

# Converting Tense Of A Sentence Using NMT Technique



By

**Rida Fatima**

**Fall 2017-MS(CS-07)-00000205562**

Supervisor

**Dr. Seemab Latif**

**Department of Computing**

A thesis submitted in partial fulfillment of the requirements for the degree  
of Masters of Science in Computer Science (MS CS)

In

School of Electrical Engineering and Computer Science,  
National University of Sciences and Technology (NUST),


Islamabad, Pakistan.

(July 2021)

## Approval

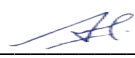
It is certified that the contents and form of the thesis entitled "Converting Tense of a Sentence using Neural Machine Translation" submitted by RIDA FATIMA have been found satisfactory for the requirement of the degree

Advisor : Dr. Seemab Latif

Signature: 

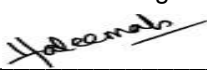
Date: 13-Aug-2021

Committee Member 1:Asad Ali Shah

Signature: 


Date: 13-Aug-2021

Committee Member 2:Engr. Haleemah Zia

Signature: 

Date: 13-Aug-2021


Committee Member 3:Dr. Muhammad Ali Tahir

Signature: 

Date: 13-Aug-2021

## **THESIS ACCEPTANCE CERTIFICATE**

Certified that final copy of MS/MPhil thesis entitled "Converting Tense of a Sentence using Neural Machine Translation" written by RIDA FATIMA, (Registration No 00000205562), of SEecs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: \_\_\_\_\_  \_\_\_\_\_

Name of Advisor: Dr. Seemab Latif \_\_\_\_\_

Date: \_\_\_\_\_ 13-Aug-2021 \_\_\_\_\_

Signature (HOD): \_\_\_\_\_

Date: \_\_\_\_\_

Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_


# Dedication

Dedicated to my parents,my family, and my teachers who have been a constant support even when I was down and out

## Certificate of Originality

I hereby declare that this submission titled "Converting Tense of a Sentence using Neural Machine Translation" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEecs or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEecs or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: RIDA FATIMA

Student Signature: 

# Acknowledgment

All gratitude is to Almighty Allah, the source of knowledge and wisdom, the most Gracious, the most Merciful, who blessed me with the acumen to complete this thesis successfully.

I would like to express my deepest gratitude to Dr. Seemab Latif, my thesis supervisor, for her continuous support and motivation. None of this would have been possible without their perceptiveness, motivation and command on the subject.

My appreciation goes to my committee members who monitored my work and took effort in reading and providing valuable comments on my presentations and other thesis documents.

I would also like to acknowledge my family and friends who have always been there to cheer and support me. Their words of encouragement and appreciation have always inspired me to keep going in life.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	3
1.2	Problem Statement . . . . .	3
1.3	Solution Statement . . . . .	4
1.4	Objective and Research Methodology . . . . .	4
1.5	Thesis Organization . . . . .	5
<b>2</b>	<b>Background Study</b>	<b>6</b>
2.1	Machine Translation . . . . .	6
2.2	Statistical Machine Translation . . . . .	7
2.3	Neural Machine Translation . . . . .	8
<b>3</b>	<b>Literature Review</b>	<b>10</b>
3.1	Neural Machine Translation . . . . .	11
<b>4</b>	<b>Methodology</b>	<b>15</b>
4.1	Objective and Research Methodology . . . . .	15
4.1.1	First Step: Dataset Generation . . . . .	15
4.1.1.1	Gathering of sentences . . . . .	16

4.1.1.2	Cleaning of sentences . . . . .	16
4.1.1.3	Data-set Generation . . . . .	16
4.1.2	Second Step: Pre-Processing of dataset . . . . .	17
4.1.2.1	Data Transformation . . . . .	17
4.1.2.2	Data Cleaning . . . . .	18
4.1.3	Third Step: Selection of NMT model . . . . .	18
4.1.3.1	Encoder-Decoder based Neural Machine Trans- lation . . . . .	18
4.1.3.2	Encoder-Decoder model with Attention . . .	19
4.1.3.3	Encoder-Decoder model with pre-trained em- bedding . . . . .	20
4.1.4	Fourth Step: Output and results . . . . .	20
<b>5</b>	<b>Implementation and Analysis</b>	<b>22</b>
5.1	Data Pre-processing . . . . .	22
5.1.1	Data transform, Clean and Preprocess . . . . .	22
5.1.2	Tokenize . . . . .	23
5.2	How Neural Machine Translation algorithm works . . . . .	24
5.2.1	Training . . . . .	26
5.2.2	Testing . . . . .	33
<b>6</b>	<b>Results and Discussion</b>	<b>35</b>
6.1	Environment . . . . .	35
6.2	Results . . . . .	36
6.2.1	Accuracy Plots . . . . .	36
6.2.2	Attention Plots . . . . .	40



6.2.3	BLEU Score . . . . .	44
6.2.4	Corpus BLEU score . . . . .	44
6.2.5	Sentences BLEU score . . . . .	44
<b>7</b>	<b>Conclusion</b>	<b>46</b>
7.1	Future Work . . . . .	46

# List of Abbreviations

SMT	Statistical Machine Translation
NMT	Neural Machine Translation
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Networks
GRU	Gated Recurrent Units
NN	Neural Network
BLEU	Bilingual Evaluation Understudy
AAC	Augmentative and alternative communication

# List of Figures

1.1	Encoder-Decoder RNN Model [4]	2
2.1	Encoder-decoder architecture – example of the general approach for NMT	9
4.1	Encoder-decoder sequence to sequence model [18]	19
4.2	Encoder-decoder model with attention [19]	20
5.1	Source and target Sentences	23
5.2	Neural machine translation – example for translating a source sentence “I am a student” into a target sentence “I was a student”. Here, “_” marks the end of a sentence	25
5.3	Encoder-decoder model implemented	29
5.4	Attention based Encoder-decoder model[21]	30
5.5	Encoder-decoder model implemented	32
6.1	Encoder-decoder model Accuracy Plot	36
6.2	Attention mechanism based Encoder-decoder model Accuracy Plot	37
6.3	Pre-Trained Embeddings Model Accuracy Plot	38

*LIST OF FIGURES*

---

6.4	Results Achieved . . . . .	39
6.5	Attention Plot1 . . . . .	40
6.6	Attention Plot2 . . . . .	41
6.7	Attention Plot3 . . . . .	42
6.8	Attention Plot4 . . . . .	43
6.9	Corpus BLEU score . . . . .	44

# Abstract

The ultimate rewarding goal, for me, for an intelligent system is being able to communicate seamlessly like human. Although there is great progress in the field of Machine Translation through Statistical Machine Translation (SMT) over last few years but SMT systems have become increasingly complex due to its so many independent components and low translation quality that does not satisfy users, rendering it extremely difficult to make further advancements. Recently, due to emerging of Neural Machine Translation (NMT) has given a promising solution to machine translation problem. At the core, NMT model is deep neural network with billions of neurons to learn directly the map for conversion of sources sentences to target. NMT is a lot powerful due to it being an end-to-end framework. Its performance is significantly better than SMT in long range dependencies capturing and generalizing well to unseen texts. This thesis presents how I used NMT in conversion of tenses of sentences as traditional Classifier based statistical models although translate source language to target language but in doing so accuracy and fluency was lost.

# Chapter 1

## Introduction

Machine Translation (MT) is a field of computational linguistics that uses software to translate speech or text from one language to another. This field has drastically evolved since 1940 and has been in use by translation industry to provide support for law firms, government officials and Language Service Providers. The challenges address by MT include polysemy, ambiguity, anaphora resolution, induction and context identification yielding to different paradigms for translation including rule based, statistics based and hybrid models. But over the last few years a new system incorporating neural networks have emerged known as Neural Machine Translation (NMT) which is starting to displace older models. The reason behind this displacement is the ability of NMT models to tackle context based generation of translation, absent in previous models. This is possible due to the use of vector representations and internal states. [1] NMT models also offer simpler transition as with single sequence model and prediction is based on entire source sentence using Recurrent Neural Network (RNN) and attention mechanism. NMT

tools such as tensor2tensor, OpenNMT, Nematus, Sockeye, and CytonMT perform remarkable for various inputs but human like translation ability is still a challenge. [2] Model architectures have been developed to improve the quality of translation, incorporating various parameters like embedding size, cell type, number of layers and many other with an ongoing research to develop a system yielding ideal results. [1] Encoder decoder models comprise of two sub units. Encoder encodes the entire sentence into a fixed size vector and decoder generates output sentence while reading through context vector as shown in Figure 1. [3] Instead of using simple encoder decoder, LSTM encoder decoder architecture is used which has feedback connection and attention is proposed to help in translation and alignment. [4]

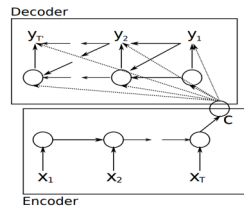


Figure 1.1: Encoder-Decoder RNN Model [4]

Although many translation platforms such as Microsoft Translator and Google Translate offer real time, highly accurate translations but some industries require highly specific domain related translations such as medical, military, education, ecommerce, finance, government and technology to improve relevancy and accuracy which is difficult to achieve when models are trained on generic data[5].

