indexing: This -> 1
is -> 2
my -> 3
home -> 4

Vector Space / Embeddings



DATA

5 tensors found
Word2Vec 10K

Label by: word
Color by: No color map
Edit by: word
Tag selection as

UMAP | T-SNE | PCA | CUSTOM

X: Component #1
Y: Component #2
Z: Component #3 ✓

PCA is approximate.

Total variance described: 8.5%.

Points: 10000 | Dimension: 200

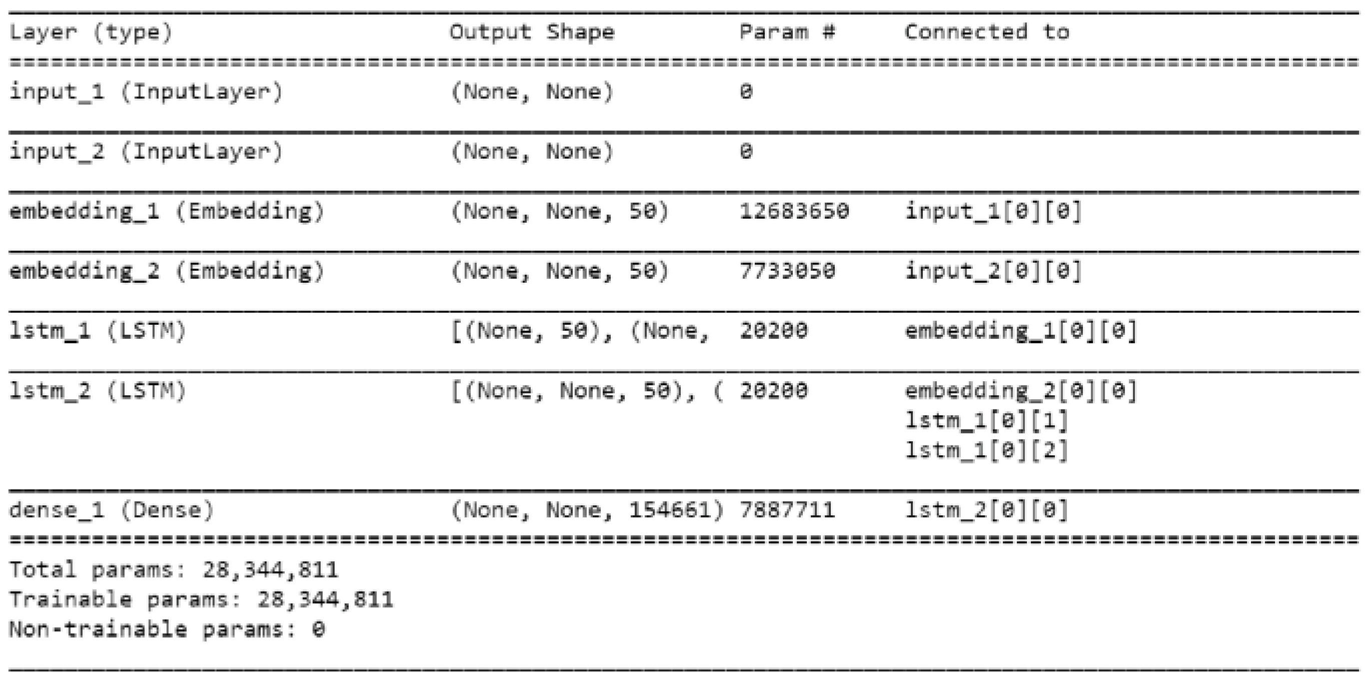
Sequence to Sequence Learning

Sequence to sequence models are used when the output is to be predicted after reading the whole sequence. A basic Sequence to sequence model has an encoder-decoder architecture. This model has two LSTMs, one each for encoder and decoder. The general working of a sequence to sequence model is outlined below:

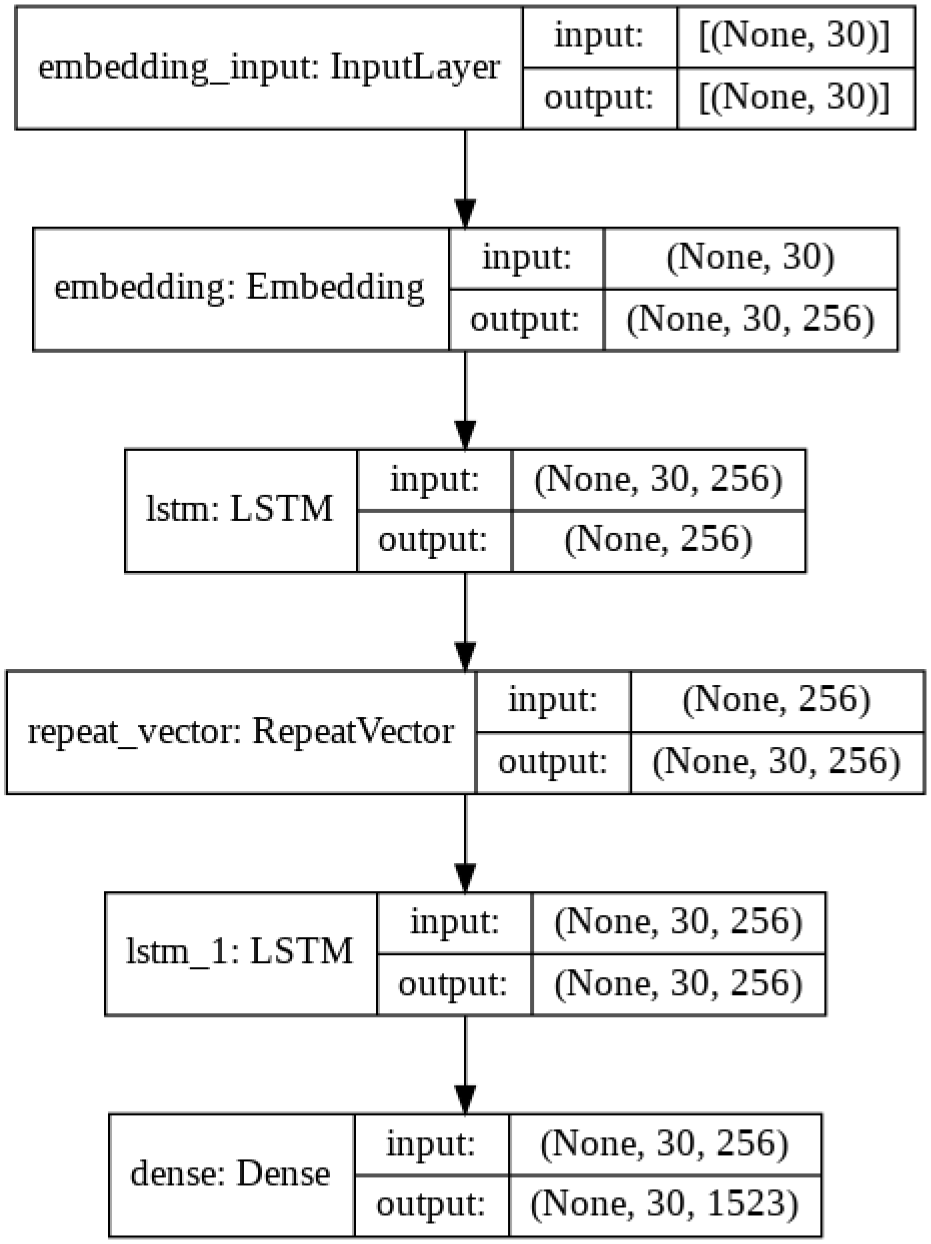
1. Feed the embedding vectors for source sequences (Hindi), to the encoder network, one word at a time.
2. Encode the input sentences into fixed dimension state vectors. At this step, we get the hidden and cell states from the encoder LSTM, and feed it to the decoder LSTM.
3. These states are regarded as initial states by decoder. Additionally, it also has the embedding vectors for target words (English).
4. Decode and output the translated sentence, one word at a time. In this step, the output of the decoder is sent to a softmax layer over the entire target vocabulary.

