Comparison of Architectures for Neural Machine Translation

Saish Desai,
sbdesai2@illinois.edu

Abstract

In this literature review, we will discuss and describe different architectures of Neural Machine Translation and compare their results when tested on shared task of machine translation presented at the Ninth Workshop of Associated Computational Linguistics (ACL) on Statistical Machine Translation (WMT) in the year 2014. The architectures studied in the review focus on the task of English to French translation, wherein an attempt is made to improve the performance of the machine translation task using deep learning to achieve results comparable to a phrase-based statistical translation baseline. The architectures selected for the review have also been a driving force behind Google's translate service.

1 Introduction

The term Machine Translation (MT) was stated by (Dostert, 1957) as *the transference of meaning from one patterned set of signs occurring in a given culture into another set of patterned signs occurring in another related culture by means of an electronic computer*. It is a sub-task of Natural Language Processing (NLP) which involves use of computer to translate textual data from one language to another. Initially introduced as Statistical Machine Translation (SMT), the performance of MT improved overtime with the advent of neural networks and the translation models used today work as Neural Machine Translation (NMT) Models. The models first introduced focused on word-to-word translation, hence were unable to handle the problem of semantic ambiguity. Translation of a word from the source language having multiple meanings is difficult to achieve in absence of the context. Models are needed for translating textual data from source to target language which embed the syntactic and semantic information while learning the translation. In this literature review, we will discuss the evolution of NMT models by first talking about the Recurrent Neural Network (RNN) based encoder-decoder architecture proposed by (Cho et al., 2014b) and (Sutskever et al., 2014), followed by stating the improvements in translation of long sentences with the addition of additive attention mechanism in the model proposed by (Bahdanau et al., 2015). The review eventually talks about the transformer architecture first introduced by (Vaswani et al., 2017) which shifts the focus from the recurrent encoder-decoder architecture and introduces the self-attention mechanism for translation. These models have been trained on WMT 14 English to French translation data set. This data set is a part of the shared task on machine translation presented at the Ninth Workshop of ACL on Statistical Machine Translation in the year 2014. **Section 2** states the problem definition for machine translation and talks about the shared translation task presented during the workshop. **Section 3** describes the corpora and data set selection used for the translation tasks. **Section 4** compares the proposed model architectures by discussing the training and decoding methods implemented in each model to achieve efficient translations. **Section 5** further provides a quantitative comparison between the models using a performance metric called the **BLEU** (Papineni et al., 2002) score and mentions the qualitative improvement using specific translation examples. The review concludes with **Section 6** which consolidates all the comparisons and claims made, followed by the reference section comprising of all the research papers cited.

2 Problem definition

A Machine Translation model should be capable enough for understanding the syntax and semantics of languages under translation. When used for target language prediction, it should be able to map the input and output sentences without loosing any information. Thus, the focus of a typical MT model involves improving the performance of existing

SMT models by training the proposed models on parallel corpora and capturing linguistic features in the learning process. Though used initially for the task of statistical machine translation, the parallel corpora has further been used to train neural network based models which have the capacity to understand complex language structures which can be used during the translation process. Model performance for the task of MT is evaluated in terms of the **BLEU** (bilingual evaluation understudy) score. This evaluation technique was first introduced by (Papineni et al., 2002), in response to human evaluation for machine translation being very tedious and expensive. Papineni et al. (2002) states that -*The closer a machine translation is to a professional human translation, the better it is.* Thus, the BLEU score quantifies the closeness of machine generated translation with a professional human translation.

