Neural Machine Translation with Attention

Technical Report · August 2021

DOI: 10.13140/RG.2.2.29381.37607/1

2 authors:

Mohammad Wasil Saleem
Universität Potsdam
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Sandeep Uprety
Universität Potsdam
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Neural Machine Translation with Attention

Mohammad Wasil Saleem
Matrikel-Nr.: 805779
Universität Potsdam
saleem1@uni-potsdam.de

Sandeep Uprety
Matrikel-Nr. 804982
Universität Potsdam
uprety@uni-potsdam.de

Abstract

In recent years, the success achieved through neural machine translation has made it mainstream in machine translation systems. In this work, encoder-decoder with attention system based on "Neural Machine Translation by Jointly Learning to Align and Translate" by Bahdanau et al. (2014) has been used to accomplish the Machine Translation between English and Spanish Language which has not seen much research work done as compared to other languages such as German and French. We aim to demonstrate the results similar to the breakthrough paper on which our work is based on. We achieved a BLEU score of 25.37, which was close enough to what Bahdanau et al. (2014) achieved in their work.

1 Introduction

Machine Translation (MT) is the task of translating text without human assistance while preserving the meaning of input text. The early approach to machine translation relied heavily on handcrafted translation rules and linguistic knowledge. Started in early around 1950s, unlike rule-based machine translation, Statistical machine translation (SMT) generated translations based on statistical models whose parameters are derived from the analysis of bilingual text corpora (Koehn et al., 2003). Though reliable, for SMT, it can be hard to find content for obscure languages and is less suitable for language pairs with big differences in word order making the quality of translation far from satisfactory. With the progress in deep learning being applied to MT, in 2014, end-to-end neural network translation model was proposed by (Bahdanau et al., 2014; Sutskever et al., 2014) where the term "neural machine translation" was formally used. Neural machine translation (NMT) is the newest method of MT that uses a single large neural network to model the entire translation process, freeing the need for excessive feature engineering. Through the rapid research and breakthroughs, end-to-end neural machine translation has gained remarkable performances (Shi et al., 2021; Bahdanau et al., 2014) and have become mainstream approach to MT.

2 Related Work

Early problem of NMT was often the poor translation for long sentences (Sutskever et al., 2014) which can be attributed to the fixed-length of source encoding in conventional encoder-decoder as suggested by Cho et al. (2014a) for which the concept of attention to NMT was introduced by Bahdanau et al. (2014) to avoid keeping a fixed source side representation.

As compared to separately tuned components in SMT, newly emerging Neural Machine translation radically departures from previous machine learning approaches as the training of NMT is end-to-end which has significantly improved translation quality across 30 different languages (Junczys-Dowmunt et al., 2016). NMT model can be attractive for various reason one being scalability issue, whether it be memory requirements or computational speed. Another being able to train all the character embedding as each characters frequently occurs in the training corpus.

Most neural machine translation models proposed use encoder-decoder where a neural network reads and encodes a source sentence into a fixed-length vector and a decoder then outputs a translation from the encoded vector, where in most of the cases the encoder and decoder are mainly implemented as RNNs, CNNs or self-attention network (Wu et al., 2018). The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is

jointly trained to maximize the probability of a correct translation given a source sentence (Bahdanau et al., 2014). After the initial proposal by (Sutskever et al., 2014; Bahdanau et al., 2014), much work has been done on the sequence-to-sequence neural machine translation model ranging from new attention mechanism (Luong et al., 2015) to working on the problem of out-of-vocabulary words (Jean et al., 2015) for which sequential RNNs are used both for encoding source sentences and generating target translation.

We draw our inspiration for machine translation with attention from Bahdanau et al. (2014). We have chosen to base our project on this paper as attention mechanism has been widely used as baseline and is thoroughly studied among the NLP community.

