Robust Neural Language Translation Model Formulation using Seq2seq approach

Meenu Gupta, Prince Kumar

Chandigarh University, INDIA
Emails: meenu.e9406@cumail.in; cu.17bcs1711@gmail.com

***** Correspondence: meenu.e9406@cumail.in

Abstract

In this work, the approach used is to sequence powerful models that have achieved excellent performance on language translation encoding-decoding tasks. A language transformer model is used in this work based on the sequence-to-sequence approach, which uses a Long Short-Term Memory (LSTM) to map the input sequence to a vector of fixed dimensionality. Then another deep LSTM decodes the target sequence from the vector. Evaluated the model efficiency through BLEU score and LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty with long-short of sentences. This work performed the deep LSTM setup English-Japanese translation accuracy at an order of magnitude faster speed, both on GPU and CPU. The variety of the data is introduced into it to evaluate the robustness using the BLEU score. Finally, a better result is achieved by merging the two different types of datasets and getting the highest BLEU score of 40.1 at the end.

Keywords: LSTM; GPU; BLEU; RNN; NNLM; NLP

1. Introduction

This work aims to make a machine translation that is designed to translate information sources from a variety of natural languages into the target languages and analyze its accuracy from the BLEU score. Due to the growing need for international communication, multilingual machine translation. With the advent of technology, computer programs can replace human that is experts in many fields. One of the most popular domains by artificial intelligence (AI) investigators. AI gives birth to too many systems where the system can be used to function as a human expert. Indigenous language analysis (NLP) is a learning program from decades back to overcome communication barriers mainly due to regional linguistic diversity. The machine translation (MT) system is most commonly found in a particular language text.

Sequence to sequence learning has been successful in many tasks such as machine translation, speech recognition [1], and text summarization, amongst others. To date, the dominant approach encodes the input sequence with a series of bi-directional recurrent neural networks (RNN). It generates a variable-length output with another set of decoder RNNs, both of which interface via a soft-attention mechanism [2]. In machine translation, this architecture has been demonstrated to outperform traditional phrase-based models by large margins. Convolutional neural networks are less common for sequence modeling, despite several advantages [3], which precisely control the maximum length of dependencies to be modeled. Convolutional networks do not depend on the computations of the previous time step and therefore allow parallelization over every element in a sequence. It contrasts with RNNs, which maintain a hidden state of the entire past that prevents parallel computation within a sequence.

DOI: https://doi.org/10.54216/FPA.050203

Fixing the number of non-linearities applied to the inputs also eases learning. Recent work has applied convolutional neural networks to sequence modeling and introduces recurrent pooling between a succession of convolutional layers that tackle neural translation without attention. However, none of these approaches has demonstrated improvements over the state-of-the-art results on large benchmark datasets [4].

In this work, the machine translations are developed with the aid of the transformer model using seq2seq for the translation of Japanese sentences into the English language. The transformers have been the dominant architecture of NLP and can be used to get to the most recent results for each of the various tasks, and it looks like this if they are to be used shortly. Recurrent neural networks are very slow to learn, and without it the LSTM center, the model does not have to be very suitable. However, in the LSTM center of the model, the training will be a lot slower. It has been found that the Seq2Seq model is just as in the base model, but the execution was inferior. To increase the productivity of a transformer a model is used for this work.

2. Literature Survey

There is a significant demand for document conversion from one language to another language. There are several ways to work on applications of neural networks to machine translation. This study has gone through many procedures for machine translations and found the simplest and most effective way of applying an RNN-Language Model [5], the Feedforward Neural Network Language Model, to a Machine Translation task. A number of the best lists of a strong MT baseline reliably improve translation quality. More recently, researchers have begun to look into ways [6] they incorporated their NNLM into the decoder of an MT system and used the decoder's alignment information to provide the NNLM with the most valuable words in the input sentence. Examples of this work include a similar approach, which combines an NNLM with a topic model of the input sentence, which improves rescoring performance. Their approach was highly successful, and it achieved significant improvements over their baseline. Similar to this work, [7] used an LSTM-like RNN architecture to map sentences into vectors and back. However, their primary focus was on integrating their neural network into an SMT system.