3 Data Sets

The WMT' 14 English to French data set from *ACL 2014 Workshop on Statistical Machine Translation (WMT)* is one of most popular data sets for implementing state-of-the-art machine translation systems. Being a publicly available data set containing parallel corpora for machine translation, it has been used for implementing end-to-end machine translation models. All the researchers have presented their machine translation results by training their proposed models on English to French translation data set as a part of the shared task for translation from the WMT'14 workshop [1].. There are two types of data sets used for training namely - parallel data comprising of text from source language (English) and its corresponding text from the target language (French), monolingual data for each language (either English or French). The translation process is divided into three parts - train, validation and test. The researchers have generated separate data sets from the corpus provided by the shared translation task organizers. Cho et al. (2014b) has used a traditional statistical machine translation approach for English to French translation, which involves training a translation and language model. The translation model consist of training a RNN encoder-decoder architecture using bilingual corpora containing English and French sentences from some datasets available on the WMT'14 workshop website such as Europarl, News Commentary, UN and Common Crawl. The target (French) language model has been trained on some crawled newspaper articles. A subset of data from the all the above corpora has been selected using methods proposed by (Moore and Lewis, 2010) and (Axelrod et al., 2011). The resulting language modelling corpus consist of 418M words and the translation modelling corpus consist of 348M words. The news-test-2014 data set has been used for testing the model performance. Sutskever et al. (2014) has used neural network for implementing an end-to-end machine translation system for English to French translation. The proposed model has been trained on a data set of 12M sentences containing 348M and 304M words from French and English language respectively. Bahdanau et al. (2015) has used the same set of corpora and data selection process used by (Cho et al., 2014b) for generating the training data set. However, only parallel (bilingual) corpora has been used for the neural language model proposed by (Bahdanau et al., 2015). The news-test-2012 and news-test-2013 corpora provided on the WMT'14 workshop website have been used for validation and news-test-2014 has been used for testing purpose. All the data sets used in (Cho et al., 2014b), (Sutskever et al., 2014) and (Bahdanau et al., 2015) contain a fixed vocabulary for each language comprising of the most occurring words and a special **UNK** for all the out-of-vocabulary words. For English to French translation Vaswani et al. (2017) has also used corpora provided by the WMT'14 workshop for training. The training data set consist of 36M sentences and these sentences are trained in batches with each batch containing sentence pairs of approximately the same length to avoid sparsity issues. Thus, there are 25000 source and target tokens in each batch. To test the model performance, news-test-2014 from workshop website has been used. Each language has a rich and peculiar morphological structure and it is very difficult to have an understanding of every language in depth. For the purpose of machine translation it is more important to know how languages with different syntactic structure convey a common semantic meaning. Most of the textual data used for training and generation of translation so far comes from news articles, proceeding of European Parliament and the United Nations website. These are rich sources of multilingual text and contain a wide variety of translation use cases which go beyond

1. https://www.statmt.org/wmt14/translation-task.html

word-to-word translation. Such corpora can help in improving the performance of machine-learning based models over the rule-base systems.

4 Approaches

4.1 Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

4.1.1 Approach and Architecture

The goal of a statistical machine translation system is to predict a translation **f** for a sentence **e** such that it maximizes the probability **p(f/e)** of the target sentence given the source sentence. To achieve this, (Cho et al., 2014b) has come up with a neural network model for learning the translation from the source to the target language. Though not proposed as an end-to-end machine translation system, the conditional probabilities learned by this model are used as features by the SMT systems to predict translation for a given sentence. The model consist of two recurrent neural networks (RNN) - Encoder and Decoder as shown in Figure 1.

