AIML-Group8

Capstone – Project synopsis

1.0 Abstract

Machine Translation is the automated translation of source material into another language without human intervention. Neural Machine Translation (NMT) has achieved state-of-the-art results. However, one of the main challenges that neural MT still faces is dealing with very large vocabularies and morphologically rich languages.

The prevalent approach to neural machine translation relies on sequence to sequence encode the source sentence. We present a faster and simpler architecture based on a succession of convolutional layers as attention-based LSTMs. This allows to encode the source sentence simultaneously compared to recurrent networks for which computation is constrained by temporal dependencies.

Despite the recent successes of NMT applications, a predefined vocabulary is still required, meaning that it cannot cope with out-of-vocabulary (OOV) or rarely occurring words. We extend beyond the current sequence to sequence backbone NMT models to the attention-based models in which the source and target sentences are aligned with each other. Our proposed topology shows consistent improvements of attention-based sequence to sequence model over basic models, German↔English.

This paper mainly focusses on language translation between European language pairs. We need to share tasks for the language pair EN→DE and DE→EN. Our objective is to design a Machine Translation model that can be used to translate sentences from English language to German language, and vice versa for best performing model.

Key words: LSTM, Translation, NMT**,** Sequence, Attention, Embeddings,

2.0 Introduction

Machine Translation is the automated translation of source material into another language without human intervention. The database comes from ACL2014 Ninth workshop on Statistical Machine Translation. This paper mainly focusses on language translation between European language pairs. We need to create a shared task for the language pair EN→DE. The idea behind the exercise is to provide the ability for two parties to communicate and exchange ideas from different countries with better performing models in recently researched algorithms of NMT technique.

3.0 Literature survey

In analysis of recent research for project, we found the randomness of neural networks leads to standard neural network machine translation models unable to effectively reflect the linguistic dependencies and having unsatisfactory results, when dealing with long sentence sequences. while other translation models have problems of vanishing and exploding gradients, training, Slow and complex training procedure, difficult to process longer sequences.

To solve these problems, a new neural network machine translation model with entity tagging improvement is proposed. First, for the low-frequency word translation problem, UNK entity tags replacement is used to compensate for the weakness of the randomness of neural networks and the encoding/decoding strategy of entity tagging is improved. Attention model has ability to handle such randomness along with entity tags, which help in weighting the weights in model.

3.1. Paper uniqueness and objective:

We are uniquely attempting in this experiment to evaluate performance of basic translation methods such as **S**tatistical **M**achine **T**ranslation Vs **N**eural **M**achine **T**ranslation along with attention mechanism. Observing the key positives based on accuracy estimations in both translation methods is our first goal. In a second step for **NMT**, we will evaluate the advanced algorithms recently introduced in sequence-to-sequence models i.e., LSTM/ LSTM (with Attention) and BiLSTM. Based on Evaluation parameters we are planning to publish the best performing algorithms for translations. Also explain key parameters which fine tune the results in both the algorithms.

3.2 Language Translation methods and techniques:

Statistical Machine Translation (**SMT**) has been the dominant translation paradigm for decades.

Practical implementations of SMT are generally phrase-based systems (PBMT) which translate sequences of words or phrases where the lengths may differ. Even prior to the advent of direct **N**eural **M**achine **T**ranslation, neural networks have been used as a component within SMT systems with some success. Perhaps one of the most notable attempts involved the use of a joint language model to learn phrase representations which yielded an impressive improvement when combined with phrase-based translation. This approach, however, still makes use of phrase-based translation systems at its core, and therefore inherits their shortcomings.

4.0 Algorithms and Models used:

LSTM is a special type of recurrent neural network. Specifically, this architecture is introduced to solve them, i.e. problem of vanishing and exploding gradients. This type of network is better for maintaining long-range connections,
recognizing the relationship between values
at beginning and end of a sequence.

The main reason is that every component of an input sequence has information from both the past and present. For this reason, *LSTM with attention* can produce a meaningful output, combining LSTM layers considering directions.