3 Model

Machine Translation is equivalent to maximizing a conditional probability of a target sentence given the source sentence. In Neural Machine Translation, we parameterize it to maximise the conditional probability. The approach used by Cho et al. (2014b) was to encode the source sentence into a fixed-length vector, which becomes difficult to compress all the necessary information into a fixed length vector, which makes it difficult for the Neural Network to handle long sentences, and thus the performance of the encoder-decoder drops as the length of the sentences increases. So we use the model proposed by Bahdanau et al. (2014), where it does not encode the input sentence into a fixed-length vector, rather than it simply encodes the input sentence into sequence of vectors, and while decoding the translation, it select subset of vectors from the using attention mechanism. And Bahdanau et al. (2014) showed that encoder decoder model with attention mechanism cope better with long sentences. In the next section, we will first give a brief introduction on encoder-decoder, the RNN, and one of its type, GRU, the one we used in our model and finally the attention mechanism.

3.1 Encoder-Decoder

Encoder-Decoder, first proposed by Cho et al. (2014b), basically consists of 2 parts, the encoder and the decoder. Encoder codes the sequence of input sentence into dense vector representation, and then decoder takes in the encoded sentence and decode the representation into another sequences of words. They are trained to maximize the conditional probability of the output sentence, given the input sentence.

RNN is necessary when we need to maintain the word order in a sentence. This is not handled by bags of words model or other statistical models. In addition to input, x_i and output $\hat{y}_i$, we also have a state vector, a_i, which is initialized with vectors of zeros. i would be the i^{th} timestep In the first layer of RNN, the input and state vector is fed into the recurrent unit, the recurent unit may look like [1]:

$$a_t = f(W_{aa}a_{t-1} + W_{ax}x_i),$$

and,

$$\hat{y}_t = g(W_{ya}a_t)$$

where, t is the time step, W_{aa} is the weights between two hidden layer, W_{ax} is the weight between input and hidden layer, and W_{ya} is the weight between hidden and the output layer, and f can be $tanh$ or $Relu$ activation function, and g can be $sigmoid$ or $softmax$ activation function. After feeding the input to rnn unit, it returns a new state vector in the next time step. This new state vector will be mapped to the output vector using some function. This output vector can be used as a prediction. The new state vector is cached and is passed across the next unit of the RNN, along with the input in order for it to return the next state vector. This happens recursively for all the input elements. So, when the model is reading the second word, instead of just predicting output using only the second word, it also gets information from the previous time step (first word) in terms of the state vector.

One of the problem of RNN is that it runs into the problem of Vaishing gradient, first described by Hochreiter (1998). This happens when we have a very long sentence, which tends to have long term dependencies. That means a word at the end of the sentence would be semantically dependent on the word occurring at the beginning of the sentence. So, the gradients during the backpropagation step would have a very hard time propagating back to affect the words or weights of the earlier units. The gradients diminishes in the backpropagation step and not able to reach the earlier units. Generally, RNN has local influences where a word is mainly influenced by words closed to it. So that makes it difficult for the output at the later unit