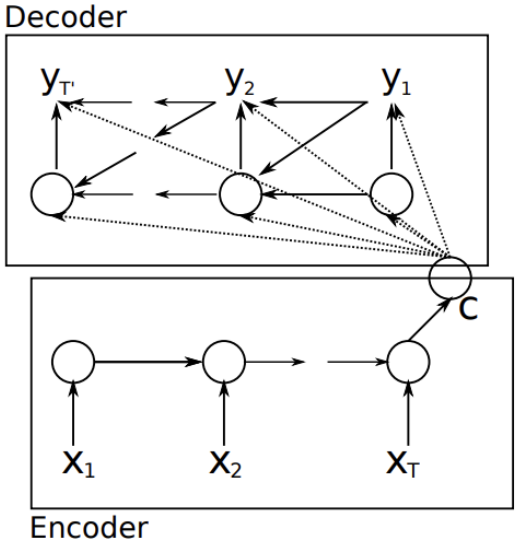

Figure 1 – RNN Encoder-Decoder Architecture for Machine Translation

The RNN Encoder learns to map the source sentence (X_1, X_2, ... , X_T) of variable length into a fixed context vector **C** capturing the information of the entire sentence. This vector is generated at the end of the source sentence (when EOS symbol is given as an input) and the output of the last the hidden state then given as input to the deocoder. The RNN Decoder then uses this context vector to predict the target sentence (Y_1, Y_2, ... , Y'_T). These two components are jointly trained to maximize the conditional probability of variable length target sequence (from target language) given a variable length source sequence (from source language). Each hidden unit has a novel gating mechanism (Cho et al., 2014a) which adaptively propagates relevant information from preceding words and discards unnecessary context information at each time step, thus capturing short and long-term dependencies within the sentence.

4.1.2 Model Training and Decoding

The RNN Encoder-Decoder model is trained on a phrase pair table containing the source and target language sentences as the phrase pairs. A SMT system can be implemented as a log-linear model trained using linguistic features and their corresponding weights. The training process involves updating these feature weights. The scores generated by the RNN based model are then used as additional features for training the SMT system. These scores are the conditional probabilities of all the target and source sentence pairs in the training data. The log-linear approach involves prediction of a target sentence using the mapping between the source and target sentences involving these features. While using the phrase pairs for training the RNN model, frequencies of each pair have been ignored. With this update, the model treats frequent and rare sentences equally and focuses more on the linguistics aspects of translation rather than the statistical aspects.

4.2 Sequence to Sequence Learning with Neural Networks

4.2.1 Approach and Architecture

Sutskever et al. (2014) uses Long-Short Term Memory (LSTM) architecture (Hochreiter and Schmidhuber, 1997) to achieve machine translation using sequence-to-sequence learning. With the LSTM architecture, the proposed model can be used for sequence learning tasks where the lengths of the input and output sequences are not known a-priori and are different from each other. To map a translation pair with different length and alignment, the model is divided into two parts - Encoder and Decoder. The architecture of the encoder is similar to that of (Cho et al., 2014b). The LSTM encoder is used to convert the variable length sequence from the source language into a fixed dimensional context vector. The words from the input

sentence are sequentially fed to the LSTM encoder. At time step t, word input x_t and previous hidden state output h_{t-1} are given as input to current hidden unit to generate the output h_t using learned weight matrices. The output of the last hidden unit for the EOS (end of sentence) token generates a fixed-dimensional context vector containing the information of the source sentence. This is followed by a LSTM decoder which at time step t, takes as input the context vector and vectors pertaining to words predicted before the current time step to predict the corresponding word from the target language at that time step. The decoder prediction runs in loop at the end of which it predicts the target sentence. The proposed model outperforms the phrased-based SMT model proposed by (Cho et al., 2014b). Use of LSTM takes care of the long-term dependencies that need to be captured in the translation process. The source sequence is reversed and given as an input to the LSTM encoder. So for a translation a, b, c to x, y, z, the input sequence is reversed as c, b, a and given as input to map it to the target sequence x, y, z. This ensures that some words at the start of the sequence have close proximity to their corresponding words in the target language without having a significant impact on the average distancing between source and target words. The training phase involves maximizing the conditional probability of the correct target sequence given the source sequence. The use of LSTM ensures that translation takes into consideration word order and maps the source and target sentences accordingly.

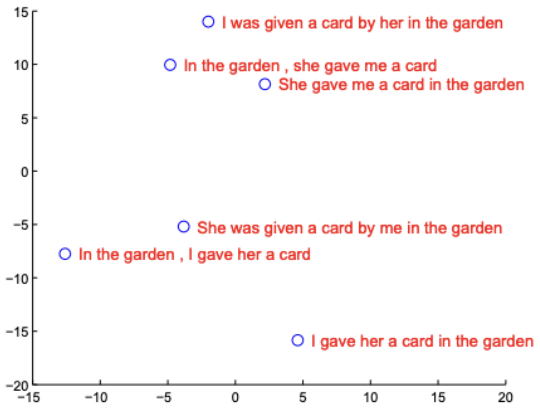

Figure 2 – 2-dimensional PCA projections of the LSTM hidden states for a group of phrases in their active and passive form.

