DD2424 Project: Neural machine translation

Andrzej Perzanowski *amper@kth.se*
Lee Badal *badal@kth.se*
Martins Kuznecovs *markuz@kth.se*
Kevin Olsson *kevinols@kth.se*

Abstract

In this project we study and evaluate sequence-to-sequence models with Bahdanau attention mechanism performing the task of bilingual translation. We test the performance of different model architectures and train models on different languages. The evaluation method for the model translation performance is the commonly used BLEU score, as well as by testing different types of inputs. We plot the attention weights for certain input sentences in order to visualise how the network uses the input when translating. We did not find evidence that bi-directional encoder structure always performs best - all architectures we tested performed similarly in our experiments. Our models received good performance scores on data similar to the one seen in the training set. But we found that in general the models struggle with proper nouns, punctuation and sometimes with longer sentences.

1 Introduction

1.1 Background and related work

Humans are the most chatty of creatures and our language is the bedrock of conscious logic and communication. We often view language barriers as obstacles when trying to communicate, they can break fluidity or alter the meaning of a conversation and open up for the possibility of misunderstanding. As individuals, companies or societies, eliminating the risk of misunderstanding each other is in our best interests. Translation from one natural language to another is difficult because it generally requires an in-depth understanding of the text [1]. Machine translation is an attempt to automate translation, neural machine translation (NMT) is an attempt to solve machine translation using artificial neural networks (ANN).
In this project we will conduct an in depth study of one of the more popular types of network architecture for NMT and evaluate its performance. We look into different potential improvements to these networks, as well as look into its potential downsides. Lastly we evaluate the strengths and weaknesses of the models using standard evaluation techniques and provide some insight into the inner workings of the models with a visualisation of the attention mechanism.

The type of network architecture we will focus on is a sequence-to-sequence network with an Bahdanau attention mechanism based on the work by Bahdanau, et al. (2014) and Luong, et al. (2015), who studied these networks in detail [2, 3]. These networks work on a basis of an encoder-decoder system which is jointly trained to maximise the probability of a correct translation given a source sentence. An attention mechanism, which is the key concept the papers discuss, improves the performance of such a network by allowing it to search for a set of positions in the source sentence itself, where the most relevant information is located. This helps the network see long distance relationships in the sentence, as well us reduce the risk of the network forgetting things when dealing with longer inputs.
Plotting the weights of the attention layer given an input sequence is a fantastic visualisation technique. This gives insight to the inner workings of the network, clearly depicting which parts of the sentence the network is looking at when generating predictions [2].
Improvements to performance to this architecture are discussed in a helpful tutorial on NMT with sequence to sequence models [4]. In our experiments we will test the claim that using bi-directional

encoders could potentially improve the performance of the models. The idea of using bi-directional encoder architectures has already been introduced in the original work on Bahdanau attention mechanism [2]. The reason for using a bi-directional encoder is because we would like the annotation of each word to summarise both preceding and following words. This is why bi-directionality would be useful, since its design allows for that, further augmenting the attention mechanism to give the network better idea of what a sentence means.
Comparison between different types of attention mechanisms has also been extensively discussed in the literature [3].
A different problem, which has been covered extensively in the literature, is the issue of evaluating NMT performance. Due to the complexity of human language it is not easy to automatise translation - often case one word can have several different contextual meanings, while in some cases we distinguish between them by understanding the following word or the word after. In fact, there are no generic rules to how language works, therefore conventional knowledge engineering methods have had a low success rate. But what is a low success rate, and how might you measure whether a translation is fault free or not? This is a simple question but there is no simple answer. In our case we chose to calculate a bilingual evaluation understudy (BLEU) score, which is a score that includes humans subjective translation and the main idea behind it is that a MT should be evaluated in comparison to a professional human translator. This a standard way of evaluating translation abilities of networks [5].

1.2 Overview of the project

The project focuses on sequence to sequence models with an attention mechanism, one of the more modern, powerful and widely used approaches to Neural Machine Translation. Our goals are to train different models using this type of network to perform bilingual translation, and observe how good performance we can get for our data set using different languages and different model architectures. We will analyse the results in order to gain understanding and first hand experience of how these networks work and perform.
We will also understand and get some practice with using the BLEU score, a standard NMT network evaluation technique. And to gain some visual insight into the inner workings of the network, we visualise the workings of the attention layer of the network (this process is explained in the Methods section), which will shed light on how the network focuses on different parts of the input sentence when translating.

