# Language Translation using RNNs

**Gita Krishna** **Jackson Eilers** **Faraz Abbasi**
Department of Computer Science
Stanford University
{gitakris, jeilers, faraz13}@stanford.edu

## Abstract

This project aims to evaluate several different approaches to neural machine translation, incorporating deep learning techniques such as sequence to sequence models, long short-term memory cells, and self-attention. This paper contains an in-depth analysis of each approach, and an error analysis of hypothesis translations made by the model. The accuracy metrics used here are BLEU scores. The models built perform French to English translation.

## 1 Introduction

The inability to communicate due to language barriers is one of the largest impediments we face as a society today. We can even see the importance is business. For example, HSBC Holdings, a major UK based bank lost 10 million dollars in rebranding costs after their campaign "Assume Nothing" was translated to "Do Nothing" in a number of countries across the world. This translator, in whatever form it took on, lacked the intelligence to identify the context within which the original statement existed. This issue is something we are working to circumvent with our system. However, this issue doesn't just stop at business. It extends to the international relations as well. Take for example aid relief across countries. Between two countries that speak different languages, take Mexico and the United States, the efficiency of communication during a natural disaster is of upmost importance. With every moment passing incredibly detrimental, it is important to have an institution in place that satisfies is more efficient than human translators, that must be contacted and are much slower than a computer. We used Data from the EMNLP 2011 Sixth Workshop on Statistical Machine Translation as our input (courpuses of sentence in both French and English). Then, we used RNNs to learn the translations of these sentences and as an output we try to translate a sentence from French to English or vice versa. Our project consists of comparing different models, namely:

- Model 1: Sequence to Sequence Recurrent Neural Network (RNN) with Adam Optimization, Gradient Clipping, Dropout, and L2 regularization
- Model 2: Sequential Long Short-Term Memory (LSTM) RNN with Nadam Optimization
- Model 3: Seq2seq RNN with LSTM cells and Stochastic Gradient Descent (SGD) as the method of optimization. Model with and without an attention mechanism.

## 2 Related work

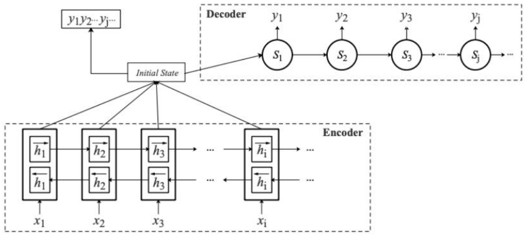
One related paper that we looked at was Park Chansung's Seq2Seq model in TensorFlow which used a sequence to sequence (seq2seq) RNN with gradient clipping, Adam optimization and dropout. One shortfall of this model was the restricted portion of the WSTM 2010 French-English Corpeum while another issue stemmed from the hyperparameters, whose usage we found were sub-optimal. In

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

these ways our model 1 differs, as we tuned hyperparameters, added L2 regularization to bolster the efficiency of training and also added Bleu scoring to better tune and test the model.

The model used in "Sequence to Sequence Learning with Neural Networks" had multi-layered LSTM cells in order to map an input sequence into a fixed dimensional vector. The use of LSTM allows the model to better understand the context of the input. This model is most similar to our model 2, however it differs most starkly with this technique in LSTM training. There is similar related work in the paper "Effective Approaches to Attention-based Neural Machine Translation".

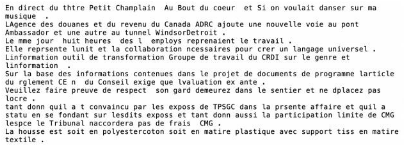
Another model (see image below) from "Neural Machine Translation with Word Predictions" uses an encoder-decoder structure similar to our Models 2 and 3. This paper uses a word prediction mechanism, wherein the decoder is able to predict individual words in a target sentence as part of the learning objective for the initial state. For this, this model incorporates a gated recurrent unit (GRU), which has been shown to perform well in speech recognition and machine translation.



