# Overview and Problem Statement

Natural language processing is a field of computing and artificial intelligence that bridges the gap between natural human language and computer input. In the past 5-10 years, natural language has made significant strides, largely due to the growing availability of data and computing power required to provide it. Deep learning has played and specially important role in this transformation. Whereas past approaches to natural language processing made explicit attempts to codify grammar and syntax, deep learning allows computers to identify linguistic patterns on its own, much as a would pick on different idioms by listening to parents or older siblings speak.

This view is, of course, not without it's detractors. Famed linguist Noam Chomsky has been notably skeptical of statistical based methods for language learning. At MITs 150th birthday Brains, Minds,and Machines Symposium (http://languagelog.ldc.upenn.edu/myl/PinkerChomskyMIT.html), he remarked frankly:

> *if you uh uh took a ton of video tapes of what's happening outside my office window, let's say, you know, leaves flying and various things, and you did an extensive analysis of them, uh you would get some kind of prediction of what's likely to happen next, certainly way better than anybody in the physics department could do.*
> *Well that's a notion of success which is I think novel, I don't know of anything like it in the history of science. Uh and in- in those terms you get some kind of successes, and if you look at the literature in the field, a lot of these papers are listed as successes. And when you look at them carefully, they're successes in this particular sense, and not the sense that science has ever been interested in.*
> *But it does give you ways of approximating unanalyzed data, you know analysis of ((a)) corpus and so on and so forth.I don't know of any other cases, frankly.*
> *Uh so there are successes where things are integrated with some of the properties of language, but I know of-((the sec-)) know of none in which they're not.*

A myriad of leading technology companies have gained traction on both speech recognition and natural language processing, through products such as Apple's Syri, Amazon's Echo, and Google's and Microsoft's machine learning APIs. Regardless of Chomsky's point as to whether these successes in natural language processing constitute genuine scientific insight, there does exist some success in the extent to which people are communicating with computers through human language.

The purpose of this project is to both to explore the capability of deep learning through Google's open sourced Tensorflow library to engage in human conversation, while at the same time side stepping some of the limitations of statistical models that Chomsky pointed out. Specifically, this project will create a chatbot that will train on a movie conversation database as well as a dataset of climate change FAQs. The point is not to teach the chatbot of concepts in climate science, but rather to have it feign an understanding by simply identifying common questions about climate change and responding with pre-constructed answers to those questions.

## Metrics

In order to properly train a natural language processing model, the most reliable metric to use is perplexity, which is essentially a measure of the predictive power of the output probabilities calculated by the model. Given a dataset of conversations, this boils down to maximizing the probability that the model would predict an observed response, given a prompt from the dataset.

# Model Architecture

## Recurrent Neural Network Architecture

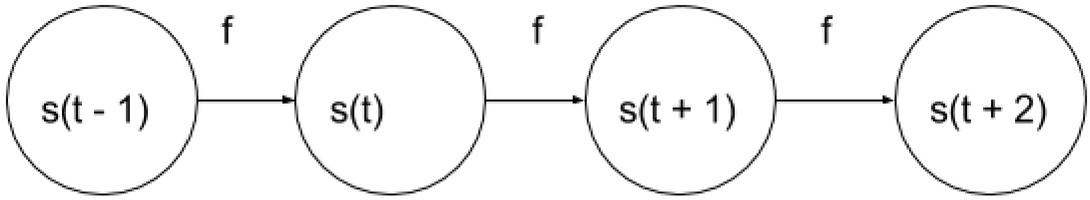
As discussed in the project proposal, Recurrent Neural Networks (RNN) represent the current paradigm of training models for development of chatbots. A critical difference between feed-forward and RNNs is that RNN units forms an acyclic computation graph. Consider the equation:

$$f(s^{t}) = f(s^{(t-1)}, x)$$

For instance:

$$f(s^3) = f((s^2), x) = f(f(s^1, x), x)$$

Basically this states that the current state of the function depends on a prior state, which can be represented as the following acyclic graph:




Note that *t* does not actually have to represent *time*, but can rather represent any sequential position - word position in a sentence in our case.