2 Experiments

2.1 Language translation with sequence to sequence network

The main goal of this paper is to train a sequence to sequence network with an attention mechanism on bilingual sentence pairs so that the network learns how to translate input sentences between the languages. We will train the models on several different languages to see if its performance is affected by the choice of language.

2.1.1 Testing performance of models with different architectures

We fine tune our training by testing how the performance of the network improves when the amount of epochs is increased in training and pick the optimal amount. Another experiment involves substituting, in both the encoder and decoder (covered in the Methods section), the GRU layer with an LSTM layer, since LSTMs have a more complex structure, meaning they potentially could improve performance, at the cost of longer training times.
As mentioned in the introduction, it turns out that Bahdanau attention networks tend to work better when the encoder uses a bi-directional RNN structure. We will test this claim by training models using encoders using BiRNN architecture and compare their performance versus our uni-directional models. We also train one of our models on a large dataset and observe how this affects performance.

2.2 Evaluating and comparing the models

To evaluate translation performance we compute the BLEU score of the different models. To do this we use nltk's BLEU score implementation[1]. On top of that, we will test their translation performance

on chosen examples, since there are several issues that sequence-to-sequence models usually have a hard time handling, which we look into. Even with an attention mechanism, handling long sentences could be problematic for the model, therefore the models ability to do so will be tested. An interesting evaluation of the model would be to see how proper nouns are handled, which would demonstrate if the model understood how to translate them. Effects on performance of adding punctuation to the sentences, such as commas and questions marks, will be examined. Lastly, we will inspect which sentences work well and which ones don't, focusing on observing the extent of models ability to generate grammatically correct sentences, and noting any issues.

2.3 Model visualisation

In order to give an intuitive and visual explanations of what is going on inside the networks attention layer, we will plot the annotation weights [2], also known as attention weights, of the model given a certain input sequence and use them to understand how the model thinks. We will only do this for one of the Swedish to English models.

3 Methods

3.1 Data and preprocessing

The data, consisting of tab-delimited bilingual sentence pairs (English and paired with another language) was obtained from Manythings.org [2].

In order for this data to be used it must be pre-processed by removing special characters. We add a $< start >$ token to the beginning of each sentence and a $< end >$ token at the end. Then, each word is mapped to an integer. The sentences are also padded to reach max length, to be able to pass them to the model (since the model is fixed and must be able to handle varying size sentences).

Table 1: Amount of data used in training models

Model	Sentence pairs amount	Training and validation amount
Swedish	17692	15000
Spanish	123770	100000
Russian	380911	350000

Table 1 shows how the data was used to train our models. The training and validation sets are created using a 80:20 split. The remaining data was used for computing the BLEU score.

3.2 Model architecture performance

Tensorflow was used to implement the model, based on Tensorflow tutorial on attention models[3]. Once the hyperparameters are defined, the data set is input into the model. The model is a sequence-to-sequence model, with an attention mechanism. The structure consists of an encoder-decoder architecture, the blue and red part of Figure 1. An encoder processes the input sequence and turns it into a context vector, a sentence embedding, of a fixed length. This representation summarises the meaning of the whole source sequence. The input into the decoder is the output of the encoder and the decoder computes the transformed output. The decoder also uses its hidden state as input to perform its predictions, depicted as dotted lines in Figure 1. Both the encoder and decoder have similar structure, of simply an embedding layer, followed by a RNN, the RNN structure represented as blue and red cells in Figure 1. In the network we use Bahdanau attention mechanism [2] which allows the model to automatically search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. The attention layer can be seen in Figure 1.