[1] https://www.coursera.org/learn/nlp-sequence-models/home/welcome

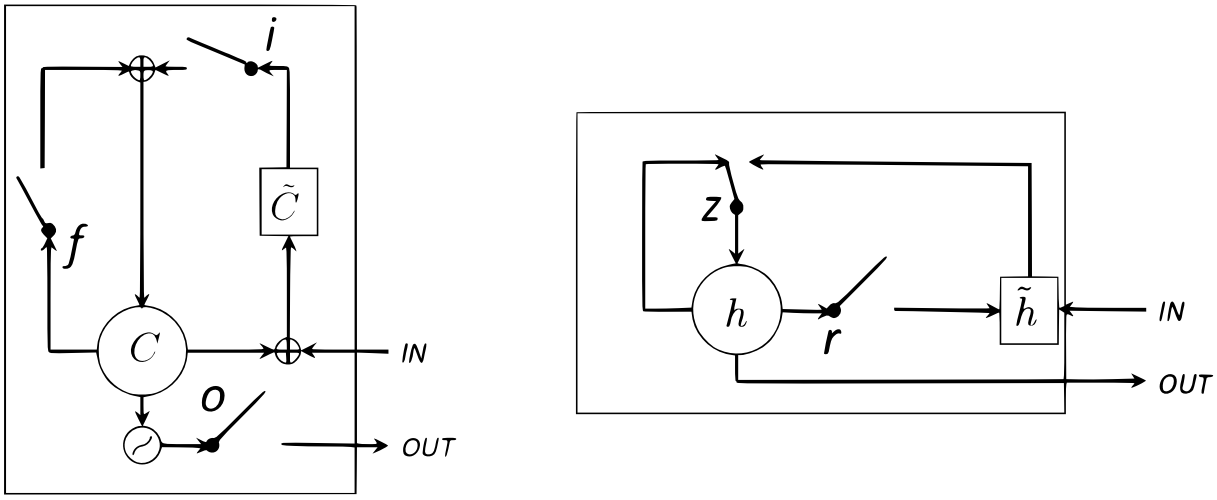

Figure 1: The left diagram represents Long Short Term Memory Unit, with i as an Input gate, o as an Output Gate, and f as a Forget Gate. The right diagram represents Gated Recurrent Unit, with r as Reset Gate, and z as an Update Gate. (Chung et al., 2014)

to be strongly influenced by a word that was very early in the sequence.

But we can further improve the training by using other RNN units, like GRU (Chung et al., 2014) or LSTM (Hochreiter and Schmidhuber, 1997), which are better at capturing long-range dependencies. The state vector in simple RNN can be considered as a memory, where the memory access was not controlled. At each step, the entire memory state was read and updated. But in GRU and LSTM, we use a gating mechanism to control the memory. Since, we used GRU in our model, so we will only describe GRU here.

In GRU, (Chung et al., 2014; Rana, 2016), the activation h_t^j is a linear interpolation between the previous activation h_{t-1}^j and the candidate ativation $\tilde{h}_t^j$:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j$$

where, z_t^j is the update gate, that decides how much GRU units updates its activation [See Fig. 1]. The update gate is given by :

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j.$$

And the candidate activation $\hat{h}_t^j$ is computed by :

$$\hat{h}_t^j = tanh(Wx_t + U(r_t \odot h_{t-1}))^j,$$

where $\odot$ denotes element-wise multiplication and r_t^j are reset gates. When a reset gate at specific time, t is set to 0, i.e. $r_t^j == 0$, which makes the GRU to forget the past, i.e. forget the previous state vectors. This is considered same as reading the first word of the input sentence. And finally, we can compute the reset gate by :

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j.$$

One of the weakness of RNN is that it only uses information that is earlier in the sequence to make predictions but not the information which are later in the sequence. When predicting the output at time step i, it does not use the word at time step i+1 or i+2 or any other words in the later time step. So, it would be useful to know not just the information from the words from the previous time step but also the information from the later time steps.

So, we use Bidirectional RNN, first proposed by (Schuster and Paliwal, 1997). From a point in time, it takes information from both the earlier and later time step in the sequence. The First RNN, which we called forward RNN, $\overrightarrow{f}$ is fed the input sequence as it is. And the second RNN, which is also called backward RNN, $\overleftarrow{f}$ is fed the input sequence in reverse order. This gives two separate state vectors – a forward state vector, $\overrightarrow{h}_j^T$, and a backward state vector $\overleftarrow{h}_j^T$. $\overrightarrow{h}_j^T$ would be a sequence of forward hidden state vectors, $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_{T_x})$, and similarly, backward state vector $\overleftarrow{h}_j^T$ would be a sequence of backward hidden state vector, $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_{T_x})$. And the output at a specific timestep is accounted by the concatenation of output of two RNN's, concatenating $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$, i.e. $h_j = [\overrightarrow{h}_j^T, \overleftarrow{h}_j^T]$. So, when predicting the output at a specific time step, it will