Likewise, [8] attempted to address the memory problem by translating pieces of the source sentence to produce smooth translations, which is similar to a phrase-based approach. In this work, the authors achieve similar improvements by simply training their networks on reversed source sentences. Direct translations with a neural network that used an attention mechanism to overcome the poor performance on long sentences experienced. In []9], the authors first map the input sentence into a vector and then back to a sentence, although they map sentences to vectors using convolutional neural networks, which lose the ordering of the words.

3. Material and Methods

3.1. Dataset used

The English to Japanese language dataset used for this analysis is collected from Kaggle. First of all, *the Japanese English corpus* dataset is used that is collected from Kaggle [10] and converted into a data frame. Similarly, a different dataset, "*anki*" is collected from a website named ManyThings.org [11]. It is a normal daily life conversation of Japanese, and the proposed model is trained with this dataset. Again the dataset is converted into a data frame that is merged as an extensive fruitful dataset.

Further, the proposed models are trained on a subset of the data and display the data. A total of 114458 records were in the collected dataset. A translation task and specific training set/subset are used due to the public availability of tokenized training. Typical neural language models rely on a vector representation for each word, and the work used a fixed vocabulary for both languages. This dataset mainly deals with traditional Japanese culture, religion, and history. The dataset didn't have any daily life conversations or ordinary words that Japanese people use frequently. It was all about government offices, festivals, etc.

3.2. Data Preprocessing

A spacy library is used for tokenization of English and Japanese language. S***pacy*** is a free open source library for Natural Language Processing in Python. Spacy is the best way to prepare a text for deep learning. It interoperates

DOI: https://doi.org/10.54216/FPA.050203

seamlessly with TensorFlow, PyTorch, sci-kit-learn, Gensim, and the rest of Python's excellent AI ecosystem. With ***spacy***, you can easily construct linguistically sophisticated statistical models for various Natural Language Processing problems.

When a normal tokenizer is used, it would not have been recognized as there are no spaces between them. But spacy library tokenizes into each meaningful word. The below libraries are needed to be installed for preprocessing of the data:

- Spacy
- sudachipy sudachidict_core
- torchtext==0.6.0
- spacy[ja]
- spacy download en_core_web_sm

Next, the three different CSV files are created in the following steps. The collected dataset was split into train 60%, Validate 20%, and testing 20%. Hence, for training, 68674 records, for validation, 22892 records, and for testing, 22892 records were considered.

3.3. Proposed Models

As a matter of fact, Recurrent Neural Networks (RNN) (or more precisely LSTM/GRU) are very efficient in calculating and analyzing complex sequence-related circumstances on a tremendous large amount of data. They have real-time applications in speech recognition, Natural Language Processing (NLP) problems, time series forecasting, etc. Thus the transformational model that is the Seq2seq model is used for this analysis. Sequence to sequence (often abbreviated to seq2seq) models is a particular class of Recurrent Neural Network architectures typically used (but not restricted) to solve complex Language related problems like Machine Translation, Question Answering, creating Chatbots, Text Summarization, etc. [12]. This blog post aims to give a detailed explanation of how sequence models are built and give an intuitive understanding of how they solve these complex tasks. This work consists of Machine Translation (translating a text from one language to another, in our case from English to Japanese) as the running example in this blog. However, the technical details apply to any sequence-to-sequence problem in general. Since Neural Networks are used to perform Machine Translation, it is called Neural Machine translation (NMT). Similar to the Convolutional Sequence-to-Sequence model, the Transformer does not use any recurrence. It also does not use any convolutional layers. Instead, the model is entirely made up of linear layers, attention mechanisms, and normalization. Recurrent neural networks are very slow to train, and without LSTM, the model is not very accurate. But with LSTM, the model makes it much slower to train. First, seq2seq is used as the baseline model, but since it doesn't do parallel computing and no GPU is used, it is switched to the transformer model, which is much faster than the seq2seq model. In the seq2seq model, the words are passed to the encoder sequentially, and there is no use of GPU there. So to do parallel computing for the language translation, it moved with transformers.