[1] https://www.nltk.org/

[2] http://www.manythings.org/anki/

[3] https://www.tensorflow.org/tutorials/text/nmt_with_attention

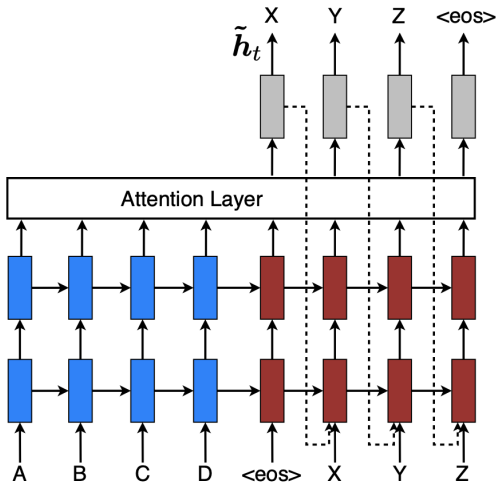

Figure 1: Picture of the model architecture, source [3]

Translation is achieved as seen in Figure 1, by feeding the model the input sequence (blue part) the model starts predicting the output (red part), and it stops predicting when the $< end >$ token is predicted. The bilingual models will be trained on Swedish-English, Spanish-English and Russian-English datasets. Our analysis will focus on the Swedish model, since that is the language we are most familiar with. Then we test the architecture changes, namely GRU vs LSTM encoder-decoder architecture on our Swedish and Spanish models. Lastly we train models with encoders using BiGRU encoder architecture and compare their performance versus our uni-directional models.

3.3 Evaluating and comparing the models

As mentioned earlier, the conventional way of evaluating NMT networks is by using the so called BLEU score. In BLEU scores we talk about n-grams, hypothesis and references. Hypothesis being the MT translation and references being human translation of the sentence. In n-grams, n is the number of words to include in the calculation. In the nltk's BLEU score implementation we use the default value 4-grams, that means we look at 4 words in sequence of the MT and if they correspond exactly to the 4 words the maximum score for that sentence will be returned. In other words, if the hypothesis is exactly the same as the reference a maximum score of 1 will be returned. When calculating the BLEU-score you can include multiple references as multiple translations can be a correct answer, the combined BLEU-score will be an average of the hypothesis in comparison to the references. In our paper we have just one hypothesis and reference for each calculation.

BLEU was developed to calculate document length translations, however when calculating scores for single sentences with words less than n there is a need for a smoothing function. In a paper Chen & Cherry [5] investigate different smoothing functions and conclude that method 4,5,7 had better sentence-level correlations with human judgement but all methods performed equally in performance tuning. That is why in our experiments we chose method4 for our final evaluation.

3.4 Model visualisation through annotation weight plots

In order to give an intuitive and visual explanations of what is going on inside the network, annotation weight plots are generated. Given an input sequence, by simply storing the resulting annotation weights of the attention layer during translation and then plotting the result, we can visualise what the layer is focusing on when looking at the input sequence. This provides us with an intuitive picture as to what the network pays attention to when processing the input.

4 Results

The final models were all trained on a GTI 980 TX for 15 epochs, since that was the best performing number of epochs in our experiments. The models reached different final loss values, but all were

below 0.1. SK-learns implementation of sparse categorical cross entropy was used for the loss function and an SGD optimizer.

Table 2: Models using Swedish/English data set, 1138 test pairs

Model	Smoothing Function	BLEU-score
BiGRU	method4	0.433
GRU	method4	0.419
LSTM	method4	0.424

Table 3: Results using Spanish/English data set, 8220 test pairs

Model	Smoothing Function	BLEU-score
BiGRU	method4	0.420
GRU	method4	0.435
LSTM	method4	0.434

Table 4: Models using Russian/English data set, 29967 test pairs

Model	Smoothing Function	BLEU-score
BiGRU	method4	0.302

4.1 Attention visualisation

In this visualisation we are focusing on just one of our Swedish-English models. Figure 2 and Figure 3 show some of the more interesting and varied attention weight plots we obtained from the GRU Swedish-English model, which highlight some strengths and weaknesses of that model architecture, as well as its thinking process. The x-axis and the y-axis of each plot correspond to the words in the source sentence and the generated translation, respectively. Each pixel shows the annotation weights, α_{ij}, where the subscripts represent the j-th source word for the i-th target word. These plots allow us to see which positions in the source sentence were considered more important when generating the target word.