The sample phrases plotted in Figure 2 show that the phrases with similar meaning are clustered together irrespective of their active passive form. Thus, it is evident that the hidden states within the LSTM architecture capture the semantic information of the phrases involved in the translation and are insensitive to the active and passive from of sentences.

4.2.2 Model Training and Decoding

The LSTM Encoder-Decoder model proposed by (Sutskever et al., 2014) works as an end-to-end MT model. For the purpose of this research the model has been trained on a parallel corpora from the WMT' 14 dataset, with English and French as the source and target languages respectively. A 4 layer deep LSTM is used for training the sentence pair using stochastic gradient descent optimization algorithm and a variable learning rate. Training process for the model involves updating weights to maximize the conditional probability of the target sentence given the source sentence. The objective function of training has been expressed through the following equation :

$$1/|N| \sum_{(T,S)\epsilon N} logp(T|S) \qquad (1)$$

where N is the training dataset, S and T are the source and target sentences from the training dataset. After training, a left-to-right beam search decoding algorithm (bea) is used to predict the correct translation for a given source sentence. A word from the vocabulary is predicted at each time step of the decoder and decoding process continues till the model predicts the end of sequence (EOS) symbol.

4.3 NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

4.3.1 Approach and Architecture

The machine translation approach used by (Sutskever et al., 2014) consist of an encoder mapping the source sentence into a fixed dimensional context vector. The output of the last hidden state in the encoder is responsible for generation of this vector which carries the information of the entire input sentence. The issue with this approach is that for sentences with large length it becomes difficult for the encoder to capture all the information from the source sentence. Thus, it may adversely affect the translation performance with increase in the length of the source sentence. The model proposed

by (Bahdanau et al., 2015) introduces an additive attention mechanism, to generate a variable-length context vector for each target word under translation. The proposed model has two parts - alignment and translation. The translation model follows an architecture similar to (Cho et al., 2014b) with separate encoder and decoder units. For generating a target word at time step t the alignment model searches for words from the source sentence with most relevance to the target word. Based on these words a context vector is generated. The target word is predicted using this context vector, decoder hidden state output and previously predicted target words. Thus, the conditional probability at time step t for the decoder output y_t is calculated as :

$$p(y_t|y_1, ..., y_{t-1}, x) = g(y_{t-1}, s_t, c_t) \quad (2)$$

where g is a non-linear function, s_t is the decoder hidden state output, y_{t-1} is the previously predcited target word and c_t is the distinct context vector at that time step.

The Encoder in the model is a Bidirectional neural network comprising of a forward and backward RNN. At time step t, hidden state of the forward RNN capturing information of the source sentence from the left end is concatenated with the hidden state of the backward RNN capturing information from the right end resulting in an encoder hidden state h_t.

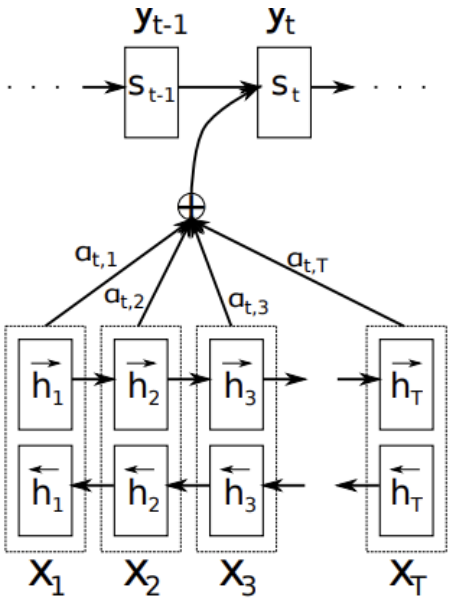

Figure 3 – Alignment model for generating a target word using a dynamic context vector from the source sentence

As shown Figure 3, the decoder hidden state output s_t is evaluated based on a dynamic context vector from the encoder and previous hidden state of the decoder. This context vector is calculated using the weighted sum of hidden states from the encoder.

$$c_i = \sum_{j=1}^{Tx} \alpha_{ij} h_j \quad (3)$$

where, c_i is the dynamic context vector, α_{ij} is the weight associated between the j*th* encoder hidden unit and i*th* decoder hidden unit and h_j is the j*th* encoder hidden unit.