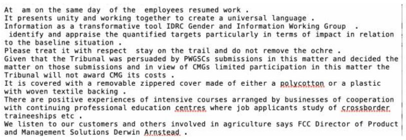
The model described in "A novel approach to neural machine translation" used CNNs, in contrast to all our models which used RNNs. As described in their paper, CNN's have two main advantages bring its ability to compute elements simultaneously and that they process information hierarchically, allowing it to capture complex relations in data. While in research RNN usually outperform CNN's, the CNN's ability to capture more complexity gives it the unique ability to better translate across more grammatically nuanced languages.

## 3 Dataset and Features

We are using Data from the EMNLP 2011 Sixth Workshop on Statistical Machine Translation, which focuses on translation between various European languages. Large portions of this data come from the European Parliament Proceedings Parallel Corpus, which contains sentence aligned text for translation systems in 21 languages. This dataset also includes data from the News Commentary corpus. Overall, the dataset contains about 45 million words of training data per language from Europarl, and 2 million words per language from News Commentary. This dataset is extremely large, and we are using a 98-1-1 split for our train, development, and test sets. While building the model, we intend to use a fraction of the dataset for implementation convenience (around 150,000 sentences in our training set and 1,500 for both our dev and test set). Below are examples of the French and English training data, respectively.



```
En direct du thtre Petit Champlain  Au Bout du coeur  et Si on voulait danser sur ma
musique  .
LAgence des douanes et du revenu du Canada ADRC ajoute une nouvelle voie au pont
Ambassador et une autre au tunnel WindsorDetroit .
Le mme jour  huit heures  des l  employs reprenaient le travail .
Elle reprsente lunit et la collaboration ncessaires pour crer un langage universel .
Linformation outil de transformation Groupe de travail du CRDI sur le genre et
linformation  .
Sur la base des informations contenues dans le projet de documents de programme larticle
du rglement CE n  du Conseil exige que lvaluation ex ante .
Veuillez faire preuve de respect  son gard demeurez dans le sentier et ne dplacez pas
locre .
tant donn quil a t convaincu par les exposs de TPSGC dans la prsente affaire et quil a
statu en se fondant sur lesdits exposs et tant donn aussi la participation limite de CMG
lespce le Tribunal naccordera pas de frais  CMG .
La housse est soit en polyestercoton soit en matire plastique avec support tiss en matire
textile .
```



```
At  am on the same day  of the  employees resumed work .
It presents unity and working together to create a universal language .
Information as a transformative tool IDRC Gender and Information Working Group  .
 identify and appraise the quantified targets particularly in terms of impact in relation
to the baseline situation .
Please treat it with respect  stay on the trail and do not remove the ochre .
Given that the Tribunal was persuaded by PWGSCs submissions in this matter and decided the
matter on those submissions and in view of CMGs limited participation in this matter the
Tribunal will not award CMG its costs .
It is covered with a removable zippered cover made of either a polycotton or a plastic
with woven textile backing .
There are positive experiences of intensive courses arranged by businesses of cooperation
with continuing professional education centres where job applicants study of crossborder
traineeships etc .
We listen to our customers and others involved in agriculture says FCC Director of Product
and Management Solutions Derwin Arnstead .
```

Preprocessing done on the data varied based on the needs of the model. In general, we used a small fraction of the data in order to run our model in a practical amount of time. Additionally, the dataset contained lots of odd symbols that were not part of the french or english vocabulary, so filtering needed to be done based on that to ensure that the text we were attempting to translate was actually part of the vocabulary. We stored our source and target vocabularies in numpy ndarrays, and used these words as features. A citation of where we got our data is included in the References section under label 4.