3.4. Training Details

- First, the tokens are passed through the standard embedding layer. Next, as the model has no recurrent, it has no idea about the order of the tokens within the sequence. The problem is solved by using a second embedding layer called a positional embedding layer. The following function after the Embedder is the Position Encoder.

DOI: https://doi.org/10.54216/FPA.050203

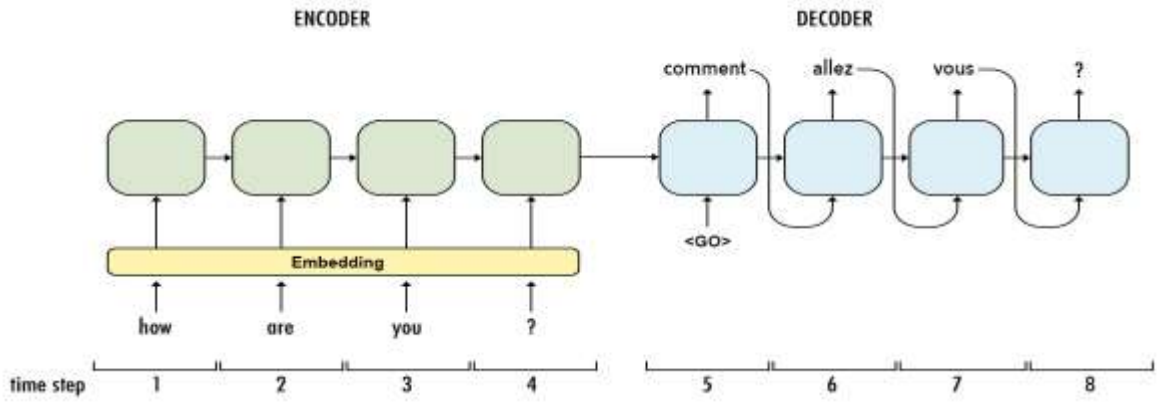

Figure 1: Encoder and Decoder technique

- As mentioned above, since our model contains no recurrence and no convolution, for the model to make use of the order of the sequence, and must inject some information about the relative or absolute position of the tokens in the sequence. Eq. (1) and Eq. (2) show the numerical analysis of the proposed model.

$$(PE(positios,2i) = sin(pos100002i/dmodel), \quad Eq.\ (1)$$

$$PE(position,2i+1) = cos(posistion100002i/dmodel).\ with\ dmodel = 512) \quad Eq.\ (2)$$

The Positional Encoder discussed [13]

- The input mask is simply the same shape as the input sentence but has a value of 1 when the token in the source sentence is not a token and 0 when it is a token. It is used in the encoder layers to mask the multi-head attention mechanisms used to calculate and apply attention over the source sentence, so the model ignores tokens, which contain no useful information.

- First, it passes the input sentence and mask into the multi-head attention layer, performs dropout on it, and passes it through a Layer Normalization layer.

- The encoder layer uses the multi-head attention layer to attend to the input sentence, i.e., it is calculating and applying attention over itself instead of another sequence.

- Multi-head attention means creating many attention vectors for each word, and the Wz weight will choose which attention vector to take. (Multiple attention vector for one word) And the rest of the things in the Attention model are regular, like Feed Forward Neural Network.

- The objective of the decoder is to take the encoded representation of the source sentence and convert it into predicted tokens in the target sentence. Further, it compares with the actual tokens in the target sentence to calculate our loss, which is used to calculate the gradients of our parameters and then use Adam optimizer to update our weights to improve our predictions.