BiLSTM will have a different output for every component (word) of the sequence (sentence). We will use standard word embedding techniques to vectorize each sentence word, and then set the weights.

4.1 Model Architecture

Our model (see Figure 1) follows the common sequence-to-sequence learning framework [41] with attention [2]. It has three components: an encoder network, a decoder network, and an attention network. The encoder transforms a source sentence into a list of vectors, one vector per input symbol. Given this list of vectors, the decoder produces one symbol at a time, until the special end-of-sentence symbol (EOS) is produced. The encoder and decoder are connected through an attention module which allows the decoder to focus on different regions of the source sentence during decoding

.

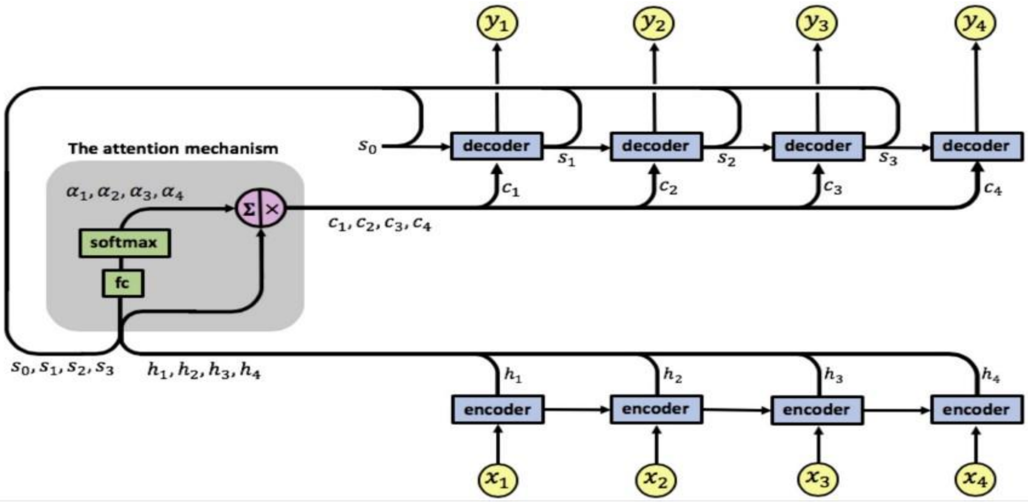
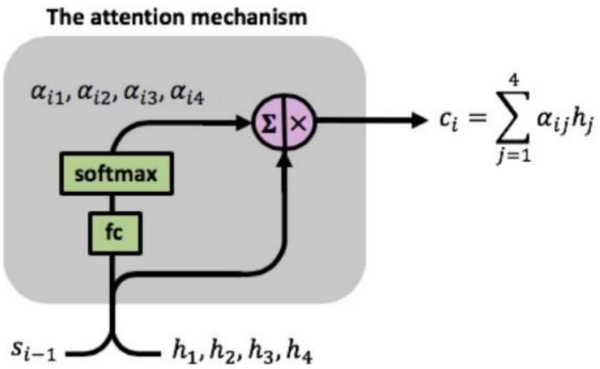


Figure 1

RNNs with an attention mechanism

Our attention model has a single layer RNN encoder, again with 4-time steps. We denote the encoder's input vectors by x1, x2, x3, x4 and the output vectors by h1, h2, h3, h4. The attention mechanism is located between the encoder and the decoder, its input is composed of the encoder's output vectors h1, h2, h3, h4 and the states of the decoder s0, s1, s2, s3, the attention's output is a sequence of vectors called context vectors denoted by c1, c2, c3, c4. Attention weights are learned using the attention fully connected network and a SoftMax function:


The attention mechanism
$\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4}$
softmax
fc
Σ ×
$c_i = \sum_{j=1}^{4} \alpha_{ij} h_j$
s_{i-1}
h_1, h_2, h_3, h_4

1.1. **Input Layer**: Input sequences are fed into the model with one word for every step. Each word is encoded as a unique integer or one-hot encoded vector that maps to the English dataset vocabulary.

