Project Report

Idea

In recent times, with breakthroughs like word embeddings in the Natural language Processing domain, there are a lot of advancements happening. Each year companies like Google, Microsoft and Apple, etc. are showcasing new advancements in natural language understanding as the main achievements in their developer conferences. This will help in reducing the friction between the machine and human interaction, opening up possibilities for faster iteration of technology as everyone will be able to use a machine with minimum learning curve. Having said that, it is important to note that most of these advancements are currently done in very few languages and those languages do not cover even the majority of the world. The true vision lies when those advancements can be reflected and used in all the languages. If I can leverage the advancements made in any language for understanding something meaningful from the natural text in that language, then I can truly create a system, where I can impact any advancement made in the field of NLP to a global audience. With the vision I just described above, In this project I want to see if I can leverage the advancements done in English language to be used in "Hindi" (a language spoken by a large population in south Asia subcontinent) and I want to work on this language specifically because this is my native language also. So, the idea I want to focus on here is can I first convert any "Hindi" sentence into "English" and then use any natural language application created in English to see if the results can be reflected back to Hindi.

Experiments And Results

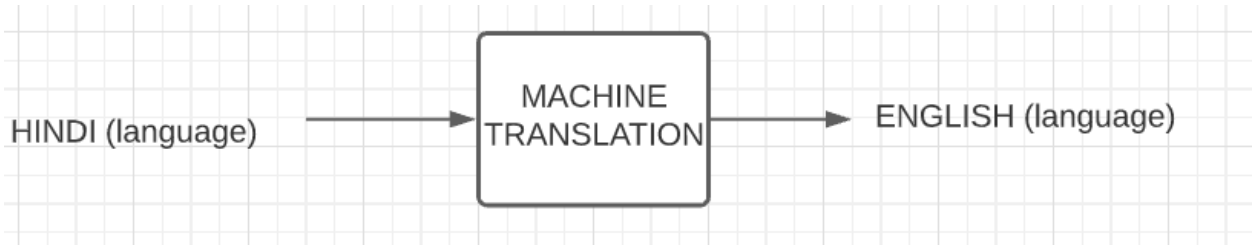
Files:

NMT_Preprocessing.ipynb: Preprocesses the data and creates necessary pickle files (used by below 2 files)

NMT_English2Hindi.ipynb: Run this file to train model for English to Hindi translation

NMT_Hindi2English_latest.ipynb: Run this file to train model for Hindi to English translation

Phase 1: Machine Translation from Hindi sentences to English sentences



Baseline used: I output ‘on’ for all the corresponding Hindi words in the Hindi sentence

```
Input:  संलग्नक बार दिखायें B
True Output:  show attachment bar
Baseline Output:  on on on on

Input:  क्या मासिक दृश्य में सप्ताह को रखना है जो शनिवार व रविवार को एक सप्ताह दिन में रखता है
True Output:  whether to compress weekends in the month view which puts saturday and sunday in the space of one weekday
Baseline Output:  on on on on on on on on on on on on on on on on on on

Input:  यह बैठक दिया जा चुका है
True Output:  this meeting has been delegated
Baseline Output:  on on on on on on

Input:  प्रतिनिधि D
True Output:  delegatees
Baseline Output:  on on

Input:  ई डाक प्रमाणपत्र प्राधिकार
True Output:  email certificate authority
Baseline Output:  on on on on

/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
0.24232627169060802
```

Figure shows Baseline model output and BLEU score (=`0.24232627169060802`) on the test data.

Encoder-decoder Model with attention mechanism:

Stats about training:
Input vocab size (Hindi) = 16031
Output vocab size (English) = 12725
Training data = 980000
Test data = 2000

Analyzing Model output:

Given Input: प्रश्न
Given Output: ask a question
Model Output: query <EOS>

Comments: It’s a good translation as question and query means the same thing. That is the model output resembles the given output and input source closely.

Given Input: कोई हाल में प्रयुक्त परियोजना नहीं
Given Output: no recently used project
Model Output: no recently used project <EOS>

Comments: The output is exactly the same. It is one of the best examples of the translation task.

Given Input: बनाएँ अतरफलक फ़ाइल

Given Output: create gtk builder interface file
Model Output: create gtk file <EOS>

Comments: The model output is very close in meaning to the given output. Thus it has done a good job in translating the source sentence.