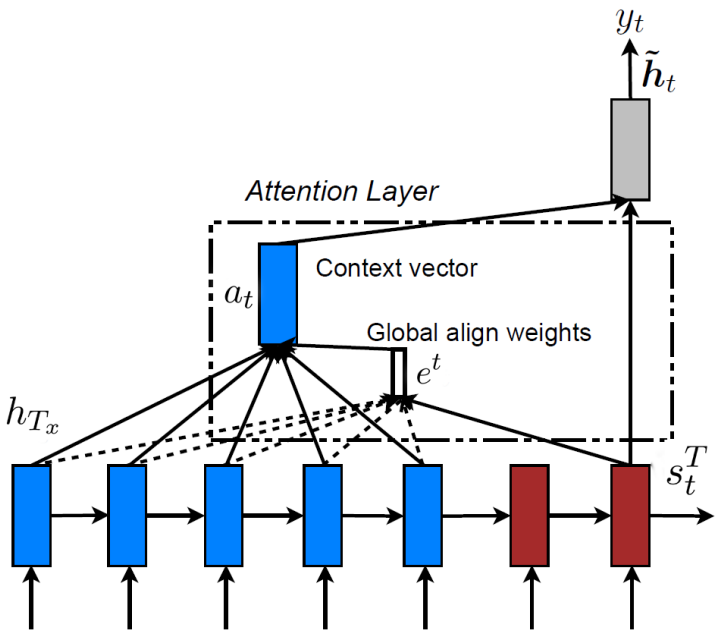

Figure 2: Attention Model (Luong et al., 2015)

use the information from the past, present as well as from the future. We need the entire sequence of data before we can make any predictions.

The architecture that we are proposing here is based on the Encoder-Decoder Framework. The encoder takes in the input sentence and converts them into a vector representation

The encoder can be an RNN (Cho et al., 2014b) or LSTM unit (Sutskever et al., 2014). They processes the input sentence, pass it through RNN or LSTM, and when it encounters the end of sentence, then the hidden state, that captured all the relevant information passes it to the decoder. Then this information is used to predict the translations in the decoder, which can be RNN or LSTM, until it predicts the end of the sentence token. The hidden state needs to remember every word from the input sentence. So that is why this model tends to work for short sentences and not long sentences. Even though if we used LSTM or GRU, which tends to remember the words that occured very early in the sequence, it will still not be able to learn the alignment between the source word and the target word. They often forget the initial part of the sentence once they are processed in the encoder. That is why we use an alignment mechanism called attention. They help to memorize this information for longer sentences. But we need to learn this alignment. It can vary from language to language.

3.2 Attention

In this section, we will specifically define the alignment mechanism, [See Figure 2] that we used in our model. In RNN Encoder-Decoder model (Sutskever et al., 2014), we faced with the bottleneck problem, where the complete sequence of information of the source sentence, must be captured by one single vector, i.e. the last hidden unit of the encoder RNN is used as a context vector for the decoder, which becomes difficult for the decoder to summarise large input sequence at once. This also poses a problem where the encoder is not able to memorize the words coming at the beginning of the sentences, which leads to poor translation of the source sentence. The Attention mechanism just addresses this issue, by retaining and utilising all the hidden state of the input sentence during the decoding phase.

During the decoding phase, the model creates an alignment between each time step of the decoder output and all of the encoder hidden state. We need to learn this alignment. Each output of the decoder can selectively pick out specific elements from the sequence to produce the output. So, this allows the model to focus and pay more ”Attention” to the relevant part of the input sequence.

The first attention model was proposed by Bahdanau et al. (2014), there are several other types of attention proposed, such as the one by Luong et al.

Spanish	en la estrategia 2020 , reconocimos el hecho de que , si queremos mantener nuestro nivel de prosperidad en europa , tenemos que aumentar nuestra productividad .
English	in the 2020 strategy , we acknowledged the fact that , if we are to maintain our level of prosperity in europe , we need to increase our productivity .
Spanish	sin embargo , es algo que debemos hacer si queremos demostrar a los estados unidos que nos deben considerar como un socio serio en la alianza contra el terrorismo .
English	yet do it we must , if we are to demonstrate to the usa that we are to be taken seriously as a partner in an alliance against terrorism .
Spanish	sabrán ustedes que fue también a instancias de esta cámara que la comisión entabló negociaciones , y estas han dado un resultado encomiable .
English	you will be aware that it was not least at the insistence of this house that the commission entered into negotiations , and these have produced a creditable result .

Table 1: Examples of Spanish and English sentences from the dataset

(2015).

