Recurrent Deep Learning Applied to Written Slang (GenZ) to Produce Standard English Translations

Sabina Miani & Gabriel Job

Final Project Report - Deep Learning CS6353/5353
Fall 2022

Introduction

Problem

Every generation of children find new ways to communicate with language using slang, text shorthand, and emojis. Slang is used in almost every setting and across cultures and other languages. English slang is some of the most commonly used around the world. The various usages of slang can make it hard to understand the intention and meaning behind the words, especially for those unfamiliar with certain slang. This holds true for the current generation of emerging slang (GenZ slang). Slang is more prevalent now than in any previous generation. Thus, for those unfamiliar with current slang, this project hopes to find a way to translate modern slang into reasonable English translations.

Dataset

To predict correct translations, a dataset with slang used in a variety of settings and contexts was needed. This was difficult to find as there are few datasets with this criteria available for public use that are large enough to contain enough contextual data to be useful when training a deep learning model. In order to remedy this problem, we found a dataset that utilized a large amount of basic English sentences originally meant for translation to French from a github user, Romel Torres. The next step was duplicating the English version and replacing key verbs and nouns with modern slang language including developing slang found on popular social media sites like Facebook and Twitter.

The dataset is ~138k sentences, making use of common topics such as fruits, locations, and weather conditions. We chose to randomly select ~80k sentences for use in the training of the model and ~58k sentences to use in the validation set; this number was chosen arbitrarily, and more research and experimentation on splitting the dataset may prove advantageous to produce better results. This dataset is focused on the various ways a few slang phrases can be implemented in a sentence. Below are example inputs from the dataset with GenZ slang bolded.

the **us** is usually a **lil** cold in july , and its usually **freezin** in **nov** .
cali is usually quiet in march , and its **fr** always hot in june .
the **us** is **kinda mid** in june , and its cold in **sept** .

Text 1. Example input sentences from the used dataset. The slang words are in bold.

These can be translated to mean the following:

the united states is usually chilly during july , and it is usually freezing in november .
california is usually quiet during march , and it is usually hot in june .
the united states is sometimes mild during june , and it is cold in september .

Text 2. Standard English translation of *Text 1* examples from the used output dataset.

The slang in the above examples include abbreviations and acronyms for standard and slang phrases. Names of places and months are shortened like ‘cali’ is ‘california’ and ‘us’ is ‘united states’. There are phrases more closely associated with GenZ slang such as ‘fr’ which stands for ‘for real’ and is commonly used as a confirming question or statement of truth depending on context. In the above example, it is used as a statement of affirmation.

This dataset was kept in mind while creating a deep learning model to translate GenZ slang.

BLEU

Bilingual Evaluation Understudy or more commonly known as BLEU is a known algorithm to evaluate machine-translated text. Translated sentences are scored between zero and one, where a score closer to one represents a better translation. Calculating a BLEU score relies on the Brevity Penalty and N-Gram Overlap. N-Gram Overlap is calculated using a modified n-gram precision function to count the number of uni-grams, bi-grams, tri-grams, etc., the standard being 4-grams, using uniform weights. The version of BLEU we used is a function in the NLTK library which is used to calculate an individual sentences’ BLEU score. After calculating each sentences’ BLEU score, the average of the validation set sentence scores is calculated. This will serve as the final accuracy testing of the created deep learning model for slang translation.

Research

GRU

Translating generational language into more standard English is best accomplished using a recurrent deep learning model as a natural language processor (NLP) to analyze text sequentially. Doing some research into recurrent models, a highly effective model known as the Long Short Term Memory (LSTM) model takes a long time to compute, but it has a similar, yet less computationally heavy equivalent known as the gated recurrent unit (GRU). GRU was proposed by Cho et al. in 2014, and is in a lot of ways similar to LSTM, however, one of the major differences between the two is that GRU has 2 gates, an update gate and a reset gate, whereas LSTM has 3 gates: update, reset, and forget. One thing discovered while researching GRUs, was that GRUs tend to yield results more rapidly, which would drastically reduce the training time (see *Figure 1*).