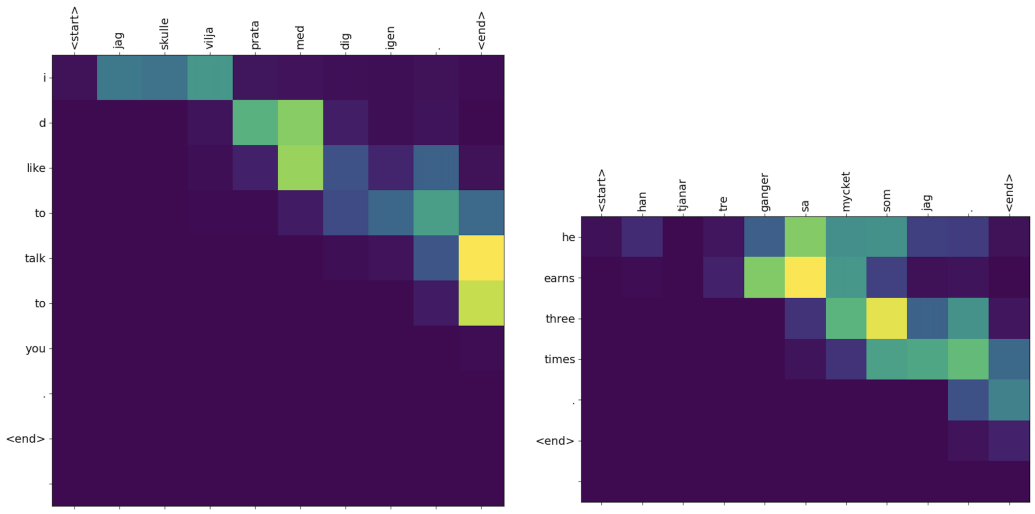

Figure 2: Plots showing annotation weights for certain sentences for the Swedish to English GRU model.

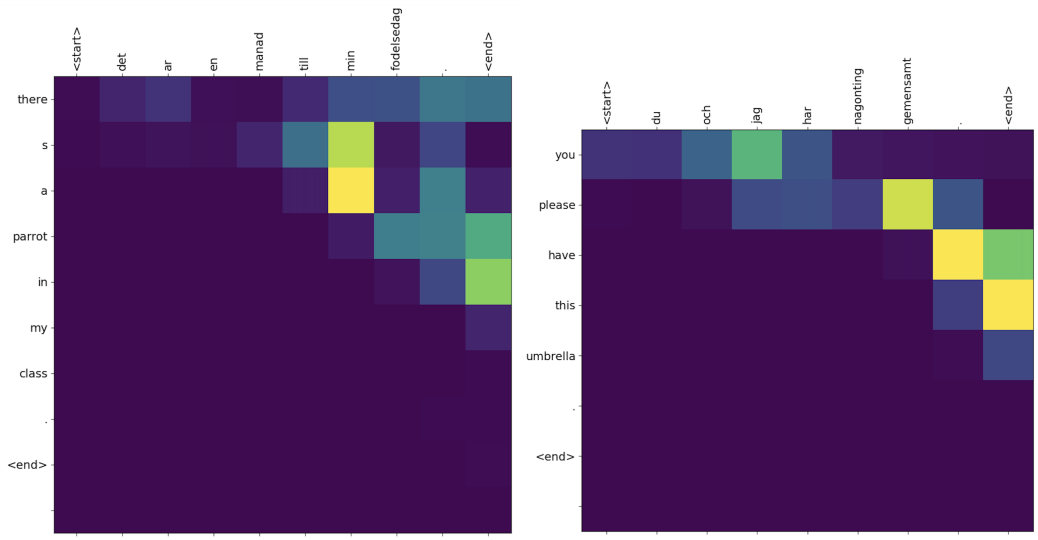

Figure 3: Plots showing annotation weights for certain sentences for the Swedish to English GRU model.

The correct translations of the sentences shown in Figure 2 and Figure 3 are
Figure 2 left plot correct translation: I would like to talk to you again.
Figure 2 right plot correct translation: He earns three times as much as I do.
Figure 3 left plot correct translation: It is one month left to my birthday.
Figure 3 right plot correct translation: You and me have something in common.

5 Analysis

5.1 Evaluating and comparing the models

By comparing the different models using BLEU score, namely BiGRU, GRU and LSTM we don't have a definite answer to which model architecture performs best according to our results. In the Spanish/English models, presented in Table 3, GRU was the best performing model, while in the Swedish/English model BiGRU was the best performing model according to our BLEU score. Perhaps bi-directionality is less effective in Spanish than it is in Swedish, or perhaps the dataset was not complex enough for the difference of architecture to matter. We would need to test multiple language model and of various sized data sets to draw any meaningful conclusions. Due to the size of the dataset, the Russian model was only trained on BiGRU, and was expected to benefit from it. As we can see in Table 4, we get much worse results for this model. However this shows that the network architecture can still perform reasonably well on large datasets.