1.2. **Embedding Layers:** Embeddings are used to convert each word to a vector. The size of the vector depends on the complexity of the vocabulary. Words with similar meanings occupy similar regions of this space; the closer two words are, the more similar they are.

1.3. **A. Recurrent Layers (LSTM)**: This is where the context from word vectors in previous time steps (LSTM) is applied to the current word vector (input gate, forget gate, output gate).

1.4. **B.Attention layer:** Attention mechanism is employed to give different focus to the information output from the hidden layers of LSTM. Finally, the softmax classifier is used to classify the processed context information. Attention based LSTM was able to capture both the local features of phrases as well as global sentence semantics. This is achieved by keeping the intermediate outputs from the encoder LSTM from each step of the input sequence and training the model to learn to pay selective attention to these inputs and relate them to items in the output sequence. Attention layers are fundamentally a weighted mean reduction.

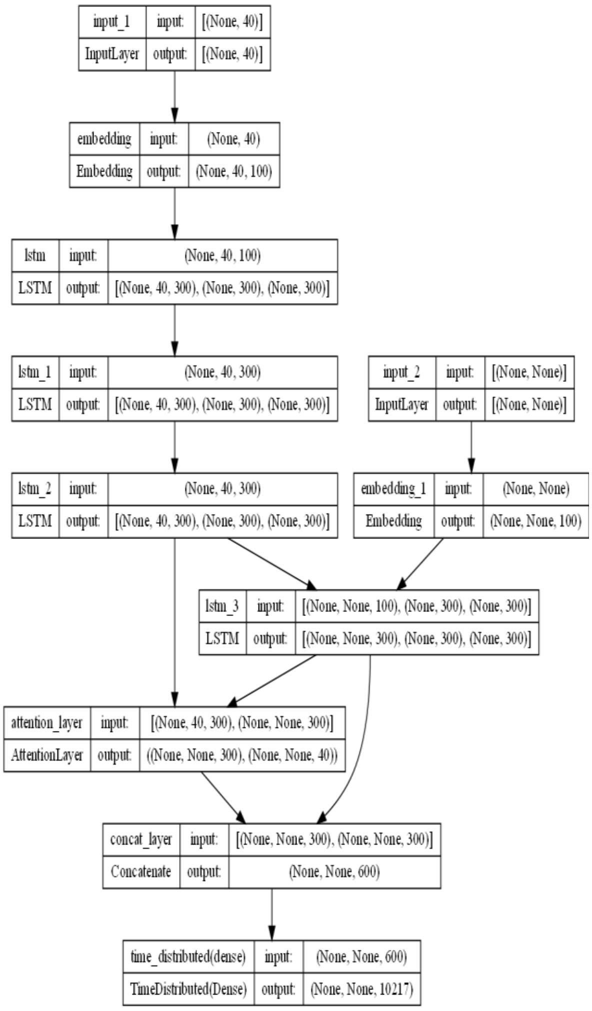
1.5. **Recurrent Layers (BiLSTM-Repeat Vector)**: This is where the context from word vectors in previous time steps is applied to the current word vector.

1.6. **Dense Layers (Decoder)**: These are typical fully connected layers used to decode the encoded input into the correct translation sequence.

1.7. **Output layer:** The outputs are returned as a sequence of integers or one-hot encoded vectors which can then be mapped to the German dataset vocabulary. The output layer is application specific. The output layer holds the result or the output of the problem. The activation function for the output layer may be different than the hidden layers based on the problem.