We will only discuss the attention model, proposed by Bahdanau et al. (2014). After the input sequence is passed through the encoder, it produces hidden state for each of the elements in the sequence $(h_1, \ldots, h_{T_x})$. Then we multiply the decoders hidden state at time step t $(s_1, \ldots, s_{T_y})$, with all of the encoders hidden state, which gives us the alignment score of each of the encoder output with respect to the decoder input and hidden state at that time step:

$$e^t = [s_t^T h_1, \ldots, s_t^T h_{T_x}]$$

The alignment score quantifies the amount of Attention the decoder will place on each of the encoder outputs when producing the next output, so instead of looking at the entire sequence, it just concentrate on few relevant parts of the sequence when predicting the next word.

After calculating the alignment score, we pass the vector e^t through the softmax layer, to calculate the probability distribution.

$$\alpha^t = softmax(e^t)$$

Then we multiply each of the attention weights with each of the encoder hidden state, to get context vector, a_t

$$a_t = \sum_{i=1}^{T_x} \alpha_i^t h_i$$

If the attention score of specific element of the input sequence is close to 1, then its influence on the decoder output at that specific time step increases. And then finally, the context vector a_t produced will be concatenated with the decoder hidden state, s_t, i.e.

$$\tilde{h}_t = [a_t, s_t]$$

and is fed into decoder RNN, which produces new hidden state.

4 Data

We used a Parallel Corpus for English-Spanish language, which was extracted from the proceedings of the European Parliament, also called Europarl dataset (Koehn, 2005). It contains 1.96 Million sentences, each for English and Spanish. Most common words and count for both languages are shown below:

Words	Count
de	1799827
,	1456229
la	1222089
que	992176
.	867284
en	790382
el	696521
y	692640
a	577052
los	548495

Table 2: Spanish

Words	Count
the	1956558
,	1371506
of	932044
to	875415
.	864674
and	747108
in	622426
that	476250
a	430093
is	401782

Table 3: English

First we filter out all the sentences having words greater than 50. Then we sort these sentences based on the number of words in each of the sentences, so that we have less padded sentences in our initial indices and sentences with high padding to be at the end of our indices, following with

usual tokenization methods. Only preprocessing we used was to lower case the words.

We selected 1 million sentences for the modeling due to the hardware constraints. We split the dataset into usual format, i.e. train, test and validation. We used 900K sentences for our training, 80K for validation and remaining 20K sentences for the test set, which was not seen by the model during training. We did not limit the vocabulary size to any hard coded number, i.e. to get top N most frequent words. The vocabulary size we got from English sentences was 36838 and for Spanish was 63220. Only token we added was End of Sentence and Start of Sentence Tokens. We used Spanish as a source sentence and English as a target sentence. See Table 1.

5 Experiments

The encoder and decoder of our model have 256 hidden units each. The encoder consist of forward and backward gated recurrent unit (GRU) each having 256 hidden units. The decoder has a single forward gated recurrent unit (GRU), with 256 hidden dimensions, unlike 1000 hidden units, as in Bahdanau et al. (2014) due to the hardware limitations. And we only trained the model for Spanish to English translation.

We used Adam optimizer to train the model, and gradient update is computed with a batch size of 32 sentences.

We initialized our weights with xavier (Glorot) initializations, (Glorot and Bengio, 2010) with Uniform Distribution, $U[-a, a]$, where

$$a = \sqrt{\frac{6}{n_{in} + n_{out}}}$$

where n_{in} is the number of input neurons in the weight tensor, and n_{out} is the number of output neurons in the weight tensor.

The total number of trainable parameters were 43,564,519. We trained the model for roughly 20 hours. After our model was trained, we use greedy search to predict the translation for the given input sentence, that maximizes the conditional probability instead of using beam search as mentioned in the paper Bahdanau et al. (2014) due to our unfamiliarity and technical difficulty dealing with Beam search.

We, then used BLEU (Papineni et al., 2002) score to evaluate how the model was working on the test data.