## 1.1 Motivation

One of the industry where neural machine translation has great potential is in Augmentative and alternative communication(AAC) for autistic children. Children that are autistic have difficulty understanding or using spoken language i.e. in pronouncing words, therefore, they require help in development of language skills. These children use speech generating device when developing language skills. The way these devices work is that autistic children choose icon on speech generation device that represents what they want to 'say'. So, if a child wants to eat mango, they can push the button with picture of mango. The device then plays in computer generated voice, ' I want to eat mango'. Using this, autistic children will gradually learn what to say when they want to eat mango. However, this speech generation approach is very basic. Due, to this language skills that gets developed in children with autism, are very limited. So, there is a need to develop an approach through which autistic children can deepen their understanding of language by understanding how different sentences relate to each other in past, present and future.

## 1.2 Problem Statement

Usage of correct verb tenses hold a significant importance in decoding events of a text and tenses vary from one language to other but the transition of tenses from past to present and present to future is difficult to determine even in one language. Classifier based statistical models [6] were built for



translating source language to target keeping verb tense structure intact but in doing so accuracy and fluency was lost. Furthermore, transition of verb tense within one language lacks research and also need to be explored.

### **1.3 Solution Statement**

For this purpose, a highly sophisticated NMT model employing Keras with attention mechanism is built to translate present tense sentence to past tense sentence and vice versa in English language to set a milestone in research and development and lay grounds for some important features for better understanding tense structure of English language. The model is built using Long Short-Term Memory (LSTM) encoder decoder architecture with attention mechanism and adam optimizer. google pre-trained embedding were used and monolingual English parallel data for present and past tense sentences was collected, which is cleaned and tokenized through nltk toolkit for better performance.

### **1.4 Objective and Research Methodology**

The objectives of this research are as follows:

- Gathering of sentences from various sources.
- Cleaning and separation of sentences into its tense category.
- Creating a script that converts simple sentences to all other tenses using rule based approach to generate simple sentences.

- Manually converting compound and complex sentences to all of its tenses forms.
- Pre-processing generated dataset.
- Creating a neural machine translation model based on recurrent neural network techniques and training it on corpus generated.
- Performing comparative analysis between classical machine translation approach and neural machine translation approach.

## 1.5 Thesis Organization

This thesis report has been organized into six main chapters. Chapter one presents introduction of our research topic by describing our motivation, problem statement, solution statement, and objective and research methodology. Chapter-Two, Background Study, covers the basics, a reader would need to know to understand this thesis report. Chapter-Three, Literature Review, takes a deep dive in to the work which has been performed already on the field of neural machine translation. Chapter-Four, Research Methodology, discusses the four-step research methodology used in the completion of this thesis. Chapter-Five, Model Implementation and Analysis, and Chapter-Six, Results and Discussion, explain the selection and testing of the models, and discusses and compares the results with other methods. Lastly, Chapter-Seven, Conclusion, contains the justification of the results along with the scope of future work.

# Chapter 2

## Background Study

This chapter covers basic and important overview of background that a reader would need to understand this thesis report properly. This chapter will start by explaining what machine translation is. Then will explain types of machine translation such as statistical machine translation and neural machine translation.

### 2.1 Machine Translation

Human languages are diverse with thousands of languages spoken worldwide. As civilization advances, the need for seamless communication and understanding across languages becomes more and more crucial. Machine translation (MT), the task of teaching machines to learn to translate automatically across languages, as a result, is an important research area. But, due to natural ambiguity and vastness of human language, the task of machine translation gets very difficult. Traditional machine translation meth-

ods uses rules to convert text from one language to another. These rules are developed by linguists, which is the key limitation of these approaches as expertise is required to develop these rules, and these experts have to design vast number of rules to accurately convert text from one language to another. Eventually, due to these limitations, we have moved to statistical based machine translation and artificial intelligence based machine translation approaches.

## 2.2 Statistical Machine Translation

In statistical machine translation(SMT) approach, statistical models are used to translate text from one language to another language. The most used approach in SMT is sequence based. It works by selecting a sentence S which translator produced for given sentence T. The sentence S is selected in such a way to minimize the error by selecting the most probable sentence S given T  $\Pr(S|T)$ . [8] The key benefit of Statistical Machine Translation is that it does not require linguists to specify the rules of translation, as it is data driven, requiring corpus of examples. It only requires, statistical model to be built, that assigns high probabilities to good translations and low probabilities to bad translations. Due to this, this approach quickly outperformed classical machine translation techniques and become the de-facto standard. The SMT based approach is although effective, but it losses broader nature of target text by keeping a narrow focus on phrases being translated. Due to hard focus on data-driven approach, SMT sometimes ignores the important syntax distinctions by linguists. Thus, statistical based machine translation requires

careful tuning of multiple parameters to have desirable translation.

## 2.3 Neural Machine Translation

Statistical machine translation (SMT) has been successfully used in many commercial systems, but it does have a drawback. The translations are done locally, phrase by phrase and long-distance dependencies are often ignored. Neural Machine Translation (NMT) is a new approach that solves this aforementioned problem. NMT is a machine learning approach which has single big neural network (with millions of artificial neurons) that is designed to model the entire MT process[9]. NMT requires minimal knowledge of domain. It just needs a parallel corpus of source and target sentence pairs, which is similar to SMT but with far less preprocessing steps to built a model. It is an end-to-end approach which works by predicting the likelihood of a sequence of words unlike SMT where multiple components of SMT system have to be learned. NMT initially used multilayer perceptron neural network models for machine translation. These models have limitation that input sequence should have fixed length. This problem is solved by Recurrent Neural Networks which is a powerful architecture for sequential data, that allows for variable length input sequences. These Recurrent Neural Networks uses encoder-decoder architecture in which encoder neural network first reads a source sentence and then encodes it into variable length vector. The decoder then takes encoded vector as input and outputs the translation. This whole encoder-decoder system is jointly trained in such a way, to maximize the probability of correct translation given the input sentence S.

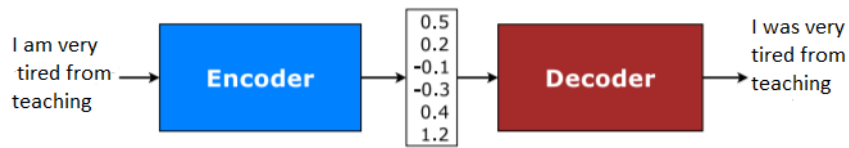
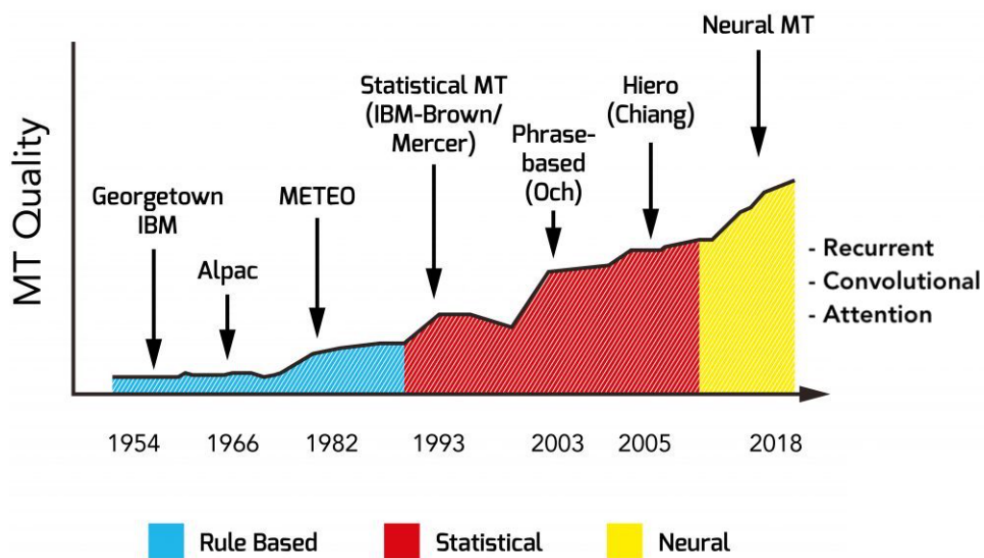


Figure 2.1: Encoder-decoder architecture – example of the general approach for NMT

Through the use of these RNNs models, better generalization can be done to very long sentences by capturing long range dependencies without storing large language models as in SMT. Due to these benefits NMT is currently state of the art approach. As, it can be seen in a chart below.



NMT offers highest quality translation. This is one of the main reason why google switched to NMT for google translation in 2016[7], after using statistical method for many years.