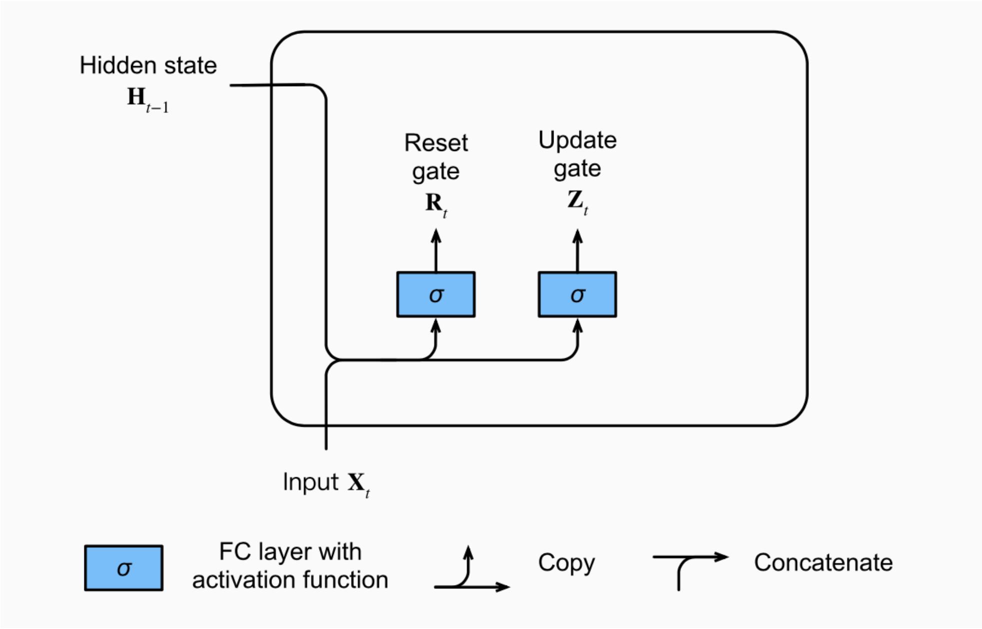

Figure 1. Gated Recurrent Unit GRU functional diagram. Labeled gates corresponding to sigmoid activation over fully connected layers.

This type of model works well when sentences are structured in a linear forward fashion; however, for more complex sentences, it does not perform as well. To help overcome this issue, further research into bi-directional models was conducted.

BiGRU

When using an RNN, there is a singular flow of context to produce an output. The forward layer would shuttle a sentence and produce an output using the singular input context. Bi-directional RNNs build upon this idea, providing additional information by feeding the input backwards, so a sentence such as, “take me out to dinner.” would be perceived by the unit as “dinner to out me take.” (see *Figure 2*).

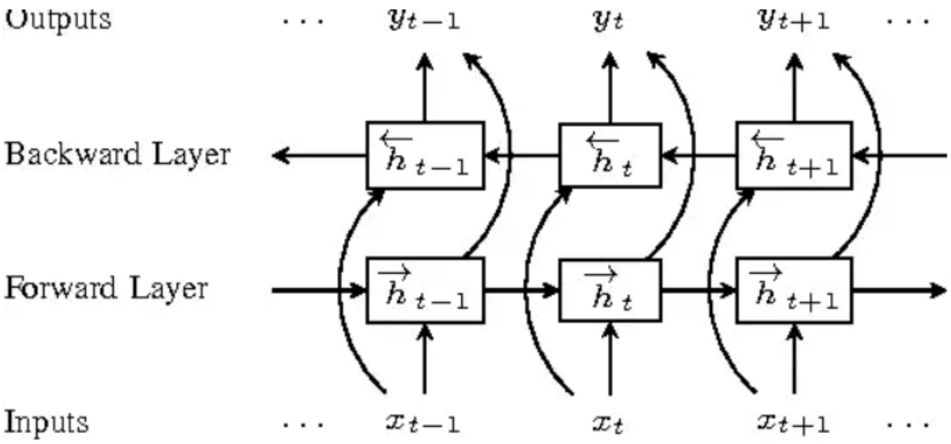

Figure 2. Explicit bi-directional RNN formation with labeled inputs (*x*), hypothesis functions (*h*), and outputs (*y*) through a forward layer followed by a backward layer.

