Machine Translation Application

- Internship Report

SIVATEJA PATHIPATI

DATE: 29-SEP-2024

1. Introduction:

This report consists a concise information about various projects made during the internship period. It contains details about background, Learning objectives, Activities and tasks, Skills and competencies, Challenges and solutions, Outcomes and Impact, and conclusion.

Out of 7 projects 1 project is about **word suggestion,** which raises error when a wrong word is entered and provides suggestions related to the word. Remaining 6 are **Neural machine Translation (NMT)** projects. In the 6 NMT projects we can see a total of 4 different model implementations namely,

* Sequence-to-Sequence model
* Encoder -Decoder with Simple Attention model
* Single Layer Multi Head Transformer model
* Multi-Layer Multi Head Transformer model

1. Background:

Machine Translation is the process of translating text from one language to another using machines (computer software). Machine translation techniques have evolved over time, with NMT being the dominant approach today. The seq2seq model uses Recurrent Neural Network’s (RNN’s) mainly **Long Short-Term Memory** (LSTM). The Encoder Decoder with Attention model also uses LSTM in encoder decoder blocks while Attention provides context from encoder to decoder which is used to improve the performance of the model. **Transformer** model does not use RNN’s, they are only based on attention mechanism. The attention mechanism used in transformers are self-attention, cross attention, casual self-attention.

The NMT projects use **TensorFlow Framework** and Python programming language. **Pandas** library is used to handle data frames for data analysis, data cleaning and data manipulation. Python **Tkinter** module is used to prepare the **GUI** of projects. Requirements text file is manually created. A separate virtual environment is created in anaconda for implementing projects. The environment installed packages are, Python, TensorFlow, Pandas, Ipykernel SpeechRecoginition, Pyaudio, OpenAI-whisper and ffmpeg. (NumPy is a requirement for tensorflow so it is automatically installed while installing tensorflow)

1. Learning Objectives:
2. Understanding RNN’s, LSTM, Embedding Layers
3. Handling Text data for data cleaning, manipulating, preprocessing, and tokenizing
4. Understanding padding and masking
5. Setting up the virtual environment for Project implementation
6. Creating seq2seq model using Embedding Layer and LSTM’s
7. Creating Encoder-Decoder with attention model
8. Implementing Beam Search decoding to improve translation quality
9. Utilizing Tkinter module for creating GUI
10. Creating Single Layer transformer model
11. Creating Multi Layer transformer model
12. Method used for transcribing audio to text.
13. Activities and Tasks:
14. **Project\_1**

* **Task:**

Load a pre-trained LSTM-based NMT model and use it to translate a sentence from one language to another.

* **Activity:**

The data was obtained from English French model used in Null Class source code. Data is already clean so no additional cleaning is done. Data is tokenized & padded using Keras preprocessing module. English vocabulary size is 200 and French vocabulary size is 345. Inputs are sent into embedding layer of size 128 then to 128 LSTM units (bidirectional, so total 256) and finally dense layers. Model is compiled using Adam optimizer, Sparse Categorical Cross Entropy Loss and accuracy metrics. The model has achieved train and validation accuracy of **97%**. The GUI is prepared using Python Tkinter module

1. **Project\_2**

* **Task:**

Implement beam search decoding for an NMT model to improve translation quality.

* **Activity:**

The project is implemented for **English to Spanish** translation. Data is obtained from Kaggle. Data is vectorized using Keras Layers module with maximum vocab size of 12000, and a ragged batch (i.e. varying sequence lengths based on each batch). The implemented model is Encoder-Decoder with simple attention model. The encoder decoder block has embedding of 128 dimensions and 128 LSTM units. The cross-attention layer has 1 head and 128 key dims. Model is compiled using Adam optimizer, Sparse Categorical Cross Entropy Loss and accuracy metrics. The model has achieved **89%** trained accuracy and **86%** validation accuracy. Instead of using **greedy decoding** **beam search decoding algorithm** is implemented to improve translation quality. The GUI is prepared using Python Tkinter module

1. **Project\_3**

* **Task:**

Create a feature to translate the language from French to Tamil and it should translate if the French word has only five letters if word has more than five letters or less than five letters the model should not translate the word.