On average our models tended to yield a BLEU-score of approximately 40%, which, according to the Google chart [4], means that our results are in the range of good to high quality translations. In comparison to SOA Google Translate NMT [6] our models are under-performing. Such an incredibly high score is quite surprising, however it maybe be misleading. Our models vocabulary contains only the words seen in the training data, so it is quite limited. In the test sets we only chose to test sentences with words that the model has seen before, since it cannot process them otherwise. As seen in Table 2, Table 3 and Table 4, a very large portion of test pairs from the test set were ignored, since they contained words unknown to the model, E.g. The original Swedish test set consisted of 2000 pairs, of which 1138 pairs were compatible with the model. The data we tested on is also part of the same dataset as our training data, meaning that it is of similar form to what the models have already seen during training. These effects greatly boost our models BLEU scores and need to be kept in mind. In the Russian model we see a drop in performance because the model was trained on a bigger dataset and thus knew more words, meaning that the test sentences were more diverse. A lot

[4] https://cloud.google.com/translate/automl/docs/evaluate

more data, a lot more diverse data, would be needed for our models if we wanted them to have better performance in general.

When inspecting individual sentences, we did observe BLEU-scores for single sentences as low as 0.1 in all of our models, which is not good. As mentioned earlier, it is always hard to evaluate machine translation - when testing the models using sentences we came up with our models would sometimes fail to translate trivial sentences but excel in sentences with more words. One could argue that the model is a failure in the first case but it is always a matter of subjective evaluation, the intended purpose and use cases of the MT. Further evaluation could be done by comparing the maximum and minimum of the BLEU-scores, as this addresses the issue of wide ranges of scores. The most relevant issue with using BLEU for our evaluation is the lack of translated references - for more accurate scoring more references would be needed to capture the scope of multiple possible wordings for the same sentence.
Even though we get BLEU scores as high as 40, this is not very representative of what our model would be able to do in general, it only shows that the model performs well on data similar to what it has seen during training. We suspect that in general the performance would be fairly poor, since our models have limited vocabulary and suffer from several problems, as we will now discuss.

Through observations we noticed the models tended to use some words more frequently than others, completely out of context. A logical explanation to this is that these words were more frequent in the data sets. An alternative explanation would be that the model was over-fitting on the training data or a combination of both. In the Russian to English data set the occurrence of the name "Tom" was appearing quite often and when we later investigated the issue we found 114657 occurrences of the name Tom. As our models has no mechanism to handle names and our pre-processing did not handle excessive occurrences of names. Our model treated the name as a word that was frequently used in language, which in fact is an incorrect statement. We believe this led to a decrease in the BLEU-score, even though the test set included sentences using Tom.
The models using a Swedish/English dataset we could analyse on sentence-level, since we are familiar with that language. We observed two reoccurring themes when the models failed. As mentioned previously longer sentences were hard to handle, as a result the models often presented a behaviour that could be interpreted as creating a longer sentence out of shorter sentences. In many cases this doesn't make any sense in language. An example of this was the translation:

SE:"Vi har inte mycket pengar, men vi har tillräckligt med pengar för att köpa det vi behöver"
Our NMT: "Nothing before we didn't have no money we have let them we."
Correct translation: "We don't have much money, but we have enough money to buy what we need."

The NMT seems to start a sentence "nothing before we didn't have.." but as the model realises the sentence does not end there, it starts adding key words such as "money" and a common pair such as "we have" and the sentence no longer makes sense. This behaviour was observed, but we did not further investigate how often this occurred. Another phenomena was that when the sentence used commas it seemed to get confused about the whole sentence - when removing a comma the translation often made a lot more sense. Meanwhile, question marks did not seem to affect the performance, we actually observed that using questions marks in the right conditions the translations are more accurate. As with the previous behaviours we did not run tests of this behaviour that are sufficient to draw any conclusions but it is an interesting topic that could be investigated in future work. The last issue worth mentioning is that special characters are not used in our model, this obviously causes some issues with certain translations.