The forward and backward layers are often combined for expanded functionality and comprehension of complex data. This concept is often applied to LSTM and GRU models. When specifically applied to a GRU, called a bi-directional GRU, allows for words and phrases from the sentence to be remembered both forwards and backwards, thus allowing for more complex sentence structures (see *Figure 3*).

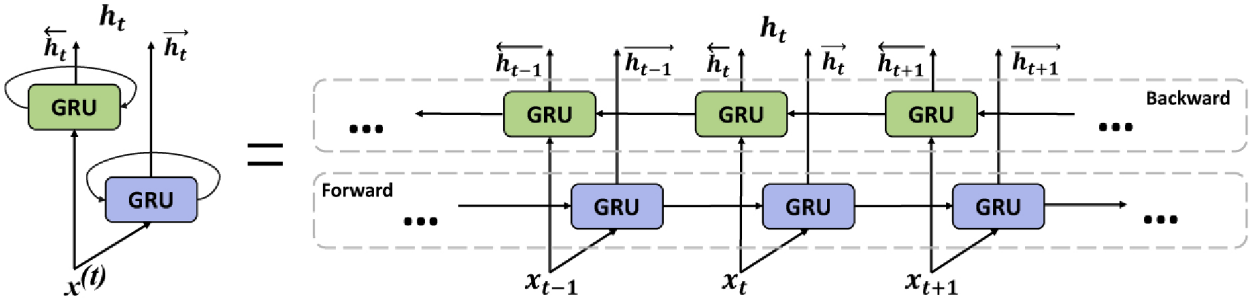

Figure 3. Bi-Directional Gated Recurrent Unit BiGRU structural diagram. Illustrates the forward and backward passes through GRU cells with respect to the inputs and resulting hypothesis function.

Studying further methods for improving RNN accuracy to create a robust model revealed the benefits of using leaky ReLU, batch normalization, dropout, embedding, and attention. As discussed in lecture, some of those methods are familiar and are good ways to improve accuracy. A few new methods, including attention, were discovered and implemented.

Attention Mechanisms

Attention within the context of RNNs describes the attention to context the algorithm gives to particular parts of an analyzed sentence. For example, a classic NLP model could most likely easily handle the translation of this sentence, “I want to go to the store to buy dinner.”, to another language. However, the relevance may be lost resulting in a sentence like, “I want to go to a beach to buy dinner.” While this shows relatively high accuracy, the original context is altered from buying food at a store to buying food at a beach.

An attention mechanism is especially beneficial when analyzing fixed length inputs, namely longer strings. There are multiple types of attention such as, generalized, self-attention, multi-head, additive, global, and local. These types of attention are used for various methods of sentence and word detection mostly altered by function they use in their computation pattern. As seen in *Figure 4*, the attention function is a series of alignment, special function, and then a summation. The visual illustrates a softmax attention function, also known as a location based attention cell because it specializes in learning text placement.