6 Results

6.1 Quantitative Results

We trained our model on maximum length of 50 sentences, so any length smaller than or equal to 50 is used for the training, We trained our model until the error on the validation data or the development data stops decreasing, in order to avoid the problem of overfitting. We achieved a BLUE score of 25.76 on the test data and the error rate for Spanish to English translation was 4.267 on our test data. According to Bahdanau et al. (2014), we can say that our model out perform the encoder decoder model, proposed by Cho et al. (2014b), for 50 sentences, where they got BLEU score of 17.82. Their performance drops when the length of the sentence is increased (Cho et al., 2014b). So, the limitation of using fixed length vector in simple encoder decoder model in Cho et al. (2014b) work was the reason that it was under performing with long sentences.

This was our motivation to use the proposed approach by Bahdanau et al. (2014), where the performance of the encoder-decoder with attention shows no deterioration with sentence of length greater than 50 sentences. The result that we got which was 25.76 was quite close to the Bahdanau et al. (2014), where they got BLEU score of 26.75,training with 1000 encoder and decoder dimensions, and training on corpus of 384M words. They were also able to achieve BLEU score of 28.45 when trained the data until the performance of the validation stopped improving.

6.2 Qualitative Results

The model proposed by Bahdanau et al. (2014) provides a way to investigate the soft alignment between the translated sentence from the model and the input sentence. The matrix given in Fig 3, each of the cells represent the weights α_{ij} of the annotation of the j-th source word for the i-th target word. This helps in visualizing and see which word from the input sentence were considered more important for generating the target word.

We see that majority of the weights are concentrated on the diagonal matrix, along with non-monotonic alignments. The non-monotonic alignments would be high for long sentences, since the words in long target sentence tends to have dependence on more than one word in source sentence.