There are two features of this architecture that make RNNs a powerful tool in natural language process. First, weights can be shared across different position *t* for a given parameter *x*. For instance, if you take a word “bat”, it will return a different based on its position without necessarily having to create completely separate parameter for the word at any position in the sentence. You will therefore get a different output state if the word is proceded by a grammatical modifier such as “baseball” or “flying”, even though “bat” only need be represented by a single parameter within the model.

Second, the same transition function *f* can be used at every time step *t* and accept an input sequence of arbitrary length because it describes transition from one state to the next, rather than a variable length history of states.

# LSTM and GRU Cells

While the above RNN structure is sufficient for a creating a neural network with a basic recurrence mechanism, it is not sufficient for creating a model that can adequately capture the larger context of a word within a sentence, let alone a paragraph.

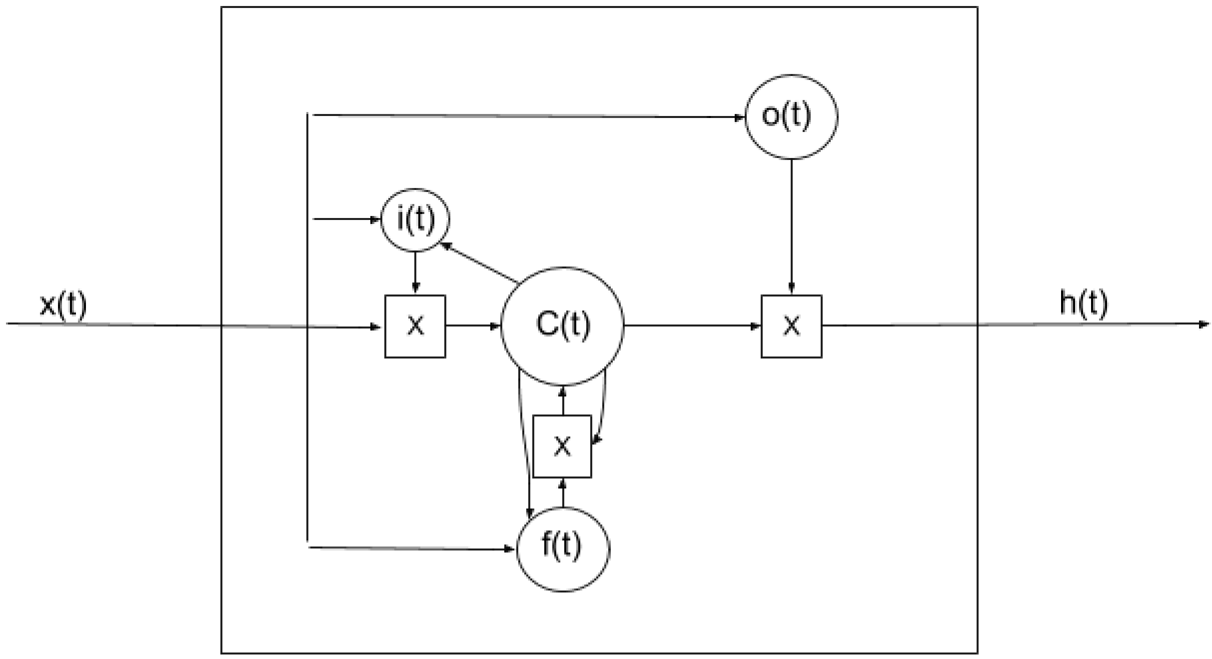
The reason for this shortcoming is a problem within other neural network architectures, namely the problem of vanishing or exploding gradients. Consider the following abstraction from the Recurrent Neural Network chapter from Goodfellow et al 2016:

$$
\begin{aligned}
h^{(t)} &= (W^t)^\top h^{(0)} \\
W &= Q\Lambda Q^\top \\
h^{(t)} &= Q\Lambda^t Q h^{(0)}
\end{aligned}
$$

where *h* is a simplified recurrence relationship without inputs or a non-linear activation function, *W* is a weight matrix with the eigen decomposition function above, and *Q* is the orthogonal matrix to *W*. It is clear from this simplified abstraction that gradients at time step τ away from the current state will be subject to exponentially decaying or growing gradient updates (Goodfellow et al 2016).