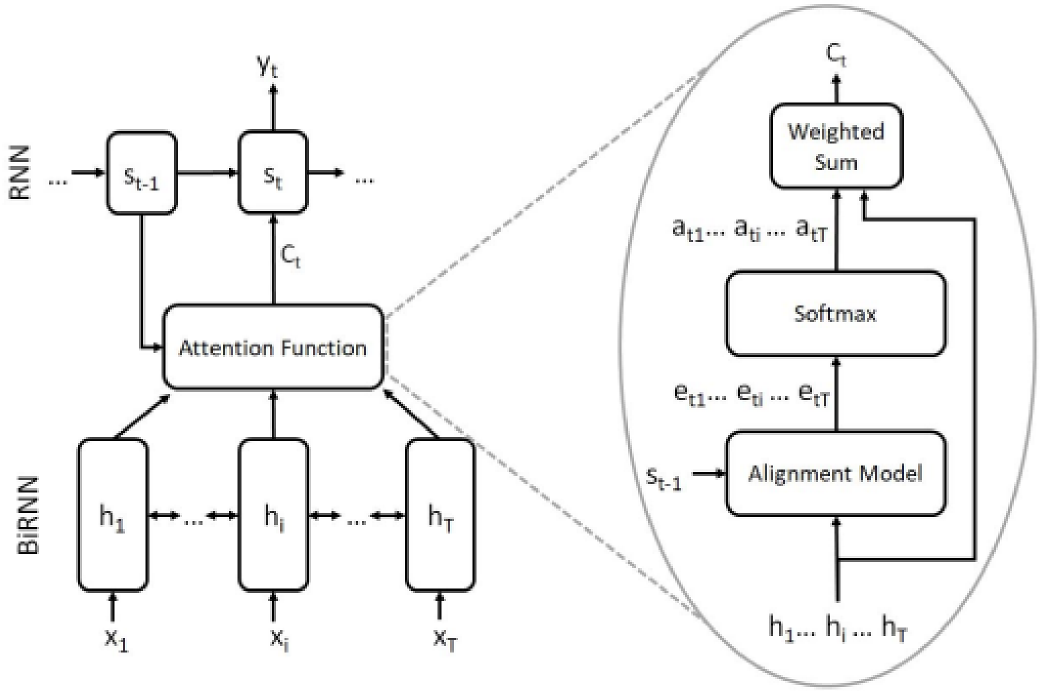

Figure 4. Showing an RNN utilizing a location based attention function (softmax). This operates by supplying the attention function with hidden states and embedded input. Which is then pushed through a softmax function, before yielding weights to be applied to the original input.

Methods

We started with a basic learning model that implemented text translation via the use of a standard RNN with embedding, but after further research opted to use separate encoder and decoder structures. At first, our architecture using separate encoder and decoder was making use of single-directional RNN's for both encoding and decoding structure (see *Figure 5*).

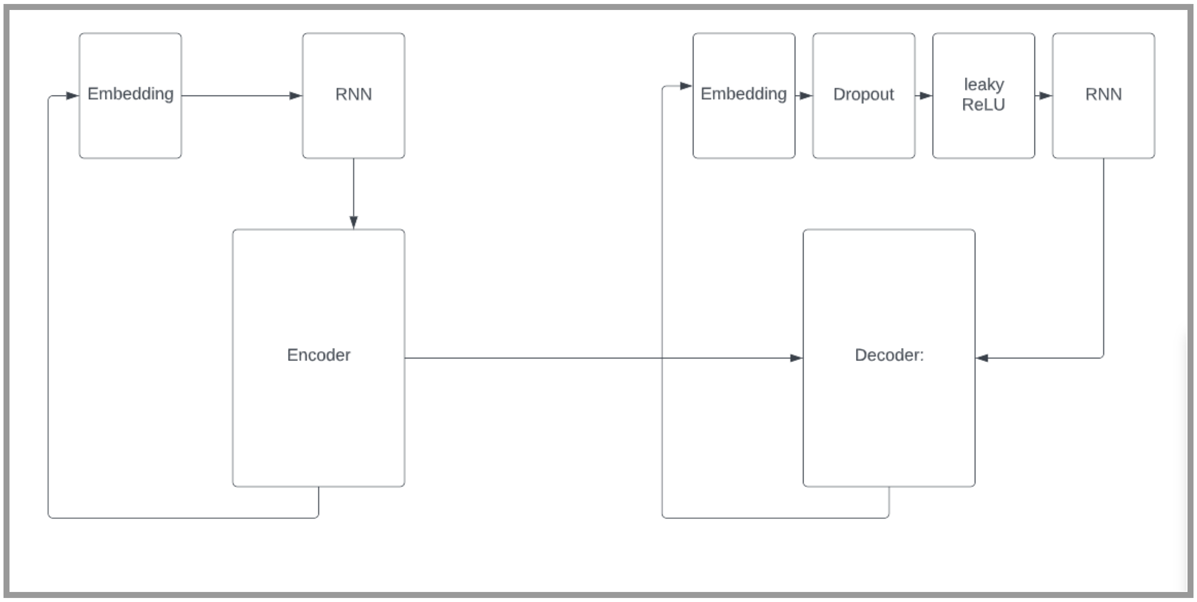

Figure 5. Our first implementation of separate encoder and decoder structures. Supporting elements for each are labeled and connected to the coders.