Custom LSTM model

The Input of a decoder LSTM goes through the Embedding layer, where word embeddings are created for the words . The embedding dimension chosen for both encoder and decoder networks is 50. Finally, the model summary looks like this:



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, None)         0
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, None)         0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 50)     12683650    input_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, None, 50)     7733050     input_2[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   [(None, 50), (None,  20200       embedding_1[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                   [(None, None, 50), ( 20200       embedding_2[0][0]
                                                                 lstm_1[0][1]
                                                                 lstm_1[0][2]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, None, 154661) 7887711     lstm_2[0][0]
==================================================================================================
Total params: 28,344,811
Trainable params: 28,344,811
Non-trainable params: 0
__________________________________________________________________________________________________
```



BLEU Score

BLEU (Bilingual Evaluation Understudy) is a measurement of the differences between an automatic translation and one or more human-created reference translations of the same source sentence.

Scoring process

The BLEU algorithm compares consecutive phrases of the automatic translation with the consecutive phrases it finds in the reference translation, and counts the number of matches, in a weighted fashion. These matches are position independent. A higher match degree indicates a higher degree of similarity with the reference translation, and higher score. Intelligibility and grammatical correctness are not taken into account.

How BLEU works?

BLEU's strength is that it correlates well with human judgment by averaging out individual sentence judgment errors over a test corpus, rather than attempting to devise the exact human judgment for every sentence.

BLEU results depend strongly on the breadth of your domain, the consistency of the test data with the training and tuning data, and how much data you have available to train. If your models have been trained on a narrow domain, and your training data is consistent with your test data, you can expect a high BLEU score.