.

* **Activity:**

The project is implemented for **French to Tamil** translations. Data is obtained by converting English sentences to French and Tamil using **Google Translations**. Data is manipulated so that it is biased towards 5 letter French words and their translations. Data is vectorized using Keras Layers module with maximum vocab size of 10000 for French and maximum vocab size of 12000 for Tamil, and for sequence length a variable **ragged tensors** are used (i.e. varying sequence lengths based on each batch).

The implemented model is Encoder-Decoder with simple attention model same as above. The model has achieved **94%** training accuracy and **85%** validation accuracy. For decoding the sentence simple greedy decoding approach is used. The GUI is prepared using python Tkinter module. After getting input sentence for translation in GUI, the sentence is filtered to obtain French words that have five letters, other words greater or less than five letters are not translated.

1. **Project\_4**

* **Task:**

Create a feature to throw an error if we enter wrong word, and provide some suggestion if we enter wrong word twice continuously it should show the list of wrong words which we enter so far in the error notification as well as some suggestions related to the word.

* **Activity:**

The vocabulary of this project is obtained from git hub. The vocab contains approx 300000 words. Several operations for a word like delete a letter, insert a letter, replace a letter, switch a letter is implemented to form set of words. One edit operation is applying all four operations to the required word. Two edit operation is applying one edit operation to all the words that are obtained from applying one edit operation one time to the main word. The GUI is prepared using python Tkinter module. If we enter correct word the interface tells your word exist in the vocab. If we enter the wrong word, the word undergoes one edit, two edit algorithms and necessary operations to the applied word to provide suggestions related to the correct word. The GUI is prepared according to the task requirements.

1. **Project\_5**

* **Task:**

Create a feature to translate the language with a combination of two languages at the same time. We should be able to convert the 2 different languages at the same time. translate English to French and Hindi at the same time. This model should work only for 10 letter English words. If we enter below 10 letters or above 10 letters it should not work

* **Activity:**

The project is implemented for **English to French** (EF) and **English to Hindi** (EH) translation. Data is obtained from Kaggle. Data is vectorized using Keras Layers module. The maximum vocab size is set for 15000 for all vocabularies (i.e. English, French vocabulary for EF model, English, Hindi vocabulary for EH model). The sequence length for all tensors for all batches is fixed (i.e. Not ragged batch). For EF model the fixed sequence length is 20 and for EH model the fixed sequence length is 50. The implemented model for both EF and EH translation data is Single Layer Multi Head Attention model. The model contains Positional Embedding (both token & position) Layer for embedding inputs and outputs. The Encoder and Decoder Layer is implemented with respective attention mechanisms like self-attention, cross attention, casual self-attention and a fully connected feed forward network (**refer transformer model**). The model embedding dimension is 100, the dense units in feed forward network is 256, and attention heads is set to 2. The EF model achieved training and validation accuracy of **90%** while the EH model achieved training and validation accuracy of **77%**. The GUI is prepared using Tkinter module. The combo box widget in tkinter is used for selecting the language from French and Hindi.

1. **Project\_6**

* **Task:**

Create a feature to translate the English word to Hindi and it should not translate if the English word starts with vowels and other words it should convert. If we enter a English word starts with Vowels it should show an error message as This word starts with Vowels provide some other words and this model should be able to convert English word starts with vowels around 9 PM to 10 PM.

* **Activity:**

The project is implemented for **English to Hindi** translations. Data is obtained from Kaggle. Extensive data cleaning and manipulation is done to prepare the data for vectorization. Keras layers module is used for vectorization, the maximum vocab size is set for 15000, whereas english vocabulary is around 10000 words and hindi vocabulary is around 11000 words. The sequence length for all tensors is fixed to 25. The implemented model is Multi Layer Multi Head Attention model. The model contains 2 layers, 2 heads, 128 embedding dimensions, and 512 dense units, The model is compiled using Adam optimizer with custom schedule Learning rate, Masked Sparse Categorical Cross Entropy Loss for loss parameter and Masked Accuracy for metrics. The model has achieved **82%** training and **74%** validation **masked accuracy**. When compiled using normal loss and accuracy the model achieved **94%** training and validation **accuracy**. Beam search decoding algorithm is used to improve Translation quality. The GUI is prepared using Tkinter module. After getting Input sentence from the user the input sentence is filtered to remove the words starting with vowels, if any words starting with vowels is present a warning message is shown. The app works normally around 9pm to 10pm. The model is trained with almost known vocabulary. So, there is little scope for Out of Vocabulary words. So OOV words does not produce [UNK] token instead most of the time it produces random translation.