# 4 Methods

**Model 1**: For this model we used a sequence to sequence RNN with adam optimization, gradient clipping, dropout and L2 regularization using TensorFlow. Much of this code follows Chansung Park's work in his "English-French Machine Language Translation in Tensorflow" project. Adam optimization allows the model to train more efficiently. We used gradient clipping to solve the issue of RNN's vanishing and exploding gradient. The idea is that gradients cannot go outside of the range of [-1,1], which makes training more efficient for models. We also employed dropout. Dropout effectively greys out certain neurons during different phases of training. This strengthens the overall model by lessening the likelihood of overfitting. Finally, we used L2 regularization which penalizes large weights, in turn lessening the likelihood of having very large gradients, making the training more efficient. The loss function used in this model was weighted softmax cross entropy loss function. An equation for this function is illustated below along with an equation for L2 regularization, repectively:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \quad - \ ||w||^2$$

**Model 2**: For our second model, we adapted our model from a sequential Encoder-Decoder LSTM model, cited below ([6]). Since the source language and target language were different than what this model was originally used for, we had to make several modifications, but retained aspects of the general structure. We trained Keras' inbuilt Tokenizer on our french and english vocabularies, and then built our encoder using these tokenized values. We post-padded these sequences, and used Keras' inbuilt Embedding layer to embed our source vocabulary. We then incorporated an LSTM cell that returned the last output in the output sequence, as well as a Time Distributed layer wrapper to our output Dense layer with an activation function of 'softmax'. Regarding our optimizers, both Adam and Nadam worked fairly well, but Nadam slightly outperformed Adam, with the key difference between the two being that Nadam uses Nesterov momentum. We used categorical cross entropy loss, with the formula as follows:

$$CCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1y_i \in C_c log_{p_model}[y_i \in C_c]$$

**Model 3:** Our third model was adapted from a seq2seq model Luong et al.'s model with our data to compare the benefits of using a sequential RNN with and without an attention mechanism. We had to do a lot of data modification in order to get this model to work with our data. This included creating our own vocabularies of the 17,000 most common words in the English and French language. This model uses a bi-directional encoder and decoder to create embeddings for the English and French vocabularies. This algorithm creates embeddings for both of the vocabularies of the source and the target language. The encoder has access to the word embeddings and the source information. The decoder also needs access to the source information, the word embeddings, and the target information. We pass the last hidden state of the encoder to the decoder to allow the decoder access to the source information. We measure the success of the translations using the Bleu score, as shown below where $c$ represents the actual number of word matches in order and $r$ represents the total possible word matches by order per input/output pair.

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

Then,

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

We also had another model with an attention mechanism. For shorter sentences, there is not as much context and thus translations are easier. However, with longer sentences, there can be context that is missed and therefore leads to worse translations. It does this by comparing the current word to be decoded with each word of the input to create an attention vector. It then creates a context vector of the weighted averages of the source words which is used to help decode each of the target words in the hidden states, as shown below.

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \qquad \text{[Attention weights]} \tag{1}$$

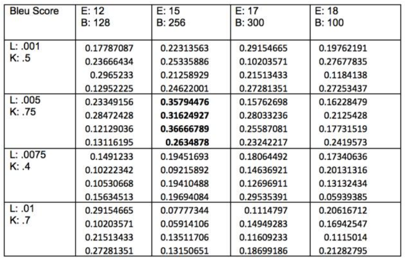
$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \qquad \text{[Context vector]} \tag{2}$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \qquad \text{[Attention vector]} \tag{3}$$

## 5 Experiments and Results

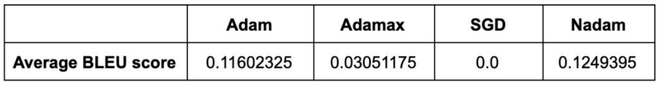
For most of the analysis of our model's performance, we used BLEU scores as our overall metric, the most common evaluation metric used for translation. A BLEU score of 1.0 indicates an identical output to the reference translation, and a score of 0.0 indicates no n-gram overlap between the reference translation and the hypothesis translation. BLEU scores take into account accuracy at various ngrams, and interpolate these values to come up with an overall accuracy metric.