- The decoder is similar to an encoder. However, it has two multi-head attention layers. A masked multi-head attention layer over the target sequence and a multi-head attention layer that uses the decoder representation as to the query and the encoder representation as to the key and value.

4. Experimental Result Analysis

After having trained models, it implemented them on the testing set and evaluated the performance by averaging the BLEU scores for all the sentences. There is only simple conversation used in daily life in Japanese and trained

DOI: https://doi.org/10.54216/FPA.050203

transformation model with this dataset. Table 1 contains testing results. It is shown that the score of English to Japanese translation. Especially. The best-performing model is English to Japanese when it merged the dataset to make a model more effective with a test BLEU score above 40.1.

Table 1: Model Evaluation using BLEU scores

Models	**Dataset**	**BLEU Score**
Baseline Model	D1	4.87
Transformer Model	D2	59
Transformer Model	D1	15
Transformer Model	Merged D: D1+ D2	40.1

The analysis is performed on an English-to-Japanese translation task; the transformer model performed the best previously reported models on a different dataset and got BLEU to score of 4.87 on the baseline model using dataset1. Similarly, the model on datasets using the transformer seq2seq model got many variations between the BLEU score of the different datasets. The model gets a good score in one of the datasets that were much more effective in our model. Next, it is experimented with to merge both datasets to get the trained model more robustly. Finally, we achieved the maximum BLEU score after merging both datasets, and then the model got trained well, and the perplexity score was 1.2. below are graphs depicting the loss and perplexity that are getting changed during the training of the model.