A myriad of solutions have been proposed to alleviate this problem, but recently researchers have success through the use of neural networks with gated recurrence units. Essentially, these gated neural networks create additional weight parameters for each unit, which are used to determine what information the unit accepts and passes on to the subsequent unit.

The more well known gated recurrence unit is the long short term memory unit (LSTM). It has the following architecture:




*i(t) - input gate*
*f(t) - forget gate*
*o(t) - output gate*

A gated recurrent unit (GRU) is very similar, however, it lacks an output gate, making the computation more efficient. Research has shown the GRU cell to achieve performance comparable to that of LSTM cells making the tradeoff worthwhile for efficiency benefits (Chung et al 2014).

## Clipping

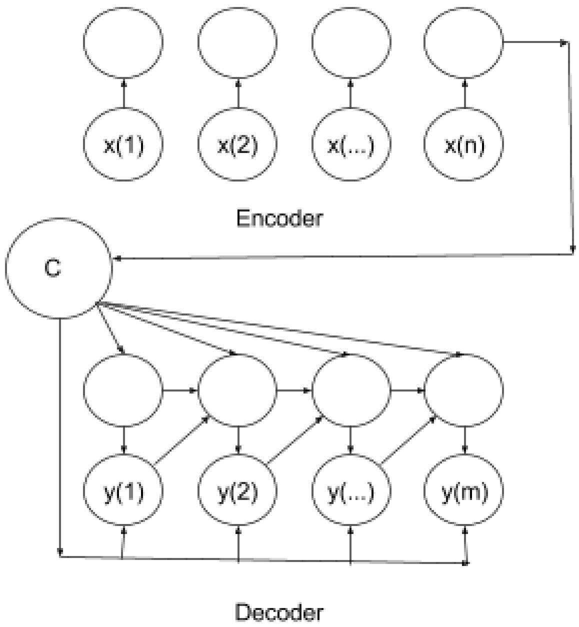
Another strategy RNN models can adopt, which can be used in combination with the gated recurrence networks above, is to clip the gradient norm before updating cells. This approach guarantees that if the gradients explode, update steps will be limited when gradients begin to explode, but can proceed at an the expected rate while at the flatter sections of the gradient slope (Goodfellow et al 2016, 415).

## Encoder-Decoder Sequence to Sequence Models

The methods discussed above are able to directly compute a vector to variable length sequence and vice versa. However, in the realm of translation and conversation, there are no grammatical rules that limit the length of either an input or output phrase or sentence. An additional abstraction is necessary in order to for the model to process input of variable length and output an optimal response of undefined length. The is where the encoder and decoder structure comes into play.

The encoder-decoder structure basically provides forges a connection between input of arbitrary length to output of arbitrary length. It does so by creating an intermediary context variable of fixed length. This variable essentially contains a summary of the input. Once this context has been encoded from the input, the decoder can calculate an optimal response. This is useful in translation as well as in chatbots. This architecture is essentially able to allow models to consider the full context of an input before generating any output. Concretely, the encoder may calculate a similar context variable for the phrases “How are you?”, “How’s it going?”, or “Cómo estás?”. Their significances are similar and may all evoke similar responses. The decoder may then generate the appropriate response or translation - “Doing great, thanks!”, “I’ve been better”, or “How are you?”.

The following is an adaptation of the encoder-decoder architecture from [Goodfellow et al 2016](), 396.