The general architecture uses a leaky ReLU activation function, chosen in place of the normal ReLU to aid with vanishing gradients, and a dropout layer to help prevent overfitting. This design did not yield desirable results, likely due to human error, with unanticipated output even after 50,000 training iterations shown below with GenZ slang bolded.

```
expected: the lemon is their fav fruit , but the yellow glizzy
is our fav .

got: . . . . in in . . , . . . . ., , , , , , , , , , , , in . .
, . . . . ., , , , . , , . , , ,
```

Text 3. Output of the initial RNN model with separate encoder and decoder structures after 40k iterations. The expected slang words are bolded.

This shows lots of room for improvement for the next attempted architecture. We attempted to implement a modified version of an RNN, a Gated Recurrent Unit or GRU (see above research sections GRU and BiGRU). After experimenting with different layering variations including GRU, the resulting architecture as shown in *Figure 6* was created.

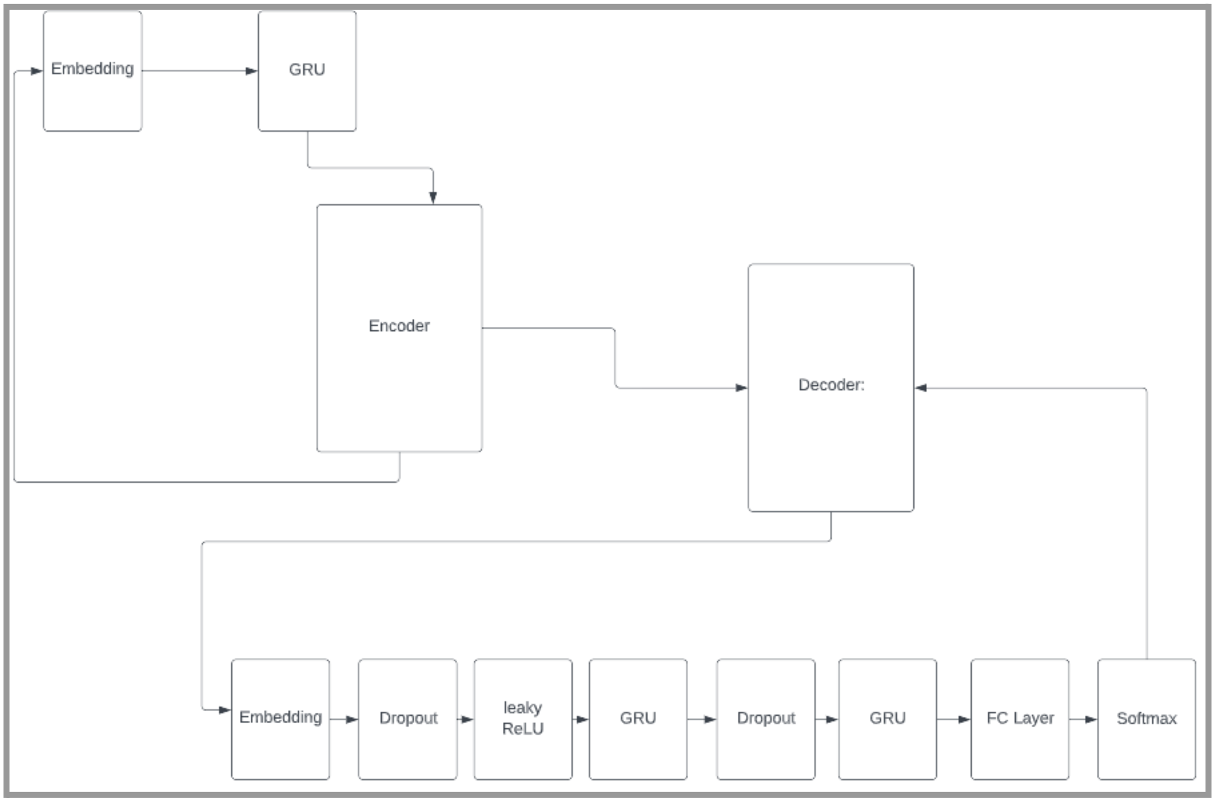

Figure 6. Architecture utilizing separate encoder and decoder. GRU cells were added to both encoder and decoder with a single cell in the encoder and two in the decoder.