# Chapter 3

## Literature Review

Neural Machine Translation has been one of the focal points in the sequential neural networks. One of the main task in neural machine translation on which researchers are working is conversion of text from one language to another. Over the years, researchers have tried various implementations to solve this challenging task. Due, to this a lot of data-sets of different languages are available to train on different architecture, along with variety of benchmarks to compare the results with. But, no research has been done before to capture the transitioning of tenses in English language before using neural machine translation. Although, statistical models were there for translating one tense to another, keeping verb tense structure intact but in doing so accuracy and fluency was lost. So, there is a need for accurate conversion of English tenses, which I have implemented using neural machine translation. To achieve desirable results I have done literature review of various NMT architectures used in language conversion task along with generating my own data-set.

### 3.1 Neural Machine Translation

Zhifeng Chen[10],uses deep LSTM network which has 8 encoder and 8 decoder layers using residual connections. Attention connections from the decoder network to the encoder are used in this paper. Authors also use beam search technique that employs a length-normalization procedure which uses a coverage penalty, that tries to generate output sentence which covers most of the words in source sentence. Authors have tested this model on the WMT'14 English-to-French and English-to-German benchmarks, this model achieved competitive results at that time. This architecture works great but it has an issue. When training with larger vocabulary, performance of this architecture starts to decrease if we do not increase the complexity of the model.

Melvin Johnson [11], proposed a solution that uses single NMT to translate between multiple languages. This paper has used same architecture as google machine translation system(GNMT) with addition of direct connection between encoders and decoders. It adds Artificial token at the beginning of the input sentence which specifies the required target language. After the addition of artificial token to input data, model is trained with all multilingual data consisting of multiple language pairs. For issue of translation of unknown words, shared word piece model was used. Results were evaluated using standard bleu score metric. This paper provided a simple approach for multilingual NMT. Advantage of this architecture is that it improves the translation quality of languages that have low resource.



Matthieu Felix[12], introduced the concept of word embeddings in NMT which solves the problem of degrading of NMT performance in case where large scale parallel corpora cannot be obtained. Authors in this paper claimed increase in upto 20 BLEU points when pre-trained word embeddings are used in NMT. Architecture authors used was standard 1-layer encoder-decoder model with a beam size of 5. Pretrained word embeddings that were used, were trained on fastText. These word embeddings incorporate character-level, phrase-level and positional information of words and are trained using CBOW algorithm . embedding layer weights were initialized using these pre-trained word vectors in model where pretrained word embeddings were used and using Glorot uniform initialization in model without pretrained word embeddings to initialize the weights of embedding layer. During training, Adam optimizer was used with batch size of 32 and initial learning rate of 0:0002, decaying the learning rate by 0:5 when development loss decreases .Model's performance is evaluated using BLEU metric. Major conclusion drawn from this paper were that models trained with pretrained embeddings have better BLEU score when there is low training data and pre-trained embeddings are more effective for similar translation pairs.

Philipp Koehn[13], explored the challenges faced by Neural Machine Translation and made comparison of how accurate the results were with NMT given its problems, compared to statistical machine translation. The problems which the author highlighted were that NMT have poor quality in out of domain translation. Another problem was that, NMT systems were very dependent on data. More the amount of data, the better the results were.

Another issue faced in NMT was that, although NMT had better accuracy than SMT on low frequency words but NMT still show weakness in this aspect. Another common problem in NMT was of low accuracy in extremely long sentences. Also, beam search decoding only improves translational accuracy of narrow beams but its performance degrades when exposed to a larger search space. To validate how NMT performs compared to SMT with these problems, common toolkits for NMT (Nematus) and traditional phrase-based SMT (Moses) with common data sets were used. After carrying out comparison between both models, authors concluded that although NMT accuracy was more than SMT but it did not show robust behaviour when test data conditions significantly differ from training data.

Yoshua Bengio[14], proposed a method based on importance sampling which allows the using of very large target vocabulary without increasing training complexity to address the limitation of NMT of unable to handle large vocabulary. It does this by reduction of complexity in computing of normalization constant of the output word probability in neural language models. Through this approach authors showed that it was possible to do decoding efficiently even with the model having a very large target vocabulary. The authors used RNNsearch architecture, with 30k source and target words. This approach was tested on English→French and English→German translation tasks and it was observed that its performance was better than the best single NMT model of that time by 1 BLEU point.

Mathias Muller[15], had done the comparison between attention networks

in RNN and CNN. As, CNN with attention network performed better in neural machine translation than RNNs as due to CNNs providing shorter networks path to connect between distant words. So, it was speculated that CNNs with self attention are better to model long range dependencies. Author of this paper wanted to test the above argument. For this, authors tested CNNs and RNNs with self-attention modules on two tasks. First one is subject-verb agreement where there is requirement of capturing long-range dependencies and second one is word sense disambiguation where extraction of semantic features is required. Sockeye toolkit was used for evaluation of CNNs and RNNs models which is based on MXNet. Training was done using Adam optimizer with mini-batch of size of 4096 tokens. Model checkpoints were saved after every 4000 updates. The initial learning rate was set to 0.0002 which was changed by multiplying with 0.7 if validation set performance did not improve for 8 checkpoints. All the neural networks have 8 layers. Performance was evaluated by scoring contrastive translation pairs. After experiments, it was concluded that CNNs did not outperform RNNs in modelling subject-verb agreement over long distances but attention CNNs were better in word sense disambiguation due to their ability to extract semantic features from the source text.

# Chapter 4

## Methodology

### 4.1 Objective and Research Methodology

Based upon the aforementioned research objectives and solution statement, a four-step research methodology was adopted to accomplish the objective and reach the solution statement.

The four-step methodology is explained below:

#### 4.1.1 First Step: Dataset Generation

Steps used to generate data-set are as follows:

- Gathering of sentences from various sources
- Cleaning of sentences gathered
- Dataset generation

#### **4.1.1.1 Gathering of sentences**

The major datasets used in sentence collection are as follows:

- Project Gutenberg[16]
- manythings.org[17]

Manythings.org website is used to gather sentences for data-set. This website has over 300,000 English sentences related to English words and word families. Furthermore, this website has also many English data-set related bilingual pairs available which are also used in gathering of sentences for data-set.

Project Gutenberg is another website I have used for sentence collection. It has large collection of free ebooks on various genres that are available in plain text for variety of languages. These ebooks are used to extract complex sentences from literature to create data-set.

#### **4.1.1.2 Cleaning of sentences**

The sentences that are gathered from above sources are in mixed tenses form. So each sentence is then categorized according to its verb form and then removal of punctuation's, annotations, short forms is done from each sentence.

#### **4.1.1.3 Data-set Generation**

More than 500,000 sentences in english language have been gathered and cleaned. Converting such a large number of sentences into other tenses is

very humongous task. So,I have developed a python script based on spaCy python library that successfully converted sentences into other tenses except for few complex sentences whose translation is done manually. Spacy is an advance Natural Language processing python library which is open-source. Let us look at a stripped version of algorithm I have developed for sentence's tense conversion.

---

**Algorithm 4.1** Dataset Generation

---

```
Input:
Sentence:
Model : Python Script
Output : Converted Sentence to other tense form
//Step 1
    // Read sentence from excel file where training data is stored
//Step 2
    // Detect sentence's tense type using its verbs
//Step 3
    // Convert sentence's tense to other tense using grammatical rules
I have implemented using spaCy which take into account both noun and
verb of sentence
//Step 4
    // Repeat step 1 to 3 until all dataset has been converted,to all
other tenses. To demonstrate my approach, I have converted present only
tenses to past tenses and vice versa,
```

---

## 4.1.2 Second Step: Pre-Processing of dataset

In second step, pre-processing is done on the dataset I have generated. For that I have taken following steps.

### 4.1.2.1 Data Transformation

As dataset is in excel file so it has to be transformed in a way that can be used in neural machine translation model. For that, I have converted dataset

in to source and target list.

#### **4.1.2.2 Data Cleaning**

Afterwards, data is cleaned to ensure it is suitable for training using NMT model. For that, all sentences are converted into lowercase, all unnecessary digits, letters and punctuation's are removed.

In last step, Addition of 'start' and 'end' tag to target sentence is done. This will help the decoder to know from where to start decoding and when to end.

#### **4.1.3 Third Step: Selection of NMT model**

Literature review is done of various models to select that model whose results outperforms others. As this is the first time, neural machine translation has been done on English tenses conversion so our results act as benchmark for future studies.

So, in this step three neural machine translation models are used to train on my dataset. The models used are as follows:

##### **4.1.3.1 Encoder-Decoder based Neural Machine Translation**

The encoder-decoder model is used mostly for sequence-to-sequence prediction problems in RNNs so it is used here. The approach is to use two RNNs, one to encode the input sequence data called encoder and other to decode the input sequence which is encoded into target sequence called decoder.