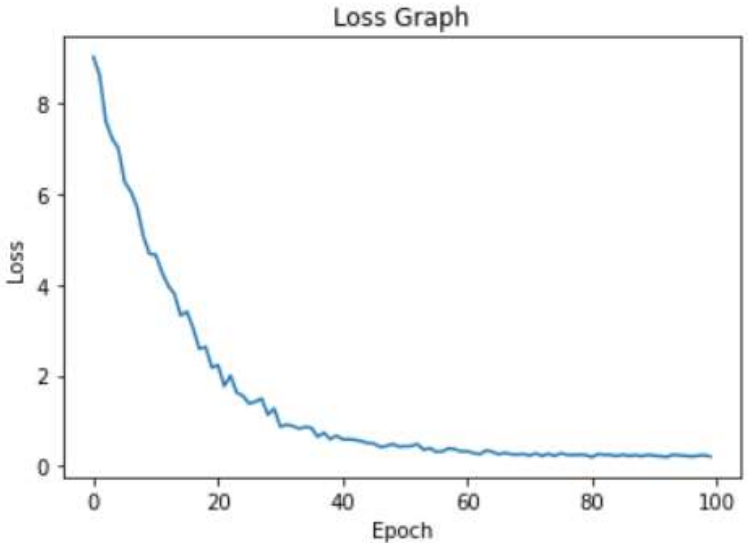

Figure 2: Loss Visualization

DOI: https://doi.org/10.54216/FPA.050203

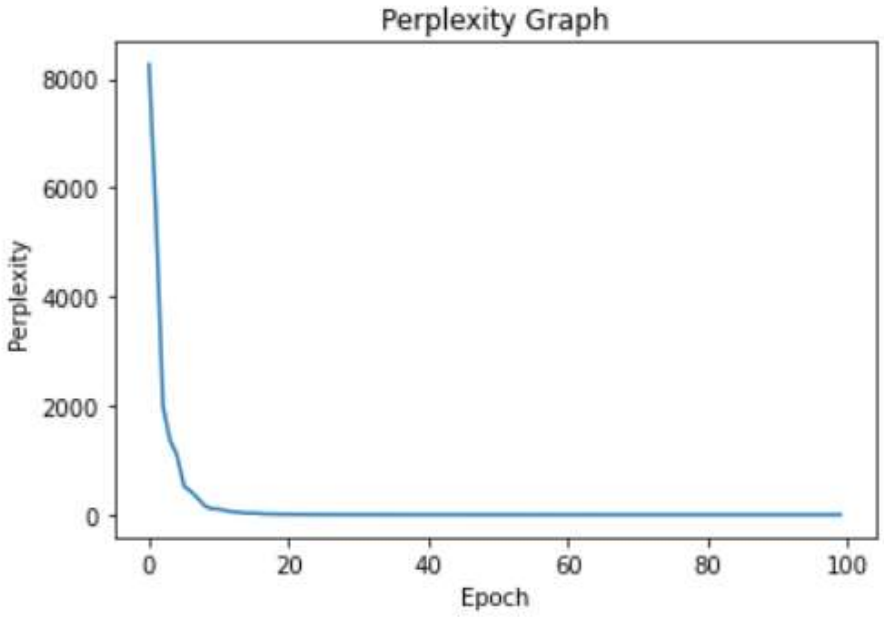

Figure 3: Perplexity Visualization.

5. Conclusion and future scope

This approach presented the sequence to sequence transformer model, the first sequence transduction model using the LSTM approach, most commonly used in encoder-decoder architectures with multi-headed self-attention. For Machine neural translation tasks, the transformer model can be trained significantly faster than the previous direct RNN. On English-to-Japanese translation tasks that help to achieve a good BLEU score. Evaluated the model efficiency through BLEU score, and LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty with long-short of sentences.

REFERENCES

[1] Sutskever, Ilya, Martens, James, Dahl, George E., and Hinton, Geoffrey E. On the importance of initialization and momentum in deep learning, ICML, 2013.

[2] Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua, Neural machine translation by jointly learning to align and translate, arXiv preprint at Xiv:1409.0473, 2014.

[3] Waibel, Alex, Hanazawa, Toshiyuki, Hinton, Geoffrey, Shikano, Kiyohiro, and Lang, Kevin J. Phoneme Recognition using Time-delay Neural Networks. IEEE transactions on acoustics, speech, and signal processing, 37(3):328–339, 1989.

[4] Jonas Gehring 1 Michael Auli 1 David Grangier 1 Denis Yarats 1 Yann N. Dauphin 1, Sequence to Sequence Learning with Neural Networks, neurips, 2014.

[5] Mikolov, M. Karafiat, L. Burget, J. Cernock ´ y, and S. Khudanpur. Recurrent neural network-based language model. In INTERSPEECH, pages 1045–1048, 2010.

[6] Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation, ACL, 2014.

DOI: https://doi.org/10.54216/FPA.050203

[7] Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation, arxiv preprint arXiv:1406.1078, 2014.

[8] Pouget-Abadie, D. Bahdanau, B. van Merrienboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. arXiv preprint, arXiv: 1409.1257, 2014.

[9] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for wmt-14, WMT, 2014.

[10] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A Key Management Scheme for Secure Communications of Information Centric Advanced Metering Infrastructure in Smart Grid," **IEEE Transactions on Instrumentation and Measurement**, vol. 64, no. 8, pp. 2072-2085, August 2015.

[11] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, A. Shalaginov, "Deep Graph Neural Network-based Spammer Detection Under the Perspective of Heterogeneous Cyberspace", Future Generation Computer Systems, https://doi.org/10.1016/j.future.2020.11.028.

[12] L. Tan, H. Xiao, K. Yu, M. Aloqaily, Y. Jararweh, "A Blockchain-empowered Crowdsourcing System for 5G-enabled Smart Cities", Computer Standards & Interfaces, https://doi.org/10.1016/j.csi.2021.103517

[13] N. Shi, L. Tan, W. Li, X. Qi, K. Yu, "A Blockchain-Empowered AAA Scheme in the Large-Scale HetNet", Digital Communications and Networks, https://doi.org/10.1016/j.dcan.2020.10.002.

DOI: https://doi.org/10.54216/FPA.050203