Smoothing Function Used?
We have used SmoothingFunction().method7 with BLEU function to calculate the score.

Comparison

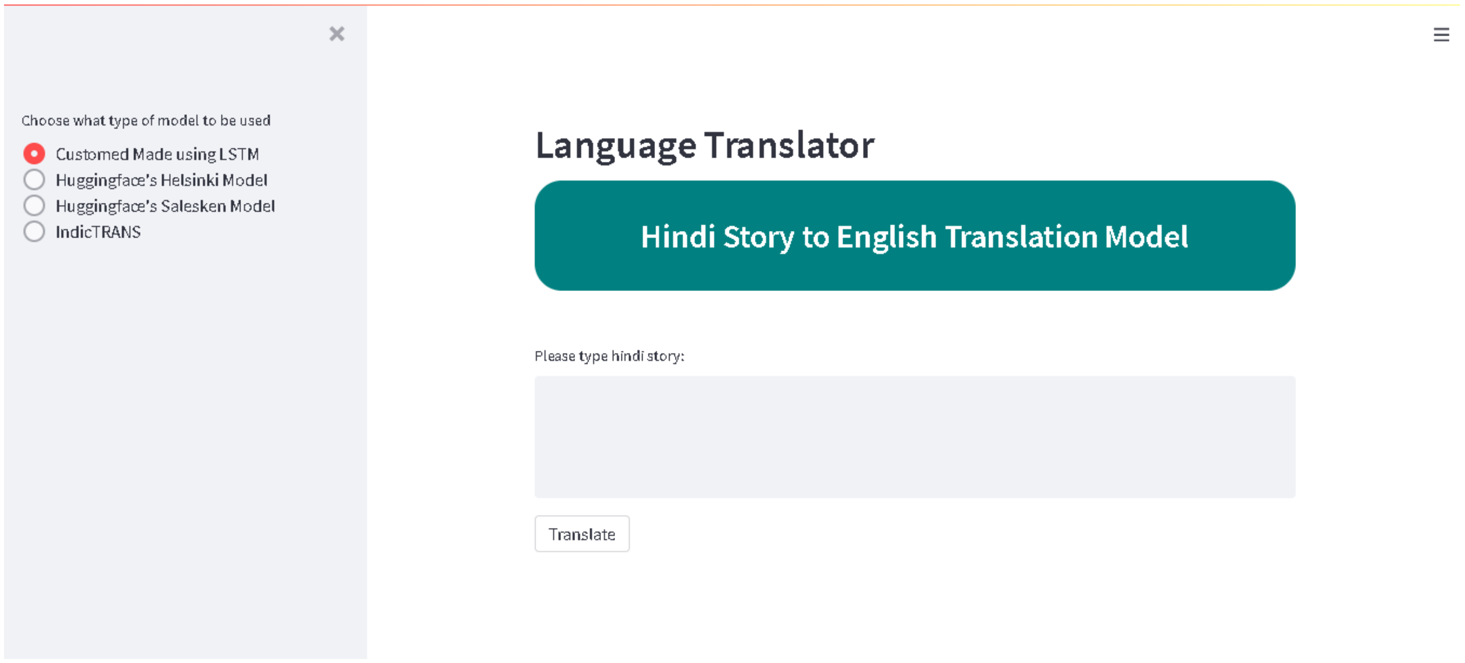
		INDIC Trans	Helsinki	Salesken	Custom LSTM
BLEU SCORE		0.62	0.55	0.408	0.159
ROUGE SCORE	Recall	0.18	0.012	0.37	0.25
	Precision	0.42	0.7142	0.45	0.153
JACCARD SIMILARITY METRICS	Accuracy	32.81	42.36%	33.04%	42.86%
	Precision	54.70	64.85%	53.47%	42.86%
TF IDF Similarity		91.8	87.6658 %	92.2328 %	65.5308 %
EMBEDDING FASTTEXT Similarity		81.46%	80.39%	81.38%	36.06%
EMBEDDING GLOVE Similarity		92.92%	92.21%	93.21%	37.25%

Stream lit

Streamlit is an open-source python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes, you can build and deploy powerful data apps.

In this project, we have created the application user interface using the streamlit module of python. So, there are primarily two functions in this application, one is to preprocess the text information input by user and convert it into the word-embeddings. And the other is to translate that into english text.

APP UI:



Choose what type of model to be used

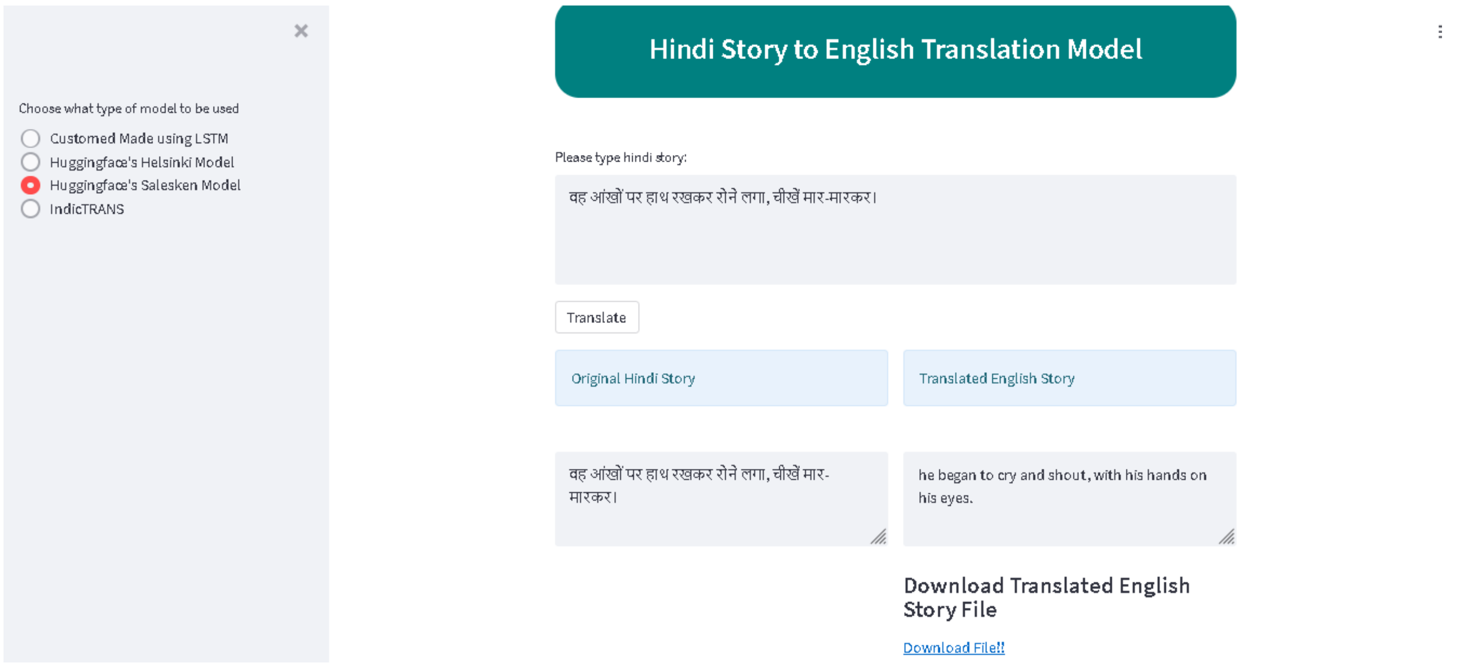
- [✓ Customed Made using LSTM]
- Huggingface's Helsinki Model
- Huggingface's Salesken Model
- IndicTRANS

Language Translator

Hindi Story to English Translation Model

Please type hindi story:

Translate



Choose what type of model to be used

- Customed Made using LSTM
- Huggingface's Helsinki Model
- [✓ Huggingface's Salesken Model]
- IndicTRANS

Hindi Story to English Translation Model

Please type hindi story:

वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।

Translate

Original Hindi Story	Translated English Story
वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।	he began to cry and shout, with his hands on his eyes.

Download Translated English Story File

Download File!!

Testing

Language Model	Hindi Text	Translated English Text
Salesken Model	वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।	he began to cry and shout, with his hands on his eyes.
Helsinki Model	वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।	He laid his hands on his eyes, and wept. Crying, he struck.
Indic Trans Model	वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।	He laid his hands on his eyes and cried loudly.
Custom LSTM Model	वह आंखों पर हाथ रखकर रोने लगा, चीखें मार-मारकर।	The cry of the he.

Checks

Text	Output
‘ ’ (blank space)	Warning
‘ ./’;.,? ’ (punctuations only)	