Given Input: अधिसूचना
Given Output: notification
Model Output: notification <EOS>

Comments: The output is exactly the same. It is one of the best examples of the translation task.

Given Input: a को चिड़ी का बादशाह पर घुमाएँ
Given Output: move a onto the king of clubs
Model Output: move a onto the seven of clubs <EOS>

Comments: The model output is almost same to the given ouput except 'king' and 'seven'. Hence, the model output nicely represents the source sentence.

Given Input: फ़ाइल सिस्टम में जोलिएट एक्सटेंशन जोड़ें
Given Output: add joliet extensions to the file system
Model Output: add file to file to the <EOS>

Comments: The model does not seem to work always, like in this example.

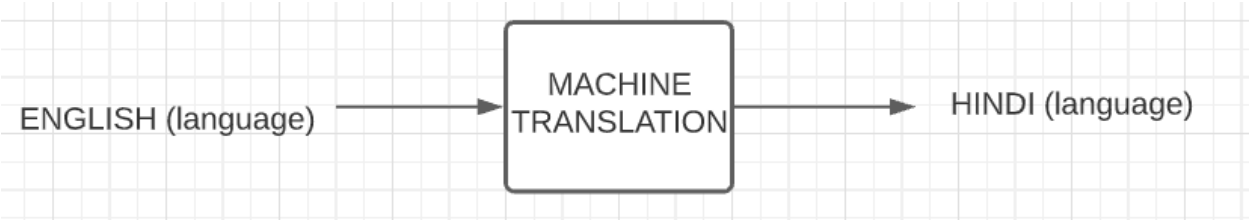
Given Input: एशिया चोंगकिंग
Given Output: asia chongqing
Model Output: asia debugger <EOS>

Comments: unique and peculiar names are sometimes not found in the vocabulary of the target language and thus one cannot expect to form a complete sense of the input sentence.

Metrics:

Bleu score on training data: 0.2784911840844412
Bleu score on test data: 0.17898235229167853

Phase 2: Machine Translation from English sentences to Hindi sentences



Baseline used: I output 'का' for all the corresponding english words in the english sentence

```
Input:  memo information sent
True Output:  ज्ञापन सूचना प्रेषित
Baseline Output:  का का का

Input:  update attendee status
True Output:  उपस्थित प्रस्थिति अद्यतन करें ु
Baseline Output:  का का का

Input:  can not get source list s
True Output:  स्रोत 2 को मुक्त नहीं कर सकता है
Baseline Output:  का का का का का का

Input:  the proxy tab will be available only when the account is enabled
True Output:  प्रॉक्सी टैब के उपलब्ध होने पर खाता सक्रिय हो जायेगा
Baseline Output:  का का का का का का का का का का का का

/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
0.2570796996425867
```

Figure shows Baseline model output and BLEU score (= `0.2570796996425867`) on the test data.

Encoder-decoder Model with attention mechanism:

Stats about training:
Input vocab size (Hindi) = 16031
Output vocab size (English) = 12725
Training data = 98000
Test data = 2000

Analyzing Model output:

Given Input specify session management id
Given Output: सत्र प्रबंधन ID निर्दिष्ट करें

Model Output: सत्र प्रबंधन ID निर्दिष्ट करें
Comments: The model has accurately learnt to translate the training data as shown in the example above. All the words match to the given output. There is 100 percent match with the source sentence semantic or meaning. प्रबंधन ID is session management.

Given Input: reload
Given Output: पुनः लोड करें R
Model Output: पुनः लोड करें R

Comments: perfect example of translation task. Shows how accurately the model learns over training data by minimizing the train loss.

Given Input: mode
Given Output: प्रकारः
Model Output: विधि

Comments: these 2 hindi words are synonyms which means they both specify same semantics as of the input sentence.

Given Input: ssl is not available in this build
Given Output: एसएसएल नहीं उपलब्ध है इस बिल्ड में
Model Output: यह इस में नहीं नहीं

Comments: It fails to capture the complete meaning of the source sentence. It only learns the negation affect in the source sentence.

Given Input: uri of an image file to burn autodetected
Given Output: किसी छवि फ़ाइल की Uri जिसे लिखा जाना है autodetected
Model Output: छवि फ़ाइल की URI जिसे लिखा जाना है

Comments: All the words match exactly and also conveys the semantics as conveyed in the source sentence.