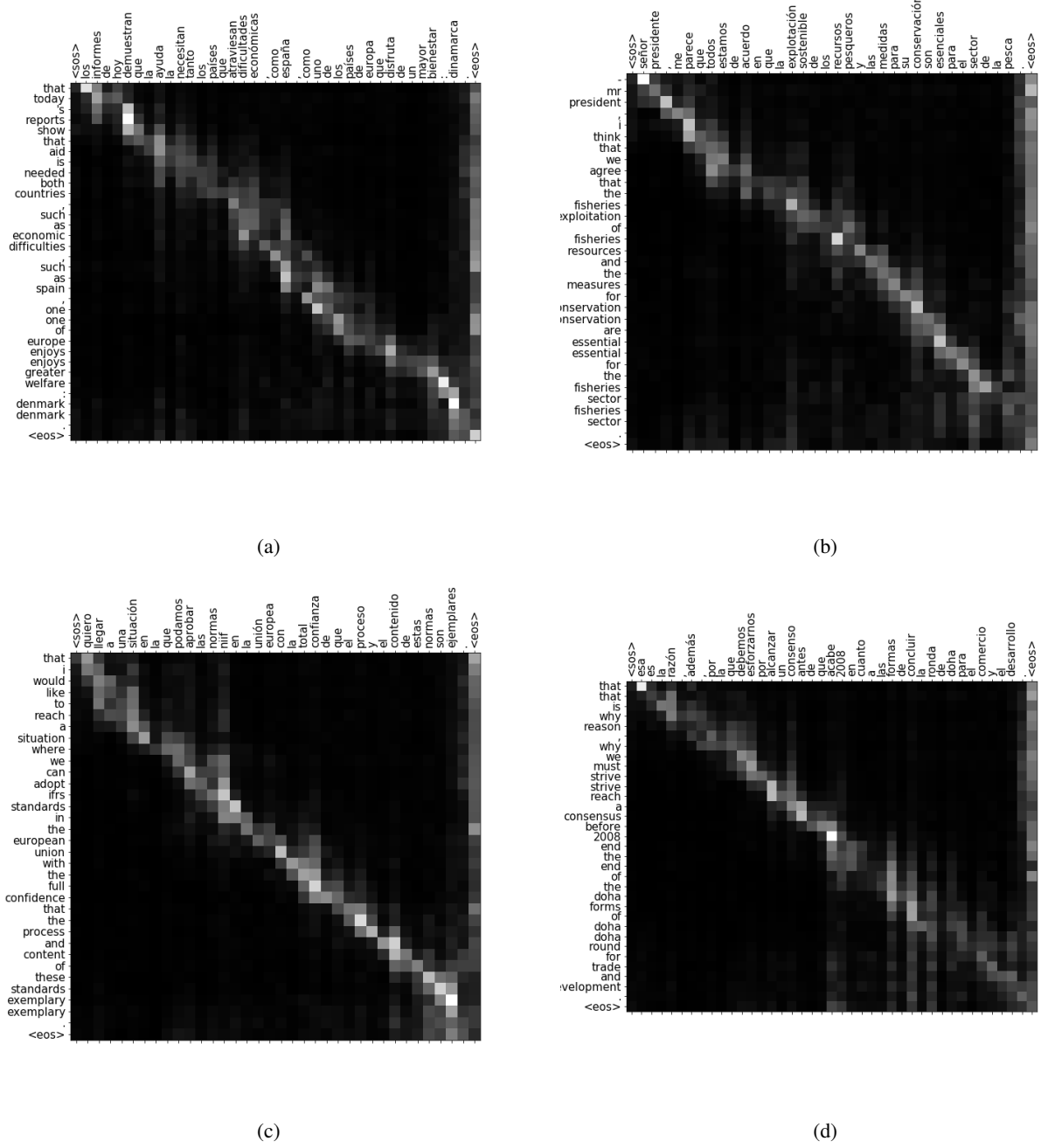

Figure 3: Alignments translated from Spanish to English by our model. The row represents the translated sentence, English and the column represents the source sentence, Spanish. Each of the cells of the matrix represents weights, α_{ij}, of the annotation of the j-th source word for the i-th translated word. (1:White, 0:Black)

Let us take an example from the test set, consider the source sentence:

> *son los estados miembros de la zona del euro los que no han cumplido , en especialla republica federal de alemania que se niega a mantener su promesa.*

And its translation by our model is :

> *that it is the member states of the euro area which have not complied honour, particularly the federal republic of germany that refuses to keep their promise to sustain their promise.*

And the reference is:

> *it is the member states of the euro area that have not delivered - and particularly the federal republic of germany, which is refusing to keep its promise.*

We can observe from our translated sentence, the model generates *"have not complied "*, instead of *"have not delivered"* (from reference sentence), which has the same meaning. It tries to preserve the meaning of the whole sentence, and it does not blindly takes the word from the reference, it tries to generalize.

Refer to Table 4 at the end of the paper for more translations from Spanish to English using Encoder Decoder model with Attention Mechanism.

7 Discussion

After trying to achieve the result similar to what was presented in the paper, we are satisfied with our result though there is much that can be improved. Limitation caused by the hardware held us back from achieving better results. We used the server provided by the university and as a backup used google colab for our work, so we had to be wary of the maintenance schedule happening in the server and the limitation of 24 hrs of worktime on google colab, which otherwise could interrupt while we were training our model. So, to overcome these challenges we decided to use 1M sentences from each form Spanish and English dataset, and reduce the parameters for encoder-decoder.

We also faced problem with length of the vocabulary size of English and Spanish sentences, where the vocabulary size of the English sentence were the output dimension, and vocabulary size of Spanish sentence were input dimensions of our model. This leads to increase in number of trainable parameters, the decoding complexity increases with number of target words (vocabulary size of the English), where this problem has been addressed by Jean et al. (2015).

8 Conclusion

The approach proposed by Cho et al. (2014b) was to encode the whole sentence into fixed length vector, and this becomes problematic when dealing with long sentences. We extend this basic encoder-decoder model by an Attention mechanism (Bahdanau et al., 2014), where the model searches for the input word computed from the encoder, which best align with the target word, when generating each target word. This frees the model from having to encode the source sentence into a fixed length vector, only rely on the information relevant for generating each target word. We compared our model for Spanish to English translation with both of these approaches, and found that our model works better than the encoder-decoder approach (Cho et al., 2014b), and have slightly lower results than the architecture with Attention mechanism, (Bahdanau et al., 2014). We also observed that the model tries to align the target word with the relevant word from the translated sentence.