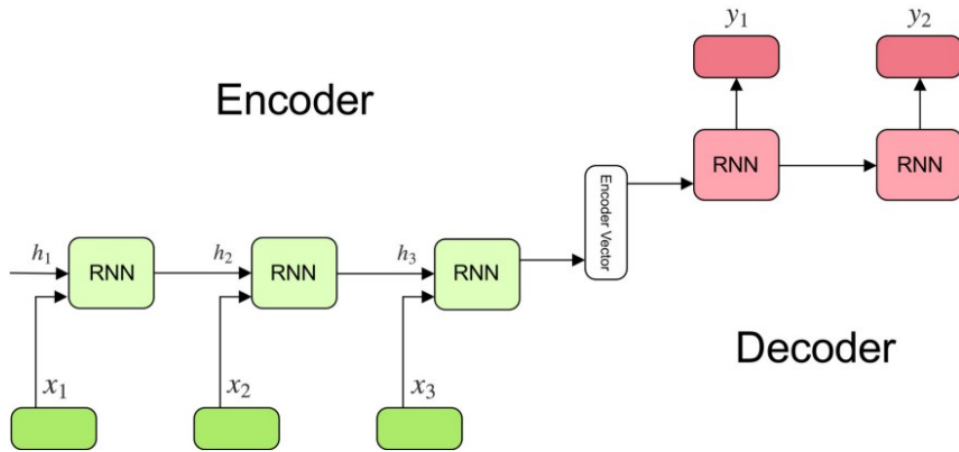


Figure 4.1: Encoder-decoder sequence to sequence model [18]

#### 4.1.3.2 Encoder-Decoder model with Attention

Encoder-Decoder has an issue that it compresses all the information from source sentence into a fixed-length vector, thus making it difficult to cope up with long sentences. Due to this, attention based model is used for better performance. What the attention model does is that it develops a context vector that is specifically filtered for each time step of output instead of encoding the input sequence into a single fixed context vector. This model performs better in case of complex sentences.



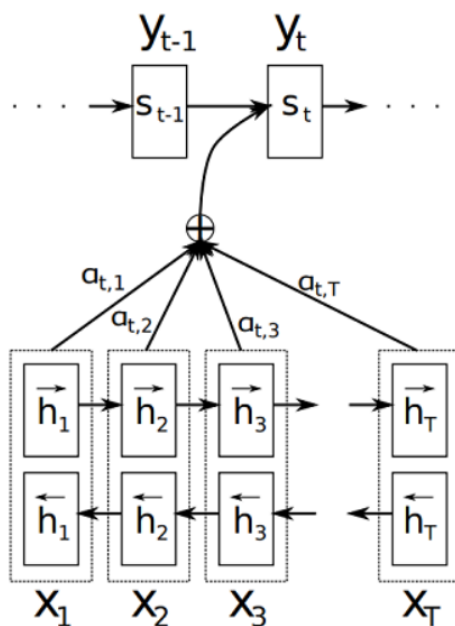


Figure 4.2: Encoder-decoder model with attention [19]

#### 4.1.3.3 Encoder-Decoder model with pre-trained embedding

Performance of NMT suffers in low resource scenarios as large parallel corpora cannot be generated. In these cases pre-trained word embeddings are preferred where embedding layer weights were initialized using these pre-trained word vectors in model. I have created my own dataset of over 500,000 sentences which is more than enough to convert sentence into other tenses but for comparison purposes I have also used pre-trained GloVe embeddings.

#### 4.1.4 Fourth Step: Output and results

Fourth and final step makes up the following:

- Running model several times with both pre-trained embeddings and without pre-trained embeddings.

- Comparison between other models and justification of the results.

# Chapter 5

## Implementation and Analysis

This chapter has three sections. First section, deals with pre-processing of the dataset before it is used for training. Second section is dedicated to detailed algorithm adopted for this research and training details related to them.

### 5.1 Data Pre-processing

Data pre-processing is an important part of Neural Machine Translation. A NMT network needs to be taught properly before it can be used. Data pre-processing is divided into following parts.

#### 5.1.1 Data transform, Clean and Preprocess

Data is first read from excel file and converted into source and target list.

	source	target
124476	The pilots test their guns	The pilots tested their guns
562937	The dog is not holding any objects	The dog was not holding any objects
19107	I still have a lot of things to take care of	I still had a lot of things to take care of
486478	Group of people on roof with animal	Group of people on roof with animal
162646	Why are you speaking in French?	Why were you speaking in French?

Figure 5.1: Source and target Sentences

After converting sentences to source and target lists, following operations are performed on both lists.

- Sentences are converted to lower case.
- Quotes are removed from all sentences.
- Digits and unknown symbols are cleaned from sentences.
- Tags are added to sentence. "start" tag is added at the start of sentence and "end" tag is added at the end of the sentence.

### 5.1.2 Tokenize

After preprocessing of dataset, the corpus needs to get vectorized. So, both lists are converted into sequence of integers using tokenizer. First, tokenizer is created and then applied to both source and target sentence list. It is to be noted that only words known to tokenizer will be taken into account when training on data. After tokenization, sequence of integers are created. Here we need to create the sequences with the same length, so post padding on sequences is done.

Lastly, dataset is split into training, validation and testset using 60,20,20 split.

## 5.2 How Neural Machine Translation algorithm works

Neural Machine translation(NMT) models are simply Recurrent Neural Networks(RNNs) that models conditional probability  $p(y|x)$  of translating a source sentence to target sentence. These models achieve this goal through encoder-decoder framework. The encoder, for each source sentence, computes a representation  $s$ . Based on that representation, translation is done by decoder, one target word at a time.

Various NMT models are currently used in literature and mostly use encoder and decoder architecture. At encoder side we have many choices as source sentence is fully observable. Due to this, even convolutional neural network can be used for encoding the source but choices on decoder side are limited as we need to generate the translation. Most popular choice for decoder is unidirectional RNN which I use in my thesis, as it produces translations from left to right which simplifies the beam search decoding.

In this thesis, I have used deep multi-layer RNNs which are unidirectional and have a LSTM as the recurrent unit. I show an example of such model in Figure 5.2.

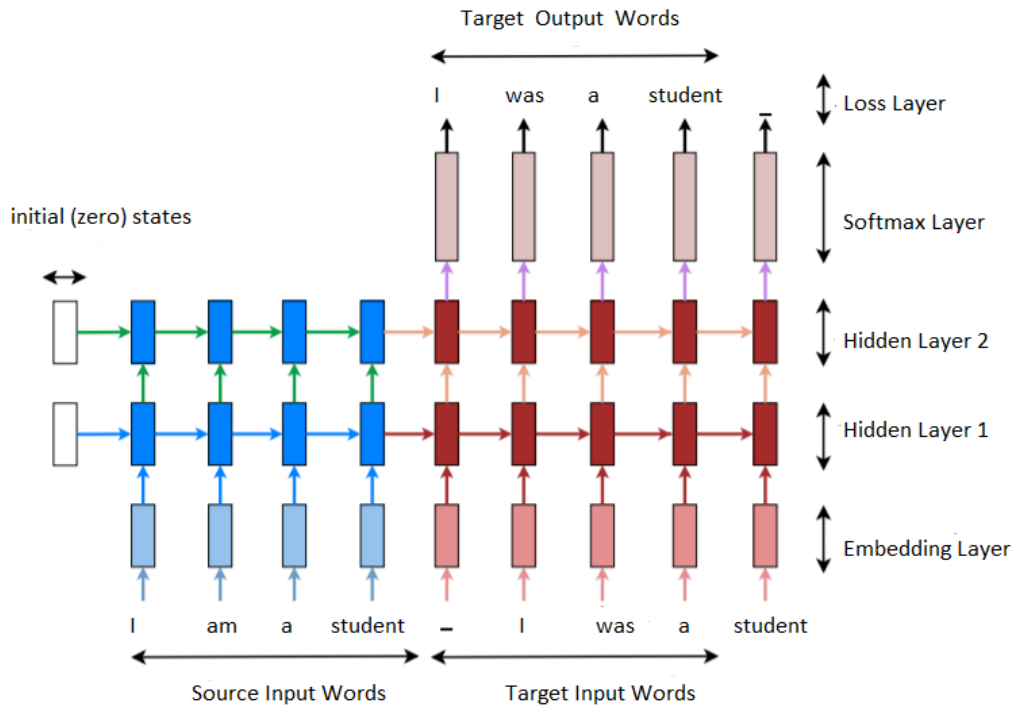


Figure 5.2: Neural machine translation – example for translating a source sentence “I am a student” into a target sentence “I was a student”. Here, “\_” marks the end of a sentence

In this example, I train my model to translate a source sentence “I am a student” into a target one “I was a student”. This NMT model has two RNNs. The encoder without making prediction, only consumes input sentence while decoder RNN processes the target sentence and predict next words.

In more detail, at the bottom layer, first the encoder RNN receives the source sentence until boundary marker \_ is received which indicates that the transition from encoder to decoder has been done. Then target sentence is received at decoder input. For these discrete words at input of encoder-decoder, model looks in embeddings of source and target to get corresponding word

representations. To generate the embeddings we have two options. First, if we have a large dataset then embedding weights are learned during training for each source and target sentence list. Otherwise, if dataset is small then embedding weights are initialized using pretrained word representations such as word2vec and Glove. In this thesis, I have learned these embeddings from scratch instead of using pre-trained embeddings as my training dataset is large. Once these word embeddings are retrieved, they are fed as input to this encoder-decoder model, who can share same weights. The starting states used for encoder is zero vector while, as, decoder has access to source information, so it is initialized with last hidden state of encoder as shown in fig 5.2. where the word "student" is last hidden state of encoder and is passed to decoder side. The hidden unit is connected from layer below to upper using feed-forward weights in vertical direction. While, information from previous time step to next one is transferred using horizontal weights.