1. **Project\_7**

* **Task:**

Create a feature to translate the audio into Hindi. The system will listen the English audio from user and it will convert into Hindi word. If the system does not understand the audio, it will ask repeat one more time to make it better. The audio should be in English word only. This translation feature work on only after 6 PM IST timing and before that it should show message like please try after 6 PM IST as well as it should not translate any english which is start with M and O apart from that it should translate all other words.

* **Activity:**

The project is implemented for English to Hindi translations. The data and model implementation are exactly as project\_6. SpeechRecognition, Pyaudio, OpenAI-whisper open source modules are used. SpeechRecognition and Pyaudio is used to take audio input using system microphone. OpenAI-whisper library is used to transcribe the speech to text. The transcribed text is filtered to remove words starting with M and O. This audio feature works only after 6 PM IST.

1. Skills and Competencies:

* Proficiency in Tkinter to create Graphical User Interface (GUI)
* Knowledge of LSTM, Embeddings, Beam Search algorithm, Attention and Transformer.
* Proficiency in Python, NumPy, Pandas, TensorFlow to implement models for NMT.
* Knowledge of SpeechRecognition & OpenAI-Whisper to implement audio and speech to text feature

1. Challenges and Solutions:
2. **Virtual Environment**

* **Challenge:**

Installing Virtual Environment to run your projects is a tedious task. To get starting my project I installed some random versions while creating my virtual environment. After some corrections the project started running smoothly, but they are not working on Kaggle so I need to reinstall packages in the environment.

* **Solution:**

A solution is provided below to create a virtual environment according to my project requirements. If versions are not specified, please refer to requirements.txt file in the project.

**Conda create -n tf-env python=3.10.14**

**Conda activate tf-env**

**Pip install tensorflow==2.16.1**

**Pip install ipykernel**

**Pip install pandas**

For project 7 Additional installments are required.

**Pip install SpeechRecognition**

**Pip install pyaudio**

**Pip install openai-whisper==20231117**

**Conda install -c conda-forge ffmpeg**

1. **Preparing Data**

* **Challenge:**

Procuring data and cleaning data are very important steps for training a model, because the data directly effect the accuracy of the model. Project 3 & 5 does not contain the required data on internet

* **Solution:**

For project 3 the initial translations are made using Google Translation. The data is manipulated in a way which enriches the 5 letter words in the French sentences. Similarly

In project 6 the data is manipulated to enrich 10 letter English words.

1. **Custom Models**

* **Challenge:**

Encoder-Decoder with Attention, single layer transformer, multi-layer transformer are custom models. While starting of internship I have zero knowledge on how to implement them.

* **Solution:**

Extensive research is done on the models to implement them**. Null Class** Training source code, **Coursera** NLP specialization, Transformer model on **TensorFlow** page are most referred sources to create the models.

1. **Creating Interface**

* **Challenge:**

Creating Interface to meet the project requirements is a bit challenging because knowledge of different widgets in tkinter is must.

* **Solution:**

Understanding various widgets applications. Identifying required widgets necessary for implementing the algorithm according to project requirements.

1. Outcomes and Impact:

The projects successfully shown various data preprocessing techniques and machine learning models in implementing machine translations. It also showed implementation of numerous user interfaces required for different projects. with this offline translation application in your computer, you can have basic communication with French, Spanish, Tamil and Hindi speaking people.

1. Conclusion:

The project provided a real time translation app for you to communicate in other languages. There is still a lot of scope in the projects like increasing data quality and quantity to train on more sentences and more vocabulary, Tokenizing using byte-pair encoding, increasing the transformer layers and heads, adapting to new transformer XL model etc. As stated above NMT being the most dominant approach today also have a lot of scope in developing new models required for Natural Language Processing (NLP).