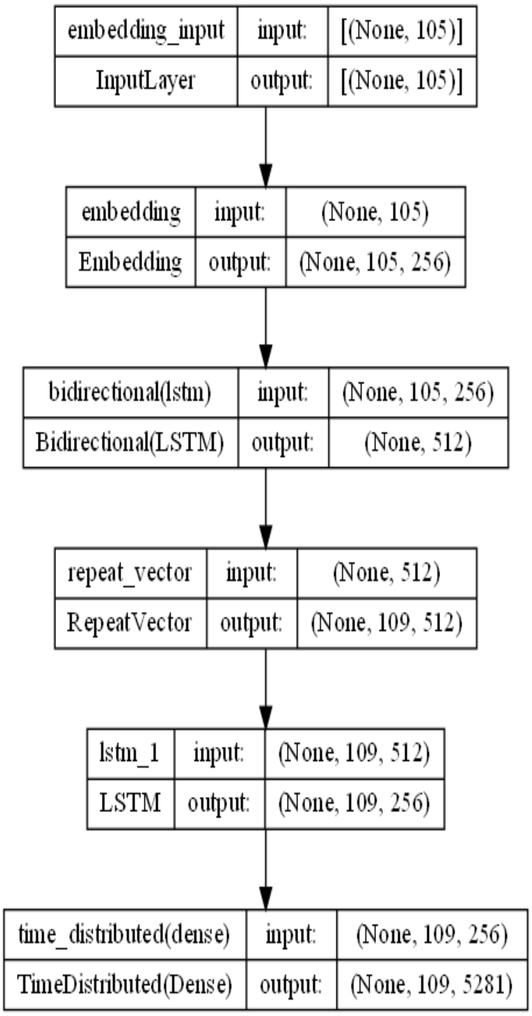
Following is the pictorial representation of Model attempted with attention layer.

LSTM With Attention



input_1 | input: | [(None, 40)]
InputLayer | output: | [(None, 40)]
embedding | input: | (None, 40)
Embedding | output: | (None, 40, 100)
lstm | input: | (None, 40, 100)
LSTM | output: | [(None, 40, 300), (None, 300), (None, 300)]
lstm_1 | input: | (None, 40, 300)
LSTM | output: | [(None, 40, 300), (None, 300), (None, 300)]
input_2 | input: | [(None, None)]
InputLayer | output: | [(None, None)]
lstm_2 | input: | (None, 40, 300)
LSTM | output: | [(None, 40, 300), (None, 300), (None, 300)]
embedding_1 | input: | (None, None)
Embedding | output: | (None, None, 100)
lstm_3 | input: | [(None, None, 100), (None, 300), (None, 300)]
LSTM | output: | [(None, None, 300), (None, 300), (None, 300)]
attention_layer | input: | [(None, 40, 300), (None, None, 300)]
AttentionLayer | output: | ((None, None, 300), (None, None, 40))
concat_layer | input: | [(None, None, 300), (None, None, 300)]
Concatenate | output: | (None, None, 600)
time_distributed(dense) | input: | (None, None, 600)
TimeDistributed(Dense) | output: | (None, None, 10217)

BiLSTM_Recurrent



embedding_input | input: | [(None, 105)]
InputLayer | output: | [(None, 105)]
embedding | input: | (None, 105)
Embedding | output: | (None, 105, 256)
bidirectional(lstm) | input: | (None, 105, 256)
Bidirectional(LSTM) | output: | (None, 512)
repeat_vector | input: | (None, 512)
RepeatVector | output: | (None, 109, 512)
lstm_1 | input: | (None, 109, 512)
LSTM | output: | (None, 109, 256)
time_distributed(dense) | input: | (None, 109, 256)
TimeDistributed(Dense) | output: | (None, 109, 5281)