## Including an Attention Mechanism

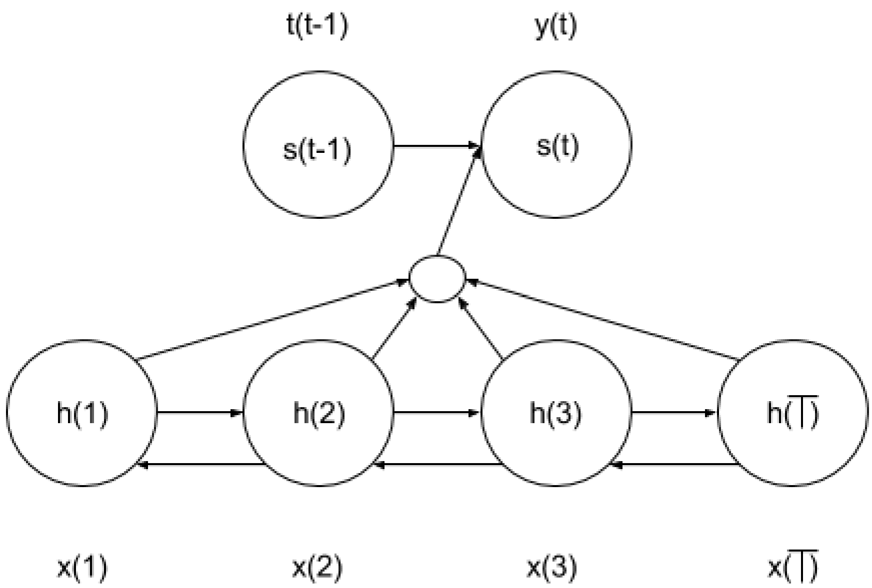
While the encoder-decoder structure enables the model to operate in more real world scenarios with variable input and output lengths, information from the original input may be lost as it is abstracted into a single context variable.

To address this limitation, Bahdanau et al 2016 suggested modeling an attention mechanism for the decoder. Essentially this allows the decoder to search for specifically relevant inputs in the source sentence. Such an attention mechanism is composed of three parts (Goodfellow et al 2016, 476):

1. An encoder process that reads raw input data and converts them to a vector representation.
2. A memory consisting of a list of these feature vectors, which the decoder can later directly access outside the bounds of the sequence.

3. A decoder process that accesses the stored memory and can place *attention* on vectors of particular relevance.

The key architectural difference here is that the context vector is actually a sequence of *annotations* (represented by *h* in the figure below) where each annotation "container information about the whole input sequence with a strong focus on the parts surrounding the i-th word of the input sequence". Critically, this requires a backpropagation step in order for the i-th annotation to reflect its significant within the context of the whole sentence, not just in the words preceding it. Below is a general depiction of this model (Bahdanau et al 2016, 3):




# Implementation in Tensorflow

A very similar to the encoder-decoder architecture described above were originally prescribed for machine translation in Cho et al 2014. With the additional insights from Bahdanau et al 2016, Google created a tutorial for implementing this architecture using Tensorflow.

I actually used a slightly modified model from Google's original, llSourcell/tensorflow_chatbot, which was used for a Youtube chatbot competition. This model required Tensorflow r0.12. Below is a summary of the Tensorflow functions used for the model.

## tf.nn.rnn_cell.GRUCell

First, our sequence to sequence cell is initialized. The model accepts either an LSTM or GRU type. For efficiency's sake, as discussed above, we used the GRU cell. This accepts a parameter for the number of units within each layer of the model. In our case we use 256.

## tf.nn.rnn_cell.MultiRNNCell

Additionally, we could customize the number of layers within the model. In the case that we use more than one layer (we, in fact, used 3), we pass a GRU cell from above for each layer. These are then wrapped in this MultiRNNCell.