**Model 1**: In this model we chose the following hyperparamters: Epochs = 15, batch size = 256, learning rate = .005, keep probability = .75. Originally, our data was severely overfitting our training data. In order to circumvent this process, we introduced dropout, with the hyperparameter .75 for the keep probability for any arbitrary neuron. For all of these parameters, we randomly sampled different combinations and re-ran our training algorithm using these unique initializations. We made sure to not make the steps uniform, in order to maximize the number of values we tried for each value. We then chose the combination that gave us the highest rating for Bleu scores, averaging the 1-gram, 2-gram, 3-gram and 4-gram scores. A table below shows the outputs from these various models. (Legend: **E**: epochs, **B**: batch size, **L**: learn rate, **K**: keep rate for dropout)



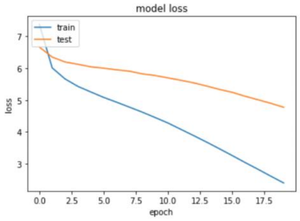
| Bleu Score | E: 12<br>B: 128 | E: 15<br>B: 256 | E: 17<br>B: 300 | E: 18<br>B: 100 |
|---|---|---|---|---|
| L: .001<br>K: .5 | 0.17787087<br>0.23666434<br>0.2965233<br>0.12952225 | 0.22313563<br>0.25335886<br>0.21258929<br>0.24622001 | 0.29154665<br>0.10203571<br>0.21513433<br>0.27281351 | 0.19762191<br>0.27677835<br>0.1184138<br>0.27253437 |
| L: .005<br>K: .75 | 0.23349156<br>0.28472428<br>0.12129036<br>0.13116195 | **0.35794476**<br>**0.31624927**<br>**0.36666789**<br>**0.2634878** | 0.15762698<br>0.28033236<br>0.25587081<br>0.23242217 | 0.16228479<br>0.2125428<br>0.17731519<br>0.2419573 |
| L: .0075<br>K: .4 | 0.1491233<br>0.10222342<br>0.10530668<br>0.15634513 | 0.19451693<br>0.09215892<br>0.19410488<br>0.19694084 | 0.18064492<br>0.14636921<br>0.12696911<br>0.29535391 | 0.17340636<br>0.20131316<br>0.13132434<br>0.05939385 |
| L: .01<br>K: .7 | 0.29154665<br>0.10203571<br>0.21513433<br>0.27281351 | 0.07777344<br>0.05914106<br>0.13511706<br>0.13150651 | 0.1114797<br>0.14949283<br>0.11609233<br>0.18699186 | 0.20616712<br>0.16942547<br>0.1115014<br>0.21282795 |

**Model 2**: For this model, we experimented with several different architectures, as well as various forms of hyperparamater tuning. We also attempted to use a french stemmer and an english porter stemmer, to remove morphological affixes from words and use only the word stem. We received mixed results from our model with the stemmer, where it performed perfectly on certain sentences and very poorly on others. We experimented with several optimizers, and based on our BLEU scores, decided to use Nadam. Below is a table containing our BLEU scores with various optimizers.

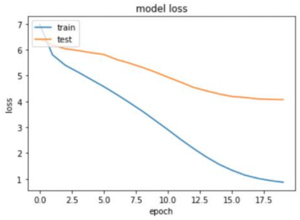


| | Adam | Adamax | SGD | Nadam |
|---|---|---|---|---|
| **Average BLEU score** | 0.11602325 | 0.03051175 | 0.0 | 0.1249395 |

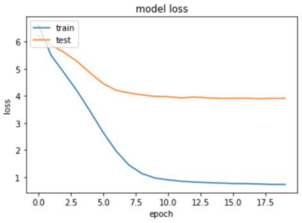
For our Nadam optimizer, we experimented with several different ranges of learning rates. Below are graphs that represent our models validation loss over 20 epochs for various learning rates.



Learning Rate = 0.001

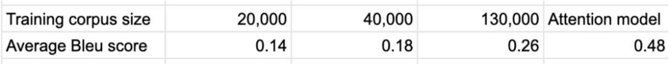


Learning Rate = 0.002



Learning Rate = 0.005

**Model 3:** For this model we had 128 dimensional hidden units. We used dropout (rate of 0.8) and SGD with a learning rate of 1 as our method of optimization. After 8 epochs we begin to decay our learning rate by half for the remaining 4 epochs (12 epochs total). Batch size of 256. We chose these based on the paper of Luong et al. based on their model. After training this model on the original dataset, cleaning the data and rerunning led to better results. We also trained it on different sizes of data whose results are in the table below. The attention model used corpus of 130,000 examples.