‘122345’ (numbers only)	
‘ one two three four’ (less than five character)	
‘ and his her hasn’t didn’t you’ (contains stop words only but no information regarding ticket)	

Warnings = {'No information has been written! Kindly write your query again.',

'You have written punctuations only. Kindly write a proper query again.',

'You have written numbers only. Kindly write a proper query again.',

Text information provided is too low. Kindly write at least five words in the query.', 'The information you have entered does not contain any appropriate knowledge. Kindly reframe your query.'}

Technology Stack

Element	Description
Pandas	Python library used for data manipulation & analysis.
Numpy	Used to perform a wide variety of mathematical operations on arrays.
Wordcloud	It is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.
NLTK	Python package used for Natural Language Processing (NLP).
Sklearn	Used for model building and evaluation using various metrics.
Pickle	Used to dump/load a model or any python object.
Streamlit	Used to create and share beautiful, custom web apps for machine learning and data science.
Embedding as service	Used for vector space models.
Indic Trans	Used for Language Translation.
Hugging Face	Transformer Library.

Conclusion:

Our goal in introducing machine translation was to increase scalability and better serve the customers. Advantages include:

- Reducing translation time by enhancing translators' productivity.
- Increasing content freshness by publishing more frequent updates.
- Reinvesting savings into high-value content and products.

Future Scope:

- We envision extending Machine Translation to other languages.
- To machine translate UI software strings and other content (i.e. knowledge articles, developer's guides, and so on) in the near future.
- We could make Machine Translation API available to our customers as an out-of-the-box or trainable product. There is so much content out there that people need to understand but not enough time and resources to human translate it all! That's where our NMT system can be the protagonist.
- We could develop a plugin to track MT quality systematically, train our translators on MT post-editing best practices, and reduce training time for the MT models per language.

References

Streamlit:

Introduction To Streamlit. What is Streamlit? Streamlit Tutorial Pt.1

Deploy Machine Learning Models Using StreamLit Library- Data Science

https://docs.streamlit.io/en/stable/api.html