4.2 Data Samples

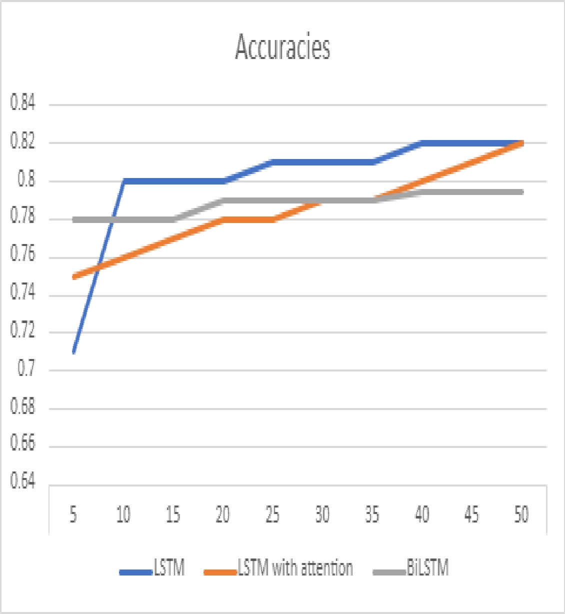
WMT ’14 contains English-German parallel corpora: Europarl (61M words), news commentary (5.5M), crawled corpora 272.5M words respectively, totaling 340M words. We are planning to reduce the size of the combined corpus. We do not use any monolingual data.

After a usual tokenization, we use a shortlist of 9,000 to 16,000 most frequent words in each language to train our models. Any word not included in the shortlist is mapped to a special token ([UNK]). We do not apply any other special preprocessing, such as stemming, replacement etc. to the data.

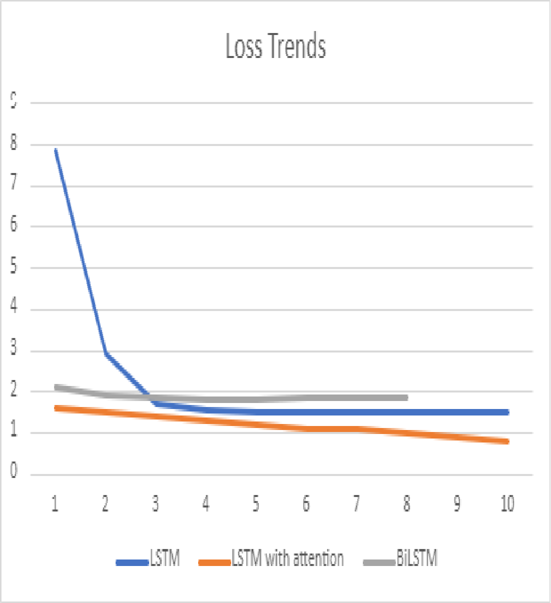
5.0 Results: Model Evaluations

To verify the translation effectiveness of the proposed English machine translation model of LSTM attention, the experimental analysis results are as shown in the below graph.

Model Accuracies

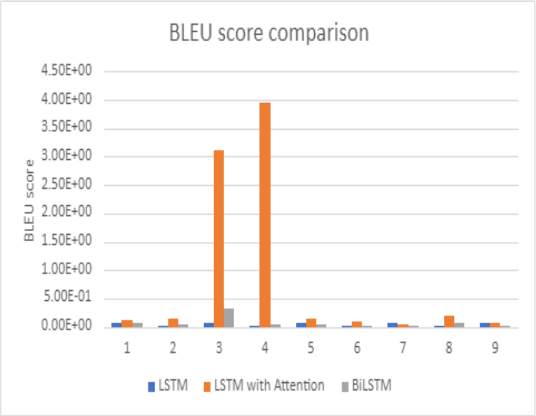


Model Loss Trends



Simple RNN model, we observed an accuracy of 5% and loss of 1.381 on train data and 6% accuracy and 1.406 loss on validation data. This clearly indicates that the model is not making correct translations and the overall performance of the model is very low.

Next, we used the attention mechanism to see if it can improve the performance of the model. We got 80% accuracy and 0.842 loss on train data and 81% accuracy and 1.453 loss on validation data which is way better LSTM than simple RNN model. We have seen a significant improvement in the accuracy of the model, when LSTM model with attention layer is used. LSTM with the attention mechanism does work quite well with Losses as well, when compared with basic LSTM models for sequence-based iterations.



Simple LSTM and Recurrent BiLSTM model have very low score of BLEU, whereas LSTM with Attention mechanism along with word embeddings provides the most appropriate results. Also, in actual translation result shares most closed word translations are of LSTM with Attention, so far in all three models.