In this architecture, we can see elements from the previous architecture that remain common across many different NLP architectures unique to the separate encoder/decoder model. One of the key features being the fully connected layer before a softmax layer with ReLU as the activation function. One of the major differences in this updated architecture is the stacked GRU setup for each decoding call. Between each GRU cell, we added a dropout layer to prevent overfitting. This architecture worked significantly better than the first one. The output included real English words structured to create sentences and not just punctuation. However, after 40,000 training iterations, this type of output was not consistent (see example output below).

```
(1)
expected:  the lemon is their fav fruit , but the yellow glizzy
is our fav .
===> got:  jersey is kinda dope in fall , but its usually warm
in fall .

(2)
```

```
expected:  jersey is kinda freezin in may , but it aint nice in
winter .
===>  got:  the is is in in , , and its aint in in . .
```

Text 4. Output from GRU model using separate encoder and decoder structures with unique decoder layers after 40k iterations. The expected slang words are bolded.

This architecture forms basic sentences when provided specific inputs, although, not the correct sentences. Even though example (2) is mainly transition words and incorrect punctuation, it did receive the slang term 'aint' in approximately the correct location. This signifies that the translator is working, but the model is failing to understand the main context of each sentence and phrase.

BiGRU With Attention

The results in the previous architecture eventually led us to try using two mechanisms, attention and bi-directionally enabled GRUs (described above in the research section). Using pytorch as the implementation method, the constructed architecture is drawn below.

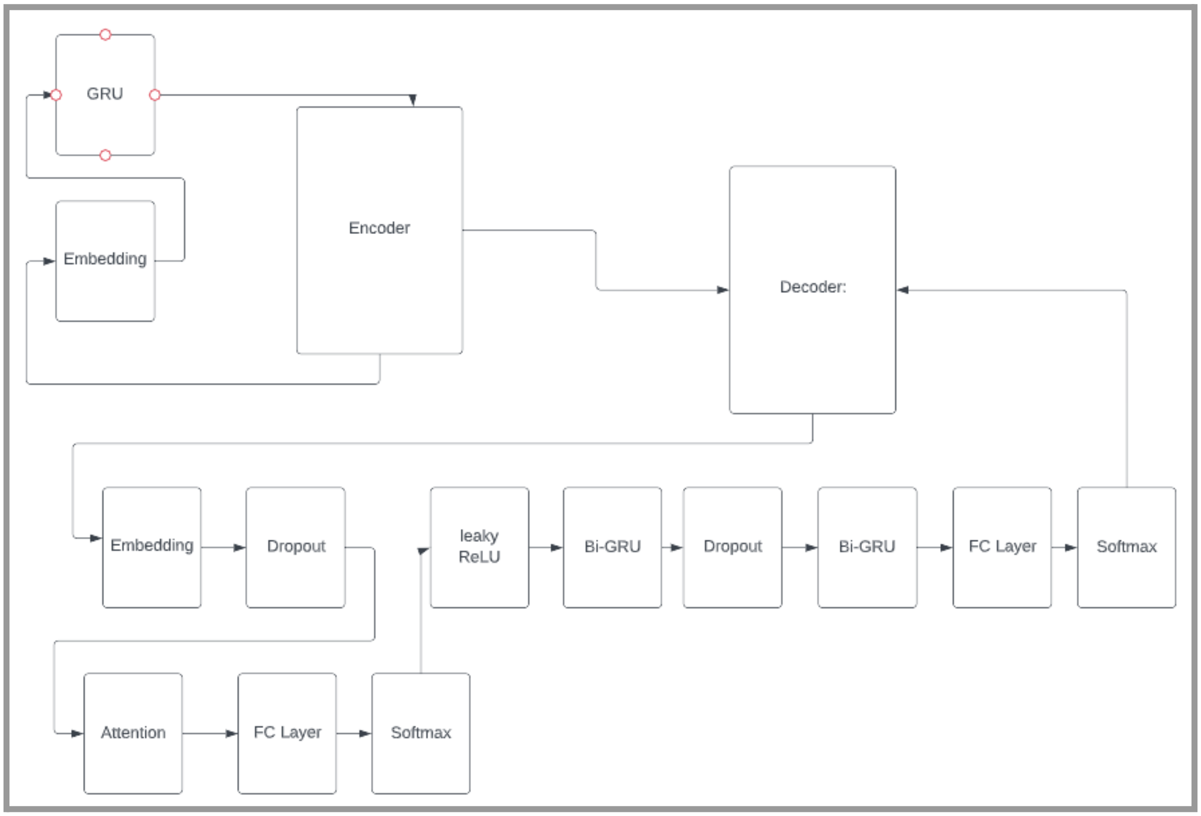

Figure 7. Architecture designed with an attention layer and two layers of BiGRU in the decoder. The encoder is similar to previous architectures.