Given Input: script
Given Output: स्क्रिप्ट
Model Output: स्क्रिप्ट

Comments: perfect match

Given Input: no recently used project
Given Output: कोई हाल में प्रयुक्त परियोजना नहीं
Model Output: कोई प्रयुक्त कोई प्रयुक्त नहीं

Comments: It is hard to understand what meaning the model output conveys But it still match some words and thus conveys partial meaning only.

Given Input: template

Given Output: टेम्पलेट

Comments: Single words sentences are easily matched as they donot have long context and no information is lost while encoding through the encoder.
Model Output: प्रविष्ट

Given Input update watch
Given Output: अद्यतन निरीक्षण करें
Model Output: अद्यतन निरीक्षण करें

Given Input certificate has been revoked
Given Output: प्रमाणपत्र वापस किया गया
Model Output: प्रमाणपत्र वापस नहीं

Comments: though most of the words match, but the meaning is exact opposite to what is conveyed by the source sentence and its given translation. Thus the model does not only have to do word by word translation. It needs to properly understand the context, whereas in this example it looks like the model has translated the sentence word by word.

Metrics:
Bleu score on training data: `0.3530796363692587`
Bleu score on test data: `0.11691222326255345`

Phase 3

In this approach I want to show the gap between doing a NLP task specifically in a certain language versus using a pre-trained model in another language with conversion support from machine translations between those languages. I also want to learn how scalable are the results I learn from a model trained on one language vs a model trained on multi-language.
To understand this scenario, I did two different methodologies for doing sentiment analysis in a foriegn language (Hindi).

1. Use multi-lingual bert model embeddings to train a classifier for sentiment analysis directly for hindi language.
   For this task the code is uploaded here: github
   SETUP AND INSTALLATION:
   1. Pip install all requirements from the requirements file.
   2. Run the ipynb notebook

   Data :
   1. Source: https://www.kaggle.com/disisbig/hindi-movie-reviews-dataset/version/1
   2. training sample :718
   3. Test samples: 180

   Results:

	Loss	Accuracy
Train	0.60	0.67
Test	0.59	0.69

2. Train a two way translator, one from Hindi to english and another from english to hindi and in between use a sentimental analysis model trained on english language.
   For this task the code is uploaded here: github
   SETUP AND INSTALLATION:
   3. Pip install all requirements from the requirements file.
   4. Run the ipynb notebook

   Data :
   4. Source: https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
   5. training sample : 20000
   6. Test samples: 70

   val_loss: 0.4849 - val_accuracy: 0.8810

	Loss	Accuracy

Train	0.0708	0.9755
validation	0.48	0.88

The result on the sample of the data from the valid data : 0.38

Why Hindi ? Two major reasons, the data set was available and it's my native language.

Also if you note in the above example, the training data source for above two tasks are different but evaluated on the same test set. Also, if you see the amount of data used in each is also very different. This is deliberately done, the reason behind this is the motivation of this project. In this project I want to see if the machine translation to another language is better than a model trained on that specific language. If the data size will be the same then the effort does not make sense in practicality, I want to do this effort to see how much difference I can observe by doing translation to a language which is trained on a much larger data set.

The results on the second one looks really low when compared to 0.69 of the above one, but there are two levels of noise here; first is the translation noise multiplied by the model accuracy. Since the model accuracy itself was high in the English language it shows that the major loss is during translation.

Things tried and Future Works

I tried one more domain, these days after the question generation tasks are getting very popular. Release of GPT-3 has boosted the advancement in generation tasks in NLP. So, I want to try the t5 paraphrases to generate questions given the sentence after doing machine translation. I tried both variant where I used hindi directly using a t5 trained on multilingual bert model and using t5 trained on english sentences. But due to lack of proper data, results were not getting produced in direct hindi domain using multilingual bert and the error propagation of hindi to english model trained by me followed by t5 paraphrase on the translated english is creating very noisy questions and answers. I would like to work on this in future again and the code to improve on can be found at : github

References:

https://pytorch.org/

https://huggingface.co/

IIT Bombay English-Hindi Parallel Corpus

https://www.kaggle.com/disisbig/hindi-movie-reviews-dataset/version/1