In the future work, there are several things we can try. We can train our model on much larger dataset, with a better hardware and with different attention models. We can also focus on how to handle the stop words, punctuation, as we can see on Table 2 and Table 3, which accounts for the highest word count, and also to handle the unknown words, which does not appear in the training data, but in the test data.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Kyunghyun Cho, B. V. Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*, 2014a.

Kyunghyun Cho, B. V. Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares,

Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014b.

J. Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 6:107–116, 1998.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

Sébastien Jean, Kyunghyun Cho, R. Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *ArXiv*, abs/1412.2007, 2015.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. Is neural machine translation ready for deployment? a case study on 30 translation directions. 01 2016.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MTSUMMIT*, 2005.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 48–54, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL `https://doi.org/10.3115/1073445.1073462`.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

R. Rana. Gated recurrent unit (gru) for emotion classification from noisy speech. *ArXiv*, abs/1612.07778, 2016.

M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997.

Xuewen Shi, Heyan Huang, Ping Jian, and Yi-Kun Tang. Improving neural machine translation with sentence alignment learning. *Neurocomputing*, 420:15–26, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.05.104.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Shuangzhi Wu, Dongdong Zhang, Zhirui Zhang, Nan Yang, Mu Li, and M. Zhou. Dependency-to-dependency neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26:2132–2141, 2018.

Source	señor presidente , señor presidente en ejercicio del consejo , señor presidente de la comisión , señorías , me gustaría hacer tres breves comentarios .
Reference	mr president , mr president - in - office of the council , mr president of the commission , ladies and gentlemen , i would just like to make three brief comments .
Our Model	president mr president , mr president - in - office of the council , mr president of the commission , ladies and gentlemen , i would like to make three brief comments brief comments .
Google Translate	Mr. Chairman, Mr. Chairman-in-Office of the Council, Mr. Chairman of the Committee, ladies and gentlemen, I would like to make three brief comments.
Source	como los estados miembros , la comisión procura promover el estado de derecho , sin el cual los derechos humanos no obtendrán reconocimiento en ningún territorio .
Reference	the commission is involved , as are member states , in the promotion of the rule of law , without which human rights can not operate in any territory .
Our Model	that as the member states , the commission intends to promote the rule of law , without human rights will not not be any recognition in any territory in any territory .
Google Translate	Like the member states, the commission seeks to promote the rule of law, without which human rights will not be recognized in any territory.
Source	por escrito . - he votado a favor del informe de la señora fraga , que permite a groenlandia exportar productos pesqueros a la ue a pesar de no ser miembro .
Reference	in writing . - i voted in favour of ms fraga 's report , which allows greenland to export fishery products to the eu despite not being a member .
Our Model	in writing . - i voted in favour of mrs fraga estévez 's report , which allows greenland export export to export to the eu despite despite not being member .
Google Translate	written . - i voted in favor of the report by mrs fraga, which allows greenland to export fishery products to the eu despite not being a member.
Source	(pl) señor presidente , me gustaría una vez más manifestar mi satisfacción por la importancia que la comunidad confiere a la necesidad de innovación en europa .
Reference	(pl) mr president , i would like once again to express my pleasure at the importance that the community attaches to the need for innovation in europe .
Our Model	that (pl) mr president , i would once again like to express my satisfaction satisfaction that the community attaches to the community to the need for innovation in europe .
Google Translate	(pl) mr president, i would like once again to express my satisfaction with the importance that the community attaches to the need for innovation in europe.
Source	son los estados miembros de la zona del euro los que no han cumplido , en especial la república federal de alemania que se niega a mantener su promesa .
Reference	it is the member states of the euro area that have not delivered - and particularly the federal republic of germany , which is refusing to keep its promise .
Our Model	that it is the member states of the euro area which have not complied honour , particularly the federal republic of germany that refuses to keep their promise to sustain their promise .
Google Translate	it is the eurozone member states that have not delivered, especially the federal republic of germany which refuses to keep its promise.

Table 4: Source and Reference form the test data, with translated sentence from our model along with Google translation (as of 16 August 2021)