In this model, the use of an attention mechanism, fully connected layers, softmax function, and BiGRU cells result in considerably more accurate output. An unexpected feature of this architecture was that it required dramatically fewer training iterations, needing only 14,000 with the use of an adagrad optimizer set with a learning rate of 0.1. The loss computed during training can be displayed with respect to the iterations in a graph shown in *Figure 8*. This graph shows the decrease in loss as the model improves its accuracy in producing slang translations given a sentence.

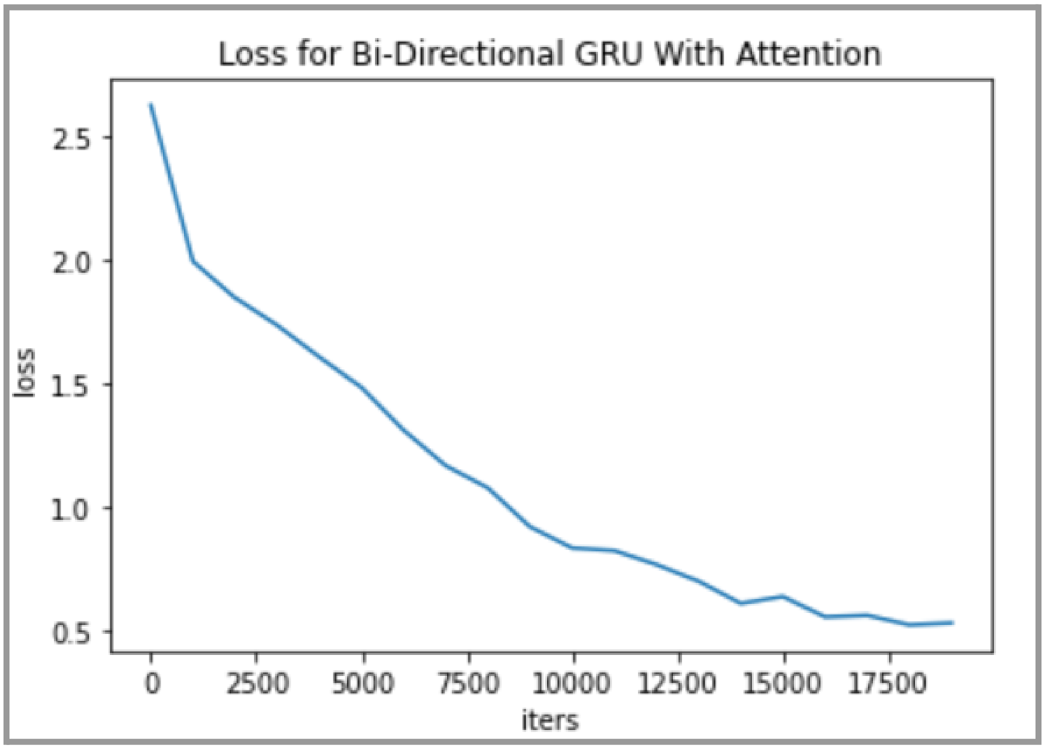

Figure 8. Loss over the number of iterations while a BiGRU model including attention is trained (reference *Figure 7* for a detailed model architecture).

As shown in *Figure 8*, the loss decreases exponentially, a large improvement over the previous model. The loss begins to plateau around 17k iterations.

```
(1)
expected:  cali aint slayin in october , and it aint snowy in
fall .
===> got:  cali aint slayin in march , and it aint snowy in
fall.

(2)
expected:  she fw peaches , lemons , and oranges .
===> got:  she fw fw peaches lemons , and oranges .
```

```
(3)
expected:  the us is dope in nov , but its kinda quiet in aug .
===> got:  the us is dope in aug , but its kinda quiet in aug .
```

Text 5. Output from final model, BiGRU with attention after 20k iterations. The red highlighted text is the incorrect translations. The bolded text is the correct slang translation. Examples (1) and (3) contain incorrect translations, whereas in example (2), it contains a duplicate of the correct translation.