Finally, using softmax layer weights, the hidden state at the top layer for each target word is transformed into a probability distribution over the target vocabulary.

### 5.2.1 Training

Neural Machine Translation training is similar to that of recurrent neural network except conditioning on source sentences has to be handled. The training objective for NMT is formulated as:

$$J = \sum_{n=x,y} -\log p(y|x) \quad (5.1)$$

Here,  $(x, y)$  refers to source and target sentence pairs. In the above mentioned architecture, the loss for NMT  $(x,y)$  in forward pass is calculated same as of RNN with only difference, which is that decoder RNN is initialized on source sentence  $x$ 's representations instead of zero states. For, back-propagation phase, the gradient of last hidden state of decoder is passed to encoder. In this way network is trained. The algorithm I have implemented for forward pass and backward pass of NMT are as follows in algorithm 5.1 and algorithm 5.2.

---

**Algorithm 5.1** NMT training algorithm – forward pass

---

**Input** : source sentence  $x$  of length  $m_x$ , target sentence  $y$  of length  $m_y$   
**Parameters:** encoder  $W_e^{encoder}, T_{lstm}^{encoder}$  ; decoder  $W_e^{decoder}, T_{lstm}^{decoder}$   
**Output** : loss  $L$  and other intermediate variables for backpropagation

```

s ← [x,_,y,_] // Length of s is  $m_x + 1 + m_y + 1$ 
 $W_e, T_{lstm}^{1..L} \leftarrow W_e^{encoder}, T_{lstm}^{encoder}$ ; // Encoder Weights
 $h_o^{1..L}, c_o^{1..L} \leftarrow 0$ ; // zero init
for  $t = 1$  to  $(m_x + 1 + m_y)$  do
  //Decoder Transition
  if  $t == (m_x + 1)$  then
    |  $W_e, T_{lstm}^{1..L} \leftarrow W_e^{decoder}, T_{lstm}^{decoder}$ ;
  end
  //Multi-layer LSTM
   $h_t^{(0)} \leftarrow \text{Emb\_LookUp}(s_t, W_e)$ ;
  for  $l = 1 \rightarrow L$  do
    |  $h_t^{(l)}, c_t^{(l)} \leftarrow \text{LSTM}(h_{t-1}^{(l)}, c_{t-1}^{(l)}, h_t^{(l-1)}, T_{lstm}^{(l)})$ ; // LSTM hidden unit
  end
  // Target-side prediction
  if  $t >= (m_x + 1)$  then
    |  $l_t, p_t \leftarrow \text{Predict}(s_{t+1}, h_t^{(L)}, W_{hy})$ ;
  end
end

```

---



---

**Algorithm 5.2** NMT training algorithm – backpropagation pass

---

```
 $dh_{m_x+1+m_y}^{(1..L)}, dc_{m_x+1+m_y}^{(1..L)} \leftarrow 0;$  // Cell and state gradients  
 $dT_{lstm}^{(1..L)}, dW_e, dW_{hy} \leftarrow 0;$  // Model weight gradients  
for  $t = 1$  to  $(m_x + 1 + m_y)$  do  
  //Encoder Transition  
  if  $t == (m_x)$  then  
     $dW_e^{decoder}, dT_{lstm}^{decoder} \leftarrow dW_e, dT_{lstm}^{(1..L)};$  // save decoder gradients  
     $dT_{lstm}^{(1..L)}, dW_e \leftarrow 0;$   
  end  
  // Target-side prediction  
  if  $t \geq (m_x + 1)$  then  
     $dh, dW \leftarrow \text{Predict\_grad}(s_{t+1}, p_t, h_t^{(L)});$   
     $dh_t^{(L)} \leftarrow dh_t^{(L)} + dh;$  // vertical gradients  
     $dW_{hy} \leftarrow dW_{hy} + dW;$   
  end  
  // Multi-layer LSTM  
  for  $l = 1 \rightarrow L$  do  
    // Recurrent gradients  
     $dh_{t-1}^{(l)}, dc_{t-1}^{(l)}, dx, dT \leftarrow \text{LSTM\_grad}(dh_t^{(l)}, dc_t^{(l)}, h_{t-1}^{(l)}, c_{t-1}^{(l)}, h_t^{(l-1)});$   
     $dh_t^{(l-1)} \leftarrow dh_t^{(l-1)} + dx;$  // vertical gradients  
     $dT_{lstm}^{(L)} \leftarrow dT_{lstm}^{(L)} + dT;$   
  end  
   $dW_e \leftarrow \text{Emb\_grad\_update}(s_t, dh_t^{(0)}, dW_e);$   
end  
 $dW_e^{encoder}, dT_{lstm}^{encoder} \leftarrow dW_e, dT_{lstm}^{(1..L)}$  // Save encoder gradients
```

---

Following architecture I have implemented for encoder-decoder NMT to test its accuracy on my dataset. But, this model has low accuracy as shown in results section. Due to which I have to implement NMT with Attention mechanism that works well with complex sentences.

Layer (type)	Output Shape	Param #
embedding_6 (Embedding)	(None, 46, 512)	17111552
lstm_15 (LSTM)	(None, 46, 512)	2099200
lstm_16 (LSTM)	(None, 512)	2099200
repeat_vector_4 (RepeatVecto	(None, 46, 512)	0
lstm_17 (LSTM)	(None, 46, 512)	2099200
dense_6 (Dense)	(None, 46, 512)	262656
dropout_3 (Dropout)	(None, 46, 512)	0
dense_7 (Dense)	(None, 46, 33634)	17254242

Figure 5.3: Encoder-decoder model implemented

The basic concept of Attention is to avoid learning of single vector representation for each sentence. Attention weights decides, how much to pay attention to specific input vectors of input sequence. I have implemented Bahdanau[20] attention mechanism which is additive attention as it does a linear combination of decoder and encoder states.

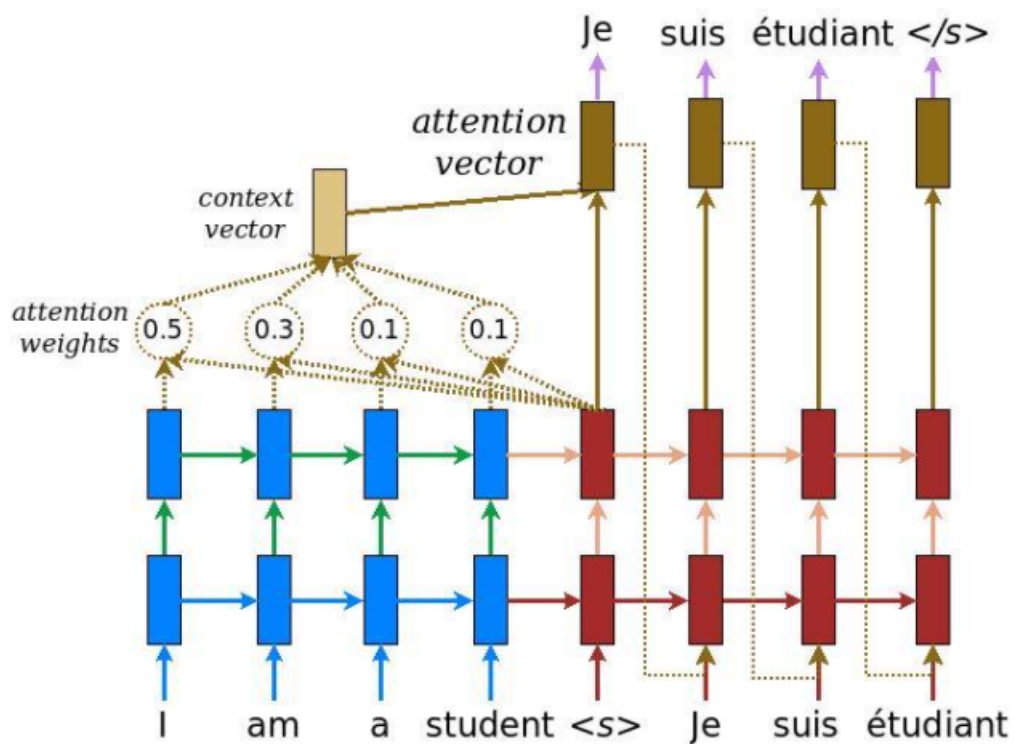


Figure 5.4: Attention based Encoder-decoder model[21]

In this attention mechanism, Context vector is generated by all the hidden states of encoder and decoder, unlike previously where only last encoder hidden state is used. The input and output sequences are aligned with score that is parameterized by feed-forward network. This helps in identifying most relevant parts of source sentence. Using this context vector, model predicts a target word associated with previously generated words and source position. Bahdanau attention equations that are used to implement attention are as follows:

$$e_{ij} = v^T \tanh(W[s_{i-1}; h_j])$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

The architecture I have implemented using attention mechanism is below:

Layer (type)	Output Shape	Param #
embedding_9 (Embedding)	(None, 46, 512)	17111552
lstm_22 (LSTM)	(None, 46, 512)	2099200
lstm_23 (LSTM)	(None, 46, 512)	2099200
seq_weighted_attention_1 (Se (None, 512)		513
repeat_vector_6 (RepeatVecto (None, 46, 512)		0
lstm_24 (LSTM)	(None, 46, 512)	2099200
dense_8 (Dense)	(None, 46, 512)	262656
dropout_4 (Dropout)	(None, 46, 512)	0
dense_9 (Dense)	(None, 46, 33634)	17254242

Figure 5.5: Encoder-decoder model implemented

## 5.2.2 Testing

After training neural translation model , we need it to test our model by translating unseen sentences using our model and measuring their accuracy.