## tf.nn.seq2seq.embedding_attention_seq2seq

This function implements the encoder-decoder sequence to sequence model with an attention mechanism described above. This function accepts the encoder and decoder inputs, as well as their respective vocabulary sizes (see Data Preparation below), the number of  units per layer, and of course the GRU cell.

This function returns a list of predicted outputs as well as the state of each decoder cell after the full sequence has been predicted.

## tf.nn.sampled_softmax_loss

This function is particularly useful for neural networks with a large number of output classes. Of course, any neural network being trained to translate or respond to inputs falls into this category, as their are generally thousand of commonly used words within any language vocabulary set.

This function will accept the inputs of to the encoder, apply corresponding weights and biases, and then return a softmax loss calculated relative to the decoder inputs, but only for a random sample of candidate output vocabularies. This dramatically speeds up training and is necessary for any natural language processing mode.

## tf.nn.seq2seq.model_with_buckets

Lastly, we wrap the sequence to sequence model with attention mechanism in a bucketed model. This allows Tensorflow to bucket inputs and outputs into buckets determined by maximum permissible lengths.

Why is this necessary? Basically, if we were to pass literal encoder and decoder inputs, we would need a separate sequence to sequence model for each length of input and output token. Alternatively, we could add padding symbols for the difference between the maximum sequence length and the actual sequence length. This, in many cases, would create sequences with many padding symbols. Bucketing is an alternative approach that allows short sequences to limit their padding symbols by being bucketed into datasets with smaller permissible maximum lengths.

This function accepts the sampled softmax loss function defined above and will output the decoder outputs, as well as the calculated loss.

## A Note on Metrics and Optimization

In order to optimize the model, it is common to use *perplexity* in language modeling. Essentially, perplexity is a measure of the predictive power of a probability distribution relative to a recorded input. In our case, the softmax function is applied to decoder outputs for a select set of randomly chosen vocabulary outputs (as described in tf.nn.sampled_softmax_loss above). These softmax outputs are then compared to observed responses. The model additionally includes applies gradient Clipping. A tensorflow gradient descent optimizer then applies these clipped gradients.

The model object is thus to minimize perplexity - ie maximize the likelihood that the model will output observed responses.

# Deviation from Original Tensorflow Model

One of my original goals was to re-factor the Tensorflow model in the tutorial to use the Tensorflow estimator interface, I found this challenge, while worthwhile, impossible to implement. The object was to capture the advantages of distributed, parallel training on the Google cloud platform, in order to speed up training.

However, while the estimator interface is straightforward and useful for regression, straightforward deep neural networks, and even convolution, the model architecture discussed above does not as easily fit into the estimator interface, due mostly to recurrence and bucketing.

While I was not able to achieve this goal, the exercise elucidated the unique qualities of recurrent neural networks with GRU cells and bucketing.

# Benchmark

The competition organizer of the adapted model from llSourcell/tensorflow_chatbot does not report any explicit benchmarks. However, there are several issues in Github related to user's achieved perplexity. The lowest reported perplexity was 1.35 (issues/8), though several users reported much higher perplexities after 12+ hours of training (issues/48). These issues may largely be related to hardware.

While these provide some useful benchmarks for calibrating success, the expected results here will differ due to the additional data processing steps discussed below.

# Data Preprocessing

The base dataset for this project is the Cornell movie dialog dataset. The dataset includes:

- 220,579 conversational exchanges between 10,292 pairs of movie characters

- 9,035 characters from 617 movies
- total 304,713 utterances

The dataset provides a sort of benchmark dataset for creating and evaluating conversational machine learning models. The data pre-processing for this dataset is as follows:

1. Create the vocabulary for both encoder and decoder inputs (ie the prompts and responses). This involves iterating over all of the prompts and responses and counting occurrences of every word. The list of prompt and response words is then filtered to the most n words (we used a parameter of 25,000 for n). These vocabularies are then stored for later processing of inputs and outputs.
2. Additionally, the following functional words are included in the vocabulary:
    a. _GO - which represents the start of a response string.
    b. _EOS - which represents the end of a response string.
    c. _PAD - which is appended to the end of all prompt and response strings up until the maximum length of their corresponding bucket.
    d. _UNK - which is used for any prompt or response word not saved in the vocabulary above.
3. Encode every prompt and response by id assigned to each prompt and response vocabulary word.
4. Add the functional vocabulary words mentioned above to create sequences of appropriate length.
5. Reverse the order of the prompt sequences, which creates more short term dependencies and simplifies the optimization problem and in turn better performance (Sutskever et al 2014, 2).

# Climate Augmented Dataset

The goal of this project was to create a chatbot with a practical and specific purpose, namely answer questions about climate change. Given the complexity of natural language processing and climate change, it made little sense to combine these two tasks without some tricks. As the point Chomsky made suggests, deep learning results are purely statistical models. They do not create nor understand fundamental concepts, such as the greenhouse effect or the significance of the word “climate”, “temperature”, or flood.

To create a chatbot that did indeed have a sense of the significance of these words would require not just loads of conversational data, but loads of climate and economic data. These data would be processed by a completely different model than the architecture discussed here. I, therefore, decided to *meta* tokenize the response data used in the model. Here is a how the climate augmented data was incorporated into the model.

1. I searched for the internet for FAQs on climate change. I saved 209 FAQs in total from the sources included in Appendix I.
2. For each of these FAQs, I created 5 additional paraphrases of the questions and gleaned a further three from Amazon’s Mechanical Turk.
    a. I originally planned to use Microsoft’s paraphrase API. They have deprecated this API. Ultimately, while this cost me an additional day and some

Mechanical Turk Fees, the input data was probably much better from a human source than another AI source.

3. For each response, I created a unique meta-token, which was saved in a JSON lookup file, and appended each paraphrase along with the response meta-token to the Cornell movie dialog dataset.
4. I included all vocabulary words from the climate augmented dataset in the final vocabularies by removing the least used vocabulary words from the Cornell movie dialog dataset.

The drawback to the meta-token approach is that the decoder loses some integral understanding of grammatical structure. However, given the goal of this project, this seemed like a worthwhile approach.

# Results

## Final Perplexity

After train for about 12 hours on a Macbook Pro 2.5 GHz i7, 16 GB memory, Nvidia GeForce GT 750M 2048 MB. After 59700 steps, I had a achieved a perplexity of about 4.15. This is on par with the better end of the perplexities reported in the competition Github issues, however, it was not the best. Ultimately, because my dataset included 209 x 9 more prompts and one word response, this perplexity seems acceptable.

## Example Chat

The chatbot did achieve its subjective goals. I wanted to build a chatbot that could respond to very basic prompts such as greetings and questions such as “How are you?”. While the chatbot does not recognize more esoteric prompts, the chatbot can maintain simple grammatical structure in its responses, even when the content does not make a whole lot of sense..

```
> Hi.
Hi .
> How are you?
Okay .
> Why just ok?
We have to find the car .
> You lost the car?
What the hell is it ?
> I can't believe you lost the car.
That ' s okay . He ' s just a little guy .
> Even so cars are not cheap.
Am I out of your work ?
> I wouldn't hire you if you lose cars.
Oh , yeah , I ' m sure that ' s all right .
```

In addition to this subjective result, I was pleased with the ability of the chatbot to find relevant responses about climate change when prompted. This illustrates the ability of the encoder to properly identify the significance of a prompt.

> What is climate change?

The Earth’s average temperature has risen by 1.5°F over the past century, and climate scientists estimate it will rise another 0.5 to 8.6°F by the end of this century, depending, in part, on future emissions.[1] That may not sound like much to worry about, since most of us experience much greater temperature changes over the course of a day or from season to season. But the global average temperature during the height of the last ice age was only 5 to 9°F cooler than it is today.[3] Relatively small changes in the planet’s average temperature can mean big changes in local and regional climate, creating risks to public health and safety[4], water resources, agriculture, infrastructure, and ecosystems.[5] Following are some examples: [5] Increasing heat waves: Heat waves have become more frequent in the United States in recent decades. Climate scientists expect the number of days with temperatures above 90°F to increase in the United States as the climate changes, especially toward the end of this century. More extreme weather: In addition to heat waves, changes in precipitation patterns, including extreme precipitation events, storms, and floods, are becoming more common and more severe in many regions, and this is expected to continue. Intensified droughts: Higher temperatures lead to increased rates of evaporation and can lead to more rapid drying of soils. Without reductions in global greenhouse gas emissions, longer-term droughts are expected to intensify in much of the Southwest, the Great Plains, and the Southeast. Impacts on crops: Over the past 40 years, climate disruptions to agricultural production have increased, and this is expected to continue. Impacts on health: Climate change is increasing our exposure to extreme temperatures, extreme weather events; degraded air quality; diseases transmitted through food, water, and insects; and stresses to mental health and well-being. These threats to human health are expected to increase with continued climate change. More wildfires: The area burned by wildfire in parts of western North America is expected to double (or more) for each 1.8°F increase in global average temperature.[6] Rising sea levels: Global sea level has risen by about eight inches since the late 1800s, and is projected to rise another 1 to 4 feet by the end of this century. Flooding is becoming more frequent along the U.S. coastline, especially in the Mid-Atlantic region where the land is simultaneously sinking. Impacts on health: Climate change is increasing our exposure to extreme temperatures, extreme weather events; degraded air quality; diseases transmitted through food, water, and insects; and stresses to mental health and well-being. These threats to human health are expected to increase with continued climate change.

> Will there be more extreme weather events due to global warming?

Scientists have published strong evidence that the warming climate is making heat waves more frequent and intense. It is also causing heavier rainstorms, and coastal flooding is getting worse as the oceans rise because of human emissions. Global warming has intensified droughts in regions like the Middle East, and it may have strengthened a recent drought in California. In many other cases, though, the linkage to global warming for particular trends is uncertain or disputed. That is partly from a lack of good historical weather data, but it is also scientifically unclear how certain types of events may be influenced by the changing climate. Another factor: While the climate is changing, people’s perceptions may be changing faster. The Internet has made us all more aware of weather disasters in distant places. On social media, people have a tendency to attribute virtually any disaster to climate change, but in many cases there is little or no scientific support for doing so.

> That's terrible.

Yes .

> Can I do anything to prevent global warming?

Because most human-produced heat-trapping gases come from burning fossil fuels, there is great potential for the collective actions of many individuals worldwide to reduce global warming by making changes in their daily and annual activities that produce heat-trapping gases and aerosols. Specifically, people can consider making the following choices in their personal lives: reduce household energy use through use of energy efficient appliances and heating and air conditioning systems; increase investments in renewable energy sources such as solar and wind power systems; avoid unnecessary household energy use through lighting and temperature control options as well as the use of power strips with switches enabling people to turn off always-on "vampire" appliances (i.e., computers and cable TV boxes); and limit travel distances in conventional automobiles and aircraft while choosing energy-efficient mass transportation options, such as trains and buses, where possible. Making the best choices to reduce emissions requires accurate and quantitative information about how our different lifestyles cause emissions. Examples of direct emissions are energy use in households, automobiles, and air travel. Indirect emissions result from production and distribution of goods used in household and businesses. More guidance on courses of action can be found in the National Academy of Sciences' 2010 report, titled Informing an Effective Response to Climate Change.  As addressed in previous questions, stabilizing global temperature at its current level requires eliminating all emissions of heat-trapping gases or, equivalently, achieving a carbon-neutral society in which people remove as much carbon from the atmosphere as they emit. Achieving this goal will require substantial societal changes in energy technologies and infrastructure that go far beyond the collective