These results show a clear improvement over the previous architecture as consistently coherent English sentences are produced and are translated with high accuracy despite some minor errors. The majority of the errors seem to be the mismatch of months, which may in part be due to the amount of times each noun is used in a variety of sentences throughout the dataset, but more research would need to be conducted to confirm this theory.

To get a numerical score for the final model, the BLEU algorithm was applied. Using an average of over 7800 choices from the validation set of sentence BLEU scores, our model scored approximately 51% on average. This implies our model was able to produce reasonably decent translations. This score seems higher than average. While we don't know the direct cause of this, it may be due to two factors including the improper dataset split between our training and validation sets meaning the model may be overfitting the data or averaging BLEU sentence scores is less accurate than a full corpus BLEU. These issues would require research to remedy.

Improvements

We saw good progress made in the attempt to solve our original problem by translating GenZ slang into standard English using a deep learning algorithm. That said, there is still room for improvement, specifically, in our algorithm architecture. As we decided to create our own architecture using known structures, most of our work was founded by happenstance. If we were to conduct further research into full translation model architectures and understand better how each layer connects and transforms the data, then we would likely have a superior architecture model to the model we are currently working with.

According to a blog post by Ketan Dosha, the highest BLEU scores tend to be in the 60-70% range. Knowing this, the BLEU score from our model was higher than expected. To further improve this score, we can try adopting some of the methods used by the research groups who produced the highest BLEU scores.

References

(1) Sam Wenke, Jim Fleming, “Contextual Recurrent Neural Networks”, https://arxiv.org/pdf/1902.03455.pdf

(2) Kyunghyun Cho, Bart van M., Dzmitry B., Fethi B., Yoshua Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”, https://arxiv.org/pdf/1406.1078.pdf

(3) Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation”, https://aclanthology.org/P02-1040.pdf

(4) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b#:~:text=Bleu%20Scores%20are%20between%200,rarely%20achieve%20a%20perfect%20match, https://arxiv.org/pdf/1412.3555.pdf

(5) Andrea Galassi, Marco Lippi, Paolo Torroni, “Attention in Natural Language Processing”. https://arxiv.org/ftp/arxiv/papers/1902/1902.02181.pdf#:~:text=The%20attention%20mechanism%20is%20a,to%20its%20higher%20level%20representation.

(6) Dichao Hu, “An Introductory Survey on Attention Mechanisms in NLP Problems”, https://arxiv.org/pdf/1811.05544.pdf

(7) A Gentle Introduction to Calculating the BLEU Score for Text in Python - MachineLearningMastery.com

(8) NLP From Scratch: Translation with a Sequence to Sequence Network and Attention — PyTorch Tutorials 1.13.0+cu117 documentation

(9) Two minutes NLP — Visualizing Global vs Local Attention | by Fabio Chiusano | NLPlanet | Medium.

(10) Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU and LSTM RNN with Python and Theano · Denny's Blog

Figure 1: 10.2. Gated Recurrent Units (GRU) — Dive into Deep Learning 1.0.0-beta0 documentation

Figure 2: https://medium.com/analytics-vidhya/bi-directional-rnn-basics-of-lstm-and-gru-e114aa4779bb

Figure 3: Xinyu Liu, Yongjun Wang, Xishuo Wang, Hui Xu, Chao Li, and Xiangjun Xin, "Bi-directional gated recurrent unit neural network based nonlinear equalizer for coherent optical communication system," Opt. Express 29, 5923-5933 (2021)

Figure 4: Attention in Natural Language Processing (5)

Figures 5-8 created by Gabe Job

Text 1-2: DLND-language-translation, a project by Romel Torres (https://github.com/RomelTorres/DLND-language-translation/blob/master/data/small_vocab_en) and edited by Gabe Job (https://github.com/gabejob/deeplearningslang)

Text 3-5 created as a product of our code