For this, following steps are taken

- In first step we need to pass the test sentence to our model.
- Next, the pre-processing is done on sentence in which it is converted to lower case, its special characters and spaces are removed.
- Afterwards, sentence is then tokenized to convert it into sequence of integers
- If the sentence has different length than current dataset, then it is post padded with 0 to have same length as max source sentence
- In next step, we create input tensor and encoder. Here, the hidden state is initialized to zero. Now, both input vector along with hidden state is passed to it.
- Then, in next step decoder is created. Decoder's first input is 'start' tag and encoder hidden state is used to initialize decoder hidden state. The decoder is passed with decoder input, its hidden state and encoder output. It outputs predictions and attention weights.
- Attention weights are stored and using the context vector, hidden and Decoder input, integer with maximum probability is found

- Integer is then converted to a word into empty array, where each predicted words keep getting appended until 'end' tag is received or max target length of sentence is reached.
- The translation quality is measured using the BLEU metric

# Chapter 6

## Results and Discussion

This chapter has three sections. First section covers the environment that was used to perform these tests. Second, covers the results. Last section, contains discussion and comparison with other Neural Machine Translation models.

### 6.1 Environment

Computation is performed using tensorflow 2.4 with excessive modification to run custom data generation. Computation is performed on NVIDIA RTX 1060 with CUDA 11, with access to almost 6GB of physical ram and 400GB of fast NVMe x4 storage for virtual ram. Having access to fast memory is important due to large number of training data.

While training the batch size is set to 32 and increased afterwards to 512 to improve processing speed, Adam optimizer is used with learning rate that is 0.001 initially and is exponentially reduced afterwards. There was no



limit to epochs per iteration as training was stopped whenever there was no improvement for continuous seven epochs with best weights restored after stopping the training.

## 6.2 Results

### 6.2.1 Accuracy Plots

Three NMT models are implemented and compared. First one is simple encoder-decoder model which is easy to implement and has accuracy of 91% but this model produces inaccurate translations in case of complex sentences.

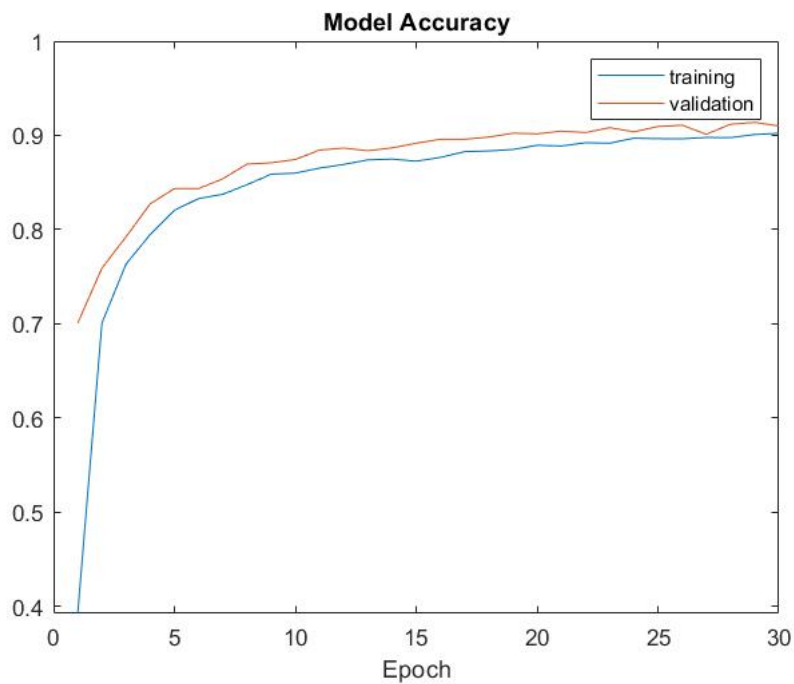


Figure 6.1: Encoder-decoder model Accuracy Plot

Second model, which is the main model I am using is Bahdanau Attention mechanism based Encoder Decoder Model. This model has advantage that it can learn long sentences and complex sentences, due to which my accuracy is increased to almost 96%.

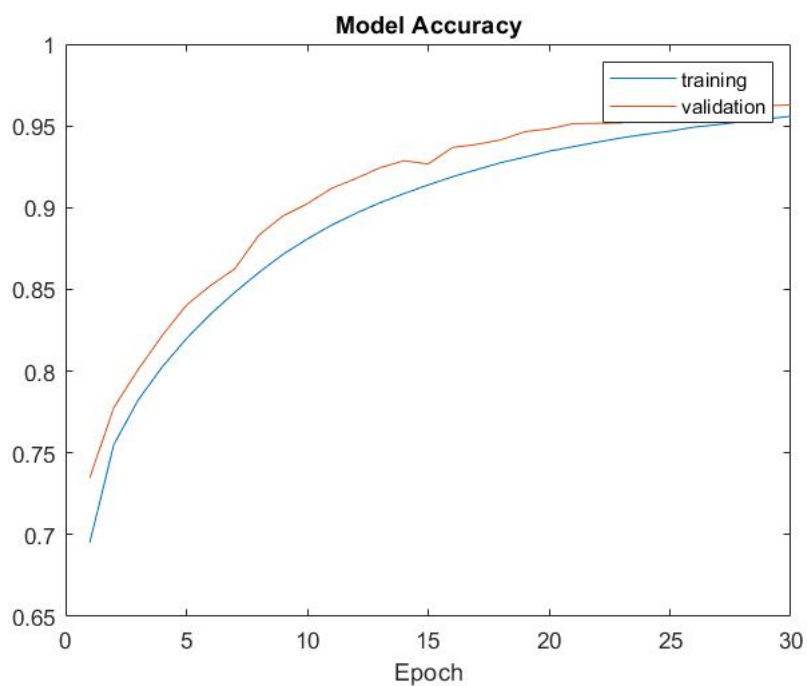


Figure 6.2: Attention mechanism based Encoder-decoder model Accuracy Plot

Third model, which I have used is Attention based Encoder Decoder Model using GLOVE pretrained word embeddings. These embeddings are very helpful when source and target parallel corpora is not very large but nonetheless I wanted to check the effects on accuracy of these word embeddings. Exactly, as I have predicted the accuracy does not have much effect but it is slightly less than what I have achieved using my own dataset. Accuracy is 95%

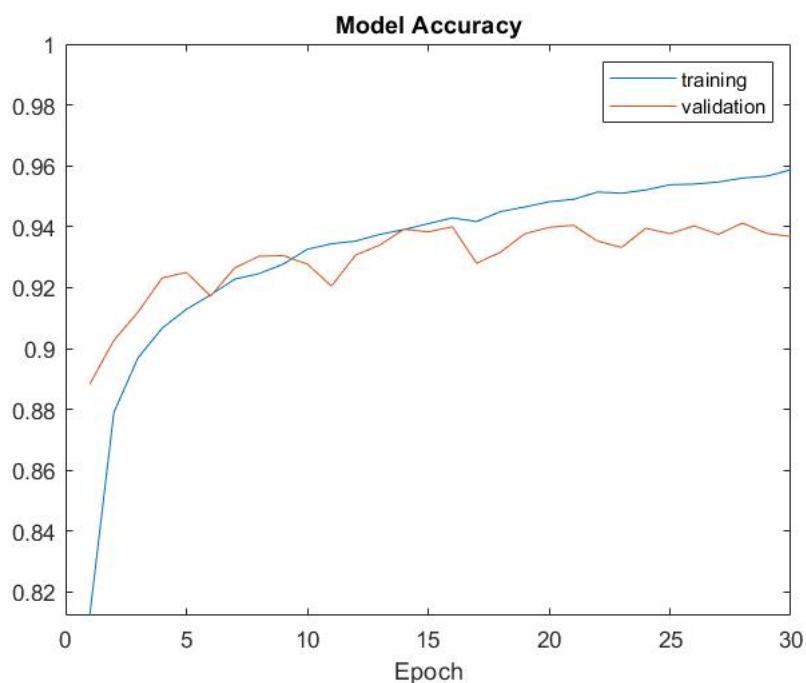


Figure 6.3: Pre-Trained Embeddings Model Accuracy Plot

Few of the test results , that are achieved using Attention mechanism based Model which converts all present tenses to their respective past tenses or past tenses to present tenses. Both are done