The weight α_{ij}, treated as an alignment score between each combination of the encoder and decoder hidden states, enables information from selected parts of the source sentence to be used for predicting a target word. The resulting alignment model is jointly trained as a feed forward neural network along with the translation model.

4.3.2 Model Training and Decoding

Bahdanau et al. (2015) has compared the performance of the proposed model called **RNNsearch** with a baseline RNN encoder-decoder model, wherein the proposed model outperforms the conventional encoder-decoder model due to the presence of alignment model embedded in its decoder. Each of these models have been trained twice, first with sentences up to length of 30 words and than with sentences up to length of 50 words. All these models have been trained to learn parameters for generating a conditional probability for predicting target sentence given a source sentence. The research uses minibatch stochastic gradient descent algorithm along with Adadelta optimizer for training each model. A learned model with these updated parameters is then used to predict the most probable translation for a given source sentence. For decoding the translation, the proposed model has used beam search algorithm similar to (Sutskever et al., 2014).

4.4 Attention Is All You Need

4.4.1 Approach and Architecture

The model proposed by (Vaswani et al., 2017) is the first one to introduce a Transformer-based architecture for solving NLP problems such as machine translation. The architecture of this model deviates from the traditional recurrence within the neural network and solely uses attention mechanism for evaluating representation of a sequence by relating different tokens within it. The Transformer model

follows the encoder-decoder architecture similar to (Cho et al., 2014b), (Sutskever et al., 2014) and (Bahdanau et al., 2015).

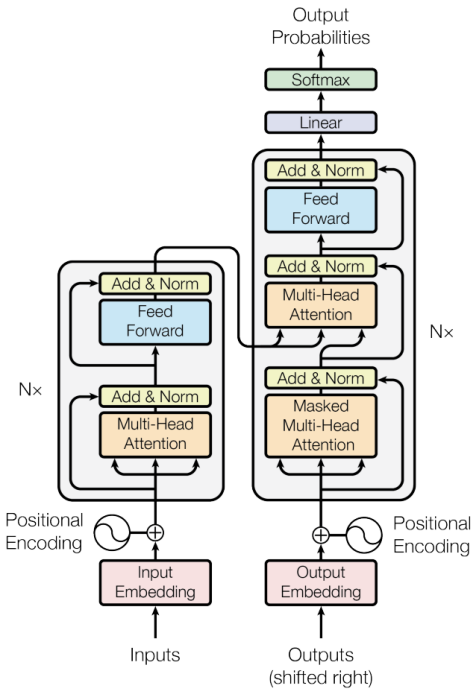

Figure 4 – Architecture of a Transformer based MT model

As shown Figure 4, the encoder consist of a set of stacked layers with each layer comprising of two sub-layers, a self-attention layer followed by a fully-connected feed forward layer. The decoder consist of a masked self-attention layer, a self-attention layer for processing the encoder output and finally a fully-connected feed forward layer. Vaswani et al. (2017) has designed the model with 6 such identical layers for encoder as well as decoder. The two types of self-attention layers proposed are scaled dot-product and multi-head attention. The source sentence is given as an input to the embedding layer where each word is converted into a vector of fixed dimension. An additional vector encoded with the information of position of each word in the sentence is concatenated to this embedding vector. The self-attention mechanism starts with generation of three types vectors from the embedding vector namely - query, key and value. A self-attention score for each word against all other words is calculated to identify focus given to different parts of the sentences while encoding the word at the current position. This helps the model to understand if two words are related to each other. For example, in the sentence *The animal didn't cross the street because it was too tired*, the word *it* refers to the word *animal*. This is very easy for humans to understand but not for machines. However, with the use of self-attention when the word *it* is being encoded, focus is on the word *animal* resulting in a strong association between these two words. To generate the self-attention scores for each word, a dot product of the query vector of that word with key vectors pertaining to all words is taken. These scores are further divided with the square root of the dimension of the key vector and passed through a softmax layer. Thus, the scores for each word relay the information of the importance of surrounding words with values between 0 and 1. These scores are multiplied to the value vector pertaining to each word to generate the output vector from the self-attention sub layer. At a matrix level taking all the words from the source sentence into consideration the equation for self-attention is epxressed as -