5.2 Attention visualisation

A very interesting behaviour seen in Figure 2, especially in the left plot, is that the plots tend to stay roughly on the diagonal. This means that Swedish seems to have trivial alignments with English, like French would [2], meaning that the meaning of the sentences can roughly be obtained by translating word by word. But we see in the right plot of Figure 2 that this is not the end of the story for our Swedish to English model. The model also sometimes uses the words surrounding some word to better capture the meaning of a sentence, as seen in the right plot of Figure 2. It is also fascinating to see how in the right plot of Figure 2 the words were used in different parts of the translation. Seems

like the focus of the words was shifted, using the words following the focus word in order to translate, instead of just looking at the focus word on its own, which is a behaviour expected from the GRU architecture. This method of visualisation is great to observe this kind of behaviour. The BiGRU architecture has a slightly different behaviour, since because of its structure it can look forward and backward in the input sequence, so it can consider words before and after the focus word when translating.
In Figure 3 we show some examples when the model tends to fail. It seems like the word "birthday" was used in contexts involving birds in the training set, since several translations we saw from the model included bird related words, such as birdcage, when translating the word "födelsedag" (birthday in Swedish). Even if the input sentence has no bird related words, this can still happen. It can be seen in the left plot of Figure 3, however here the model becomes confused at the end of the sentence and chooses a fairly random word to finish the sentence with.
The right plot of Figure 3 shows a seemingly simple sentence, that the network inexplicably fails at. Perhaps the Swedish word "gemensamt" has not been used in such a way before in the training set, and the network becomes confused and finishes the sentence in a way it thinks is best, without paying too much attention to the original input. We also see that the dot and the $< end >$ token of the input sequence get strangely large attention weights, a characteristic behaviour in bad translations.

6 Conclusion

6.1 Summary of key results

In conclusion, we trained the models on three different languages and three different dataset sizes. We used the BLEU score to evaluate the translation performance of the models, which performed on a similar level with insignificant differences. Our experiments showed that bi-directional encoder structure performs best on Swedish and GRU performed best on Spanish. The Russian model did not perform as well on the test set despite being trained on more data. We also found that in general the models struggle with longer sentences, proper nouns and mid sentence punctuation.

We learned from this project how a sequence to sequence network with attention, used for translation, works and how it can be implemented in Tensorflow. Furthermore, we observed and understood the effect of changing the architecture of this network on the translation performance. Through testing our of our models on multiple input sentences, we observed and understood the strengths and weaknesses of this type of network. Most importantly we got to evaluate the performance of these models on different languages and architectures using the commonly used method, the BLEU score. Lastly, we got real insight into the inner workings of our models by plotting the attention weights for certain input sentences, which provide a simple way of seeing which parts of the input sequence the model considers at each stage of translation.

6.2 Future research ideas

There are many possible ideas for future extensions of this project. One experiments we considered doing was trying to use pre-computed word vectors in the input sentences, using standards such as word2vec or Glove, and see how they affect performance. Another experiment we considered performing was to visualise how the network extracts meaning from the input by, given an input sequence, inspecting the hidden vectors of the decoder using PCA. In this reduced dimension space, hidden vectors for sentences with similar meaning should be close together, if the network is good at translating [7]. Another future extension could involve comparing performance of different types of attention mechanism, as done in Luong, et al. (2015) [3], or even compare performance of our models with different architectures used for performing NMT, such as the Transformer [8]. A different type of possible future network improvement is using beam search in the decoder. Beam search is similar to greedy search, but instead of just one, we consider b best hypotheses at each time step, where b is the width of the beam. We saw this suggestion in the NMT with sequence to sequence models tutorial [4]. The idea is to better explore the search space of all possible translations by keeping around a small set of top candidates as we translate. This method is not guaranteed to find the optimal solution, but is a quite efficient potential improvement to include in the network.

References

[1] Norvig Russell. *Artificial Intelligence: A modern approach*. Pearson, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, page arXiv:1409.0473, September 2014.

[3] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. *arXiv e-prints*, page arXiv:1508.04025, August 2015.

[4] Graham Neubig. Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. *arXiv e-prints*, page arXiv:1703.01619, March 2017.

[5] Boxing Chen and Colin Cherry. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[6] Milam Aiken. An updated evaluation of google translate accuracy. *Studies in Linguistics and Literature*, 3:p253, 07 2019.

[7] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv e-prints*, page arXiv:1611.04558, November 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, June 2017.