	<b>original</b>	<b>actual</b>	<b>predicted</b>
0	a girl is doing laundry	a girl was doing laundry	a girl was doing laundry
1	that is the reason we have to fight	that was the reason we had to fight	that was the reason we had to fight
2	a boy is holding a plant	a boy was holding a plant	a boy was holding a plant
3	a chef is cooking a steak in front of customers	a chef was cooking a steak in front of customers	a chef was cooking a steak in front of customers
4	a woman is holding the hands of two children	a woman was holding the hands of two children	a woman was holding the hands of two children
5	our japanese teacher is very nice to us	our japanese teacher was very nice to us	our japanese teacher was very nice to us
6	the man is sitting inside a car	the man was sitting inside a car	the man was sitting inside a car
7	the snow is on the ground	the snow was on the ground	the snow was on the ground
8	a kid is doing a kick while laying on the ceiling	a kid was doing a kick while laying on the ceiling	a kid was doing a huge while driving on the ceiling
9	almost all the work is done now	almost all the work was done now	too all the work was done now
10	tom is reading it	tom was reading it	tom was reading it
11	that is what makes this so difficult	that was what made this so difficult	that was what made this so difficult
12	a woman is wearing a tank top	a woman was wearing a tank top	a woman was wearing a tank top
13	a man is sitting holding an advertisement sign	a man was sitting holding an advertisement sign	a man was sitting holding an wives sign
14	the man is putting up a poster	the man was putting up a poster	the man was putting up a poster

Figure 6.4: Results Achieved

## 6.2.2 Attention Plots

Few of the attention plots are plotted here for past to present conversion results. These attention maps helps us visualize in which part of sentence ,NMT model is paying attention.

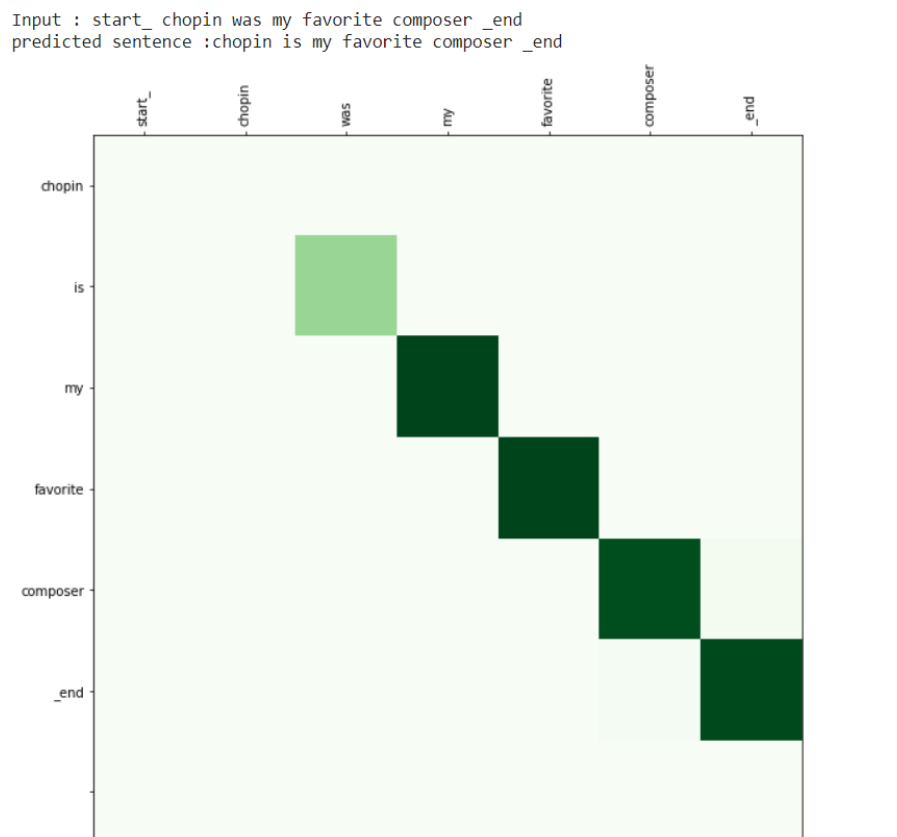


Figure 6.5: Attention Plot1

Input : start\_ chinese characters were difficult to read \_end  
predicted sentence : chinese characters are difficult to read \_end

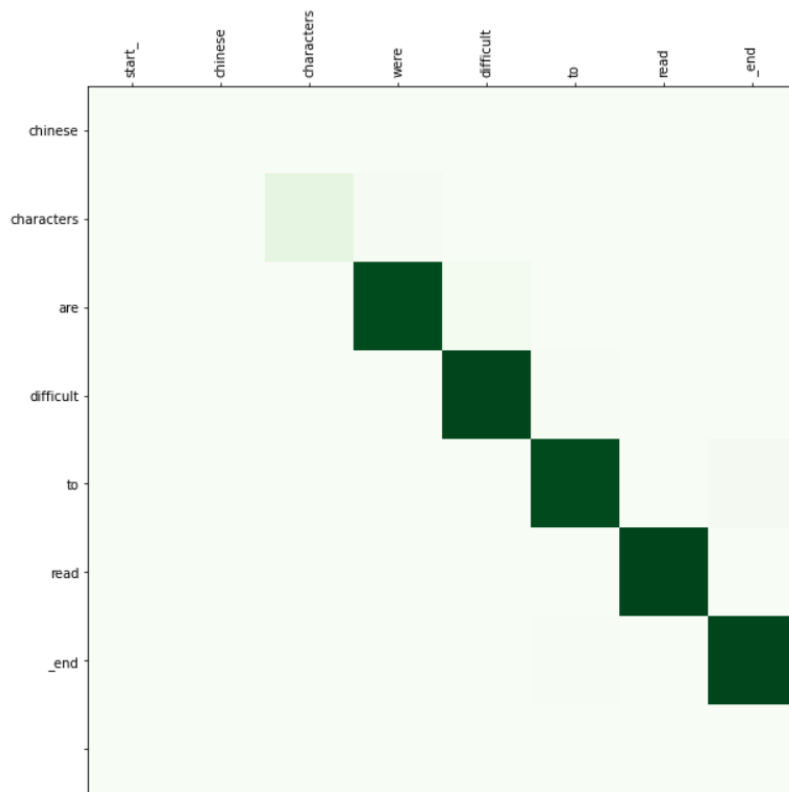


Figure 6.6: Attention Plot2

Input : start\_ children were really looking forward to summer vacation \_end  
predicted sentence : children are really looking forward to summer vacation \_end

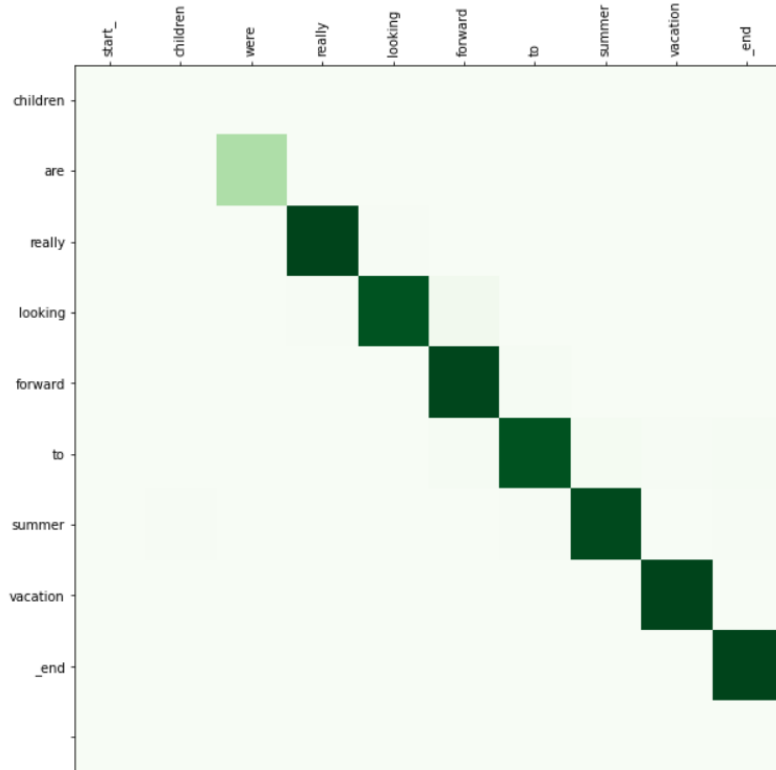


Figure 6.7: Attention Plot3

Input : start\_ children under thirteen years of age were not admitted to this swimming  
 predicted sentence :children under thirteen years of age are not are to this swimming