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (4)$$

where **Q**, **K** and **V** are matrices containing the vectors used for associating relation between the words. The model further improves its architecture by developing multiple attention heads, with each attention head having its own set of query, key and value weight matrices. This enhances the model's ability to focus on different relevant positions within the sequence. Each attention head generates a separate matrix for the input sequence. These matrices are concatenated together and multiplied with a learned weight matrix to generated a cumulative output matrix. The output matrix consist of a vector representation of all the words in the source sequence. Each vector now encodes a more detailed information regarding the focus on different parts of the sequence for the current word. The self-attention layer is followed by a full-connected feed forward layer. This layer consist of two linear transformations applied separately to each word vector generated from the self-attention layers. The output of this layer is a new set of vectors which is further given as an input to the next encoder layer. Output from each of the sub-layers in the encoder is processed by adding it to its respective input and applying normalization. In case of machine translation the source sentence is given as an input to the

encoder and all the words are processed simultaneously through the self-attention and feed forward layer. The output of the last encoder layer contains a set of attention vectors which are given as an input to the decoder. The decoder predicts the target sentence one word at a time. With attention vectors from encoder and words from the target sentence predicted till that time step, a vector representing the current word is predicted. This vector is passed through a linear layer which converts it into a logits vector with dimensions equal to the vocabulary used for machine translation. This vector is further passed through a softmax layer to generate a probability score between 0 and 1 for all the words in the vocabulary. The word with the highest probability is the word predicted by the model. This process continues in a loop till the target sentence reaches the end of sequence symbol.

4.4.2 Model Training and Decoding

The training data consist of parallel corpora containing sentence pairs from the source and target language. A group of sentences pairs are batched together for training. The source sentence is passed through the Transformer architecture to predict the target sentence. This forward pass is followed by a back propagation where the predicted sentence is compared with the actual target sentence. A loss function is used to quantify this difference and an optimizer is used to update the parameter weights. This process is continued in batches till the model learns the translations in the training corpus. A beam search decoding algorithm is further used to predict the target sentence for a given source sentence. The training process in the proposed model is comparatively faster than the RNN based translation models. The model has a achieved a BLEU score of 41.8 and has served as the state-of-the-art model when it was first introduced in 2017.

5 Performance

The performance of all the proposed models has been evaluated based on a metric known as the BLEU (bilingual evaluation understudy) (Papineni et al., 2002) score. Each research paper has proposed translation models with variations in their architecture. However, the best performance among all these variations has been listed in the table. The **CSLM + RNN + WP** is the RNN model proposed by Cho et al. (2014b). Cho et al. (2014b) computes the BLEU score for a phrase-based SMT where an RNN encoder-decoder based neural network is only used as a feature to score translations for a given parallel corpora. The baseline system considered here is a statistical machine translation model. It has been observed that the translation scoring feature added by the proposed RNN model improves the translation performance for baseline system. The RNN model has been trained on sentence pairs without providing any information pertaining to the frequency of their occurence. Hence, it captures linguistic information within the translation and performs equally well for rare and frequent translation pairs. All the other models discussed in the review can be considered as end-to-end neural machine translation models which can be used to generate a potential sentence from the target language when given a sentence from the source language. Sutskever et al. (2014) has proposed the model named **LSTM (Seq2Seq Model)** and similarly **RNNsearch-50** has been proposed by (Bahdanau et al., 2015). The performance of these models is comparable and does not have a significant difference in their BLEU score as they all follow a recurrent neural network architecture where the inputs to the encoder are fed sequentially. The **LSTM (Seq2Seq Model)** model proposed by (Sutskever et al., 2014) performs better when translating long sentences. The LSTM architecture used for constructing the encoder and decoder hidden units has the capacity to capture long-range dependencies. Encoder within the model receives input from the source sentence in reverse order. With this input it has been empirically observed that decreasing the distance of the words at that start of the sequence helps in capturing translation dependencies without affecting the average distance between the corresponding words of the source and target sentence. The model proposed by (Bahdanau et al., 2015) makes an improvement to this approach by selectively generating a dynamic context vector for a given target word based on the source words stron-