4.3 Translation Sample results

English-German Translations (LSTM with Attention)

English original	German predicted	German expected
madam president I do not have a great deal to add on this subject	frau präsidentin ich habe zu diesem thema nicht viel vorab zu bemerken	frau präsidentin ich möchte ich mich sagen daß sie die debatte und frau arbeitsschutzregelunge
i do not know your exact view in this respect mr kinnock but we want commissioners who are politically strong and politically committed	ich weiß nicht herr kinnock wie sie darüber denken aber klar ist daß wir politisch starke und politisch engagierte kommissionsmitglieder wollen	ich denke ich daß die kommission nicht nicht nicht nicht nicht sagte daß wir nicht nicht nicht möglich daß wir nicht nicht zu tun

German-English Translations (LSTM with Attention)

German Original	English Original	English Predicted
ich kann mich natürlich erkundigen und versuchen die erforderlichen informationen einzuholen	of course, i can go back and see whether we can find the necessary information	i think, that the commission is not able to ensure the precautionary information of european union.
dieser ausschuß hatte sich sehr eindeutig für die abschaffung der zentralisierten erteilung des vorherigen sichtvermerks ausgesprochen	that committee was very clear about the need to abolish the centralised ex ante visa	the commission is closed of the european union's party and the european union system of the european union.

English-German Translations (BiLSTM Recurrent)

English original	German Original	German Predicted
mr president my compliments to the rapporteur for his indepth report	herr präsident mein kompliment and den berichterstatter für seinen detaillierten bericht	herr prasident die die die die
he said we must not always be carried along by lawyers	er meinte wir sollten uns nicht immer von juristen beeinflussen lassen	er von die die die die

6.0 Discussion and Conclusions

We experienced how to deal and make progress on real time scenarios in the industry based on our knowledge and study. The major learnings from process are how to work on a problem from planning to collecting the data. Analyzing the data on different phases.

We introduced a simple encoder-decoder based LSTM model for neural machine translation based on convolutional networks. This approach is more parallelizable than recurrent networks and provides a shorter path to capture long-range dependencies in the source. We observed significant gains in terms of BLEU score and translation in the attention-based LSTM model.

The accuracy of machine translation algorithms has been impacted significantly with the size of training datasets, number of epochs and other hyperparameters. It is suitable for any language pair. However, we faced some challenges while running the model. firstly, RAM got exhausted while training when size of the input data is huge. Hence a limitation of a fixed number of records (6000) to train our model. Models take a much longer duration to train with an average of 3 hours to run for 40 epochs subject to variation as we ran the models in Google Colab.

Conclusions:

- Though with limited computational power, one can use the normal LSTM model with additions of word embeddings to explore contextual relationships to some extent, dynamic length of sentences might decrease its performance after some time, if being trained on extensively.
- Thanks to attention-based models, contextual relations are being much more exploited in attention-based models, the performance of the model seems very good as compared to the basic LSTM model. Applications of such models are sentiment analysis, language modelling, neural machine translation, text summarization, speech recognition and much more.
- BiLSTM will have a different output for every component (word) of the sequence (sentence). We attempted to read and learn the sequences with both directions, but without attention mode. Which results in missing attention with context weightages, which leads to unable to achieve concrete results with the limited resources of local/google colab with us.
- Finally, regarding the disadvantages of BiLSTM compared to LSTM, it’s worth mentioning that BiLSTM is a much slower model and requires more time for training and heavy resources. Thus, we recommend using it only if there’s a real necessity.

7.0 References

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3, 1137–1155.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., WardeFarley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2013). Audio chord recognition with recurrent neural networks. In *ISMIR*.

Contribution of linguistic features to automatic machine translation evaluation. In Proceedings of the *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference* on Natural Languages

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014b). On the properties of neural ¨ machine translation: Encoder–Decoder approaches. In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. to appear.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In Association for Computational Linguistics.

Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation. In M. Kay and C. Boitet, editors, *Proceedings of the 24th International Conference on Computational Linguistics (COLIN)*, pages 1071–1080. Indian Institute of Technology Bombay.