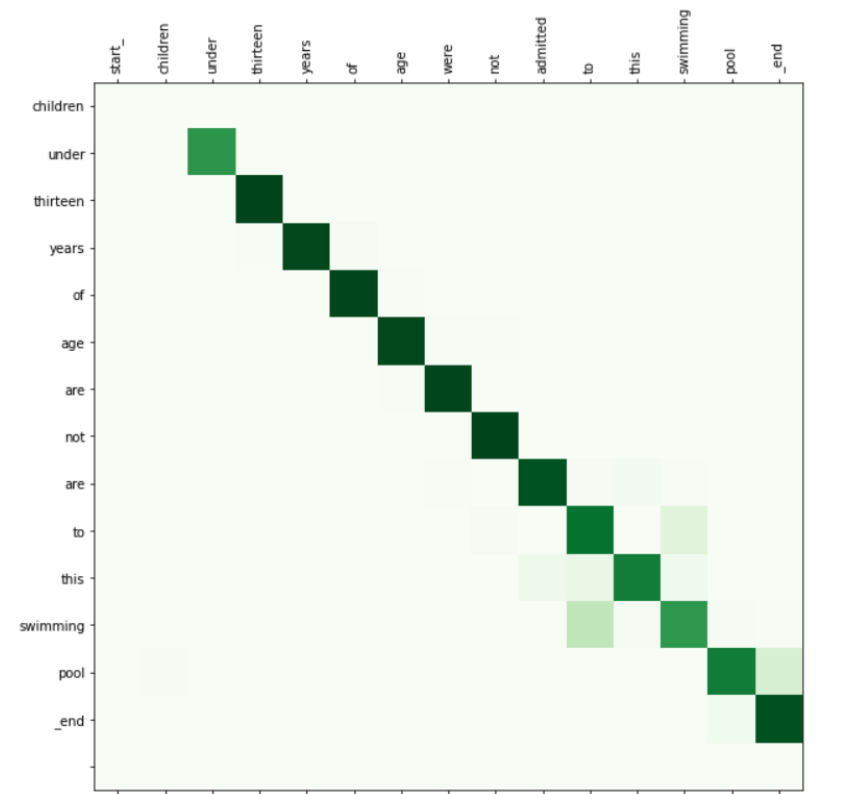


Figure 6.8: Attention Plot4



### 6.2.3 BLEU Score

For BLEU score, I have calculated detokenized BLEU score. Because, although BLEU score is standard but different tokenization libraries generate different outputs, so to cater for this, I have used detokenized BLEU score. BLEU Score range is from 0-1 with zero being lowest and one being highest but it is mostly scaled by 100 as standard practice.

For Detokenized BLEU score I have used sacreBLEU lib which works on detokenized text (unless the '-force' parameter is used).

### 6.2.4 Corpus BLEU score

For corpus BLEU score calculation, 11370 sentences are used and score is= 57.60388951064197

```
In [132]: runfile('E:/rida/Rida Thesis/ridaThesisNew/
BLEU_Score_calculation.py', wdir='E:/rida/Rida Thesis/ridaThesisNew')
Reference 1st sentence: I am starting to learn German
MTed 1st sentence: i am hoping to tell german
57.60388951064197
```

Figure 6.9: Corpus BLEU score

### 6.2.5 Sentences BLEU score

In corpus BLEU score calculation, BLEU score is calculated for whole dataset but to check accuracy of translation of individual sentences, Sentences BLEU score is calculated. Following are few samples sentences with their respective BLEU Score

- I am starting to learn German —> i am hoping to tell german ((**BLEU Score = 9.652434877402245** )

- I am sure he has other skills → i am sure he has other skills (**BLEU Score = 80.91067115702207** )
- I am sure I have never seen her → i am sure i have never mary her (**BLEU Score =16.515821590069027** )
- I am thinking about your plan → i am thinking about your plan (**BLEU Score = 75.98356856515926** )
- I am thinking of going abroad → i am sorry of going abroad (**BLEU Score = 32.46679154750991** )
- I am tired from the long walk → i am tired from the long with (**BLEU Score =61.47881529512643** )
- I am tired of waiting in line → i am tired of waiting in line (**BLEU Score = 80.91067115702207** )

# Chapter 7

## Conclusion

This is the first time the work has been done on English language tenses conversion. Previously Classifier based statistical models were built for translating tenses that does keep the verb tense structure intact but in doing so accuracy and fluency was lost. Now I have implemented attention based encoder decoder model to perform the task of tense conversion, achieving accuracy of 96%. It can handle long and complex sentences well. Due to high accuracy of tenses conversion results, model can be successfully deployed for autistic children which is the reason I started working on this problem. There is,however, still need for improvement in the architecture I have designed for tense conversion.

### 7.1 Future Work

Most of my efforts in this thesis is in generation of dataset of over 500,000 sentences for tense conversion and training it on multiple NMT architectures.

This work can be further extended to generate datasets to understand different aspects of English language e.g. active and passive voices sentence structures or direct and indirect sentence structures etc. The other field, where the major work can be done in the field of NMT is improvement of architecture to cope with very large paragraphs type sentences and coping up with complex vocabularies. This can be done by using data which is not only of translation but also from other tasks like image caption generation, unsupervised learning etc which is the future of NMT. Another aspect where work can be done in the field of NMT is making of smaller NMT models by compressing them in such a way that their performance is not sacrificed. These models can then be deployed in mobile devices which are becoming dominant in our daily life. Due to that we can use NMT in alot of applications.

# Bibliography

- [1] C. Sunita, “Empirical survey of machine translation tools,” *d International Conference on Research in Computational Intelligence and Communication Networks*, pp. 181–183, 2016.
- [2] A. Yanishevsky, “Neural won! now what?” in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*. Boston, MA: Association for Machine Translation in the Americas, Mar. 2018, pp. 84–112. [Online]. Available: <https://aclanthology.org/W18-1911>
- [3] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [4] B. Liu and I. R. Lane, “Attention-based recurrent neural network models for joint intent detection and slot filling,” *CoRR*, vol. abs/1609.01454, 2016. [Online]. Available: <http://arxiv.org/abs/1609.01454>
- [5] M. A. Farajian, M. Turchi, M. Negri, and M. Federico, “Multi-domain neural machine translation through unsupervised adaptation,” in

- Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 127–137. [Online]. Available: <https://aclanthology.org/W17-4713>
- [6] Z. Gong, M. Zhang, C. Tan, and G. Zhou, “Classifier-based tense model for SMT,” in *Proceedings of COLING 2012: Posters*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 411–420. [Online]. Available: <https://aclanthology.org/C12-2041>
- [7] “Google neural machine translation - wikipedia,” [https://en.wikipedia.org/wiki/Google\\_Neural\\_Machine\\_Translation](https://en.wikipedia.org/wiki/Google_Neural_Machine_Translation).
- [8] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, “A statistical approach to machine translation,” *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990. [Online]. Available: <https://aclanthology.org/J90-2002>
- [9] N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1700–1709.
- [10] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, “Google’s neural machine translation system: Bridging the gap between

- human and machine translation,” *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: <http://arxiv.org/abs/1609.08144>
- [11] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” *CoRR*, vol. abs/1611.04558, 2016. [Online]. Available: <http://arxiv.org/abs/1611.04558>
- [12] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig, “When and why are pre-trained word embeddings useful for neural machine translation?” *CoRR*, vol. abs/1804.06323, 2018. [Online]. Available: <http://arxiv.org/abs/1804.06323>
- [13] P. Koehn and R. Knowles, “Six challenges for neural machine translation,” *CoRR*, vol. abs/1706.03872, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03872>
- [14] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On using very large target vocabulary for neural machine translation,” *CoRR*, vol. abs/1412.2007, 2014. [Online]. Available: <http://arxiv.org/abs/1412.2007>
- [15] G. Tang, M. Müller, A. Rios, and R. Sennrich, “Why self-attention? A targeted evaluation of neural machine translation architectures,” *CoRR*, vol. abs/1808.08946, 2018. [Online]. Available: <http://arxiv.org/abs/1808.08946>
- [16] “Free ebooks | project gutenber,” <https://www.gutenberg.org/>.

- [17] “English sentences focusing on words and their word families,” <http://www.manythings.org/sentences/words/>.
- [18] “Understanding encoder-decoder sequence to sequence model | by simeon kostadinov | towards data science,” <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346>.
- [19] “Attention model in an encoder-decoder | by pragati baheti | heartbeat,” <https://heartbeat.fritz.ai/attention-model-in-an-encoder-decoder-a1ad4ac3cda2>.
- [20] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *CoRR*, vol. abs/1506.07503, 2015. [Online]. Available: <http://arxiv.org/abs/1506.07503>
- [21] “Neural machine translation with attention | text | tensorflow,” [https://www.tensorflow.org/text/tutorials/nmt\\_with\\_attention](https://www.tensorflow.org/text/tutorials/nmt_with_attention).