Model	BLEU Score	
	Baseline	**Proposed**
CSLM + RNN + WP	33.30	34.54
LSTM (Seq2Seq Model)	33.30	34.8
RNNsearch-50	26.71	34.16
Transformer (big)	38.1	41.8

Table 1 – Model Performance Evaluation in terms of BLEU score.

gly associated with it. The baseline used for comparison is a traditional RNN encoder-decoder model where the encoder maps the input sentence into a fixed dimensional context vector. Such a model has to encode the entire sentence into a single vector and the same vector is used by the decoder to predict all the target words. Such an approach makes it difficult to capture long-range dependencies and the model proposed by (Bahdanau et al., 2015) captures this information using the alignment model. Thus, the alignment model ensures that each target word predicted in the decoder is provided with a context vector containing information from the source words relevant for its translation. Generally a translation process is based on one to one correspondence between words from the source and target language. However, sometimes the word order changes based on the syntactic structure of the language. For example, the phrase **European Economic Area** in English when translated to French becomes **zone économique européenne**. In this case, the noun and adjective word order have got interchanged in the translation process. However, the alignment mechanism embedded in the proposed model is able to map the corresponding words of the both the languages irrespective of change in the word order. However, the **Transformer (big)** model from (Vaswani et al., 2017) eschews from this approach by simultaneously processing all the source sentence words using their corresponding positional embeddings and self-attention mechanism to understand the relation between different parts of the source sentence. From the relatively high value of BLEU score it is evident that a self-attention mechanism captures the semantic and syntactic information within a given translation to greater extent and achieves the translation performance within a shorter time frame. The positional embeddings concatenated to word embeddings of all the words in the sentence capture the relation between words using an operation executed in constant time. However, the recurrent architecture followed by (Cho et al., 2014b), (Sutskever et al., 2014) and (Bahdanau et al., 2015) sequentially builds on the relation between different parts of the sentence. Thus, Transformer-based models enable faster computation for achieving translation at a reduced training cost.

6 Conclusion

All the research papers involved in the review work towards solving a common problem on Neural Machine Translation. All the papers make use of the WMT'14 dataset for English to French translation, however the corpora selected for training of models across all the papers is slightly different. The work by (Cho et al., 2014b) evaluates performance of their model for scoring the phrase translations in a phrase table and remaining three papers focus on a developing an end-to-end Neural Machine Translation model for generation of sentences in the target language given the source language. The performance metric used for model evaluation is the BLEU score. Cho et al. (2014b) and (Sutskever et al., 2014) construct an RNN encoder decoder architecture using gated recurrent units and LSTMs respectively. Bahdanau et al. (2015) makes an improvement in the architecture of (Cho et al., 2014b) by introducing additive attention in the model. Vaswani et al. (2017), develops a Transformer-based model, by deviating from a recurrent neural network to an architecture with self-attention mechanism. The papers have proposed models which have tried to outperform the state-of-the art models at the time of their introduction.

References

Speech understanding systems. Summary of results of the five-year research effort at Carnegie-Mellon University.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

LE Dostert. 1957. Brief history of machine translation research. In *Research in Machine Translation*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9 :1735–80.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.