| Training corpus size | 20,000 | 40,000 | 130,000 | Attention model |
|---|---|---|---|---|
| Average Bleu score | 0.14 | 0.18 | 0.26 | 0.48 |

## 6 Error Analysis

**Model 1**: In the below example, we can see that the two words that were dropped out were "threedimensional" and "gridbased." There are two explanations for this phenomena. The first is that these are both very uncommon words and the model is not likely to understand the meaning behind these words. The other reason is both of these words are most commonly used with hyphens or spaces in between. However, given the strange nature in which they were written, the model would have a hard time coming to this translation. Without their meanings, the sentiment was lost and replaced with a simple "a.' This strange formatting of data was present in much of our dataset and contributed to much of the error we saw. **Original English:** Their simplest form is the box model, but the most Eulerian models are three-dimensional gridbased models. **Actual Translation:** Leur forme la plus simple est le modele en boite, mais la plupart des modèles euleriens sont des modèles tridimensionnels bases sur une grille. **Model Translation:** Leur forme la plus simple est le modele de boite, mais la plupart des modèles sont des modèles.

**Model 2:** This model had trouble translating sentences with more than 5-6 tokens, and generally performed better on short words or phrases. It seemed that the model was having trouble stringing together larger sequences, but was able to translate words in isolation. One example where the model was unable to retain sequential knowledge is as follows: **Source**: Merci, Papa!, **Target translation:** Thanks, Dad!, **Hypothesis translation**: thanks thanks thanks thanks thanks. Here, we see that the model is able to translate the word 'Merci' corectly, but fails to recognize the sequential information necessary to create a coherent phrase, even though the source text is short. This likely has to due with the fact that there is only one sequential layer in this model.

**Model 3:** This model mostly failed when there were words that were not in the dictionary (proper nouns). For example, one sentence read "M Sherb ntait pas partie la plainte qui avait ..." and it was translated to "Mr <unk> was not part of the complaint ...". Since this was trained with the attention model, most of the other words were translated correctly and even thought it was a long sentence, it had a Bleu score of approximately 0.5.

## 7 Conclusion/Future Work

Overall, our first model was the highest performing on average due to the combination of using an Adam optimizer and L2 normalization. However, the highest BLEU score came from model 3 which used the attention mechanism. Given the long sentences in the corpus, the attention mechanism allowed the model to keep track of the context within the sentence to choose the best words for the translation. Since this model used SGD normally, we concluded that this is the reason why it did not perform as well as model 1 and that Adam was a preferable optimizer. If we had more time, we would have loved to explore translating other corpus' on the same models to see if it was a limitation of the data. Further, we would have loved to test all of the different elements of all the models to find some combination that was most effective at translating languages.

## 8 Contributions

Project Proposal: Gita, Jackson, Faraz
Project Milestone: Gita, Jackson, Faraz
Final Project: Gita, Jackson, Faraz
Model 1: Faraz
Model 2: Gita
Model 3/4: Jackson

Project Poster: Gita, Jackson, Faraz
Project Presentation: Gita, Jackson

## References

[1] Chansung, Park. *Seq2Seq model in TensorFlow*, https://towardsdatascience.com/seq2seq-model-in-tensorflow-ec0c557e560f.

[2] Sutskever, Ilya, Vinyals, Oriol. & Quoc V. Le (2014) *Sequence to Sequence Learning with Neural Networks System.* https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

[3] Gehring, Jonas & Michael Auli (2017) *A novel approach to neural machine translation.* https://code.fb.com/ml-applications/a-novel-approach-to-neural-machine-translation/.

[4] EMNLP 2011 SIXTH WORKSHOP ON STATISTICAL MACHINE TRANSLATION (2011). http://statmt.org/wmt11/translation-task.html#download

[5] Effective approaches to attention-based neural machine translation, Minh-Thang Luong, Hieu Pham, and Christopher D Manning. EMNLP, 2015.

[6] https://github.com/vibhor98/Neural-Machine-Translation-using-Keras

[7] https://github.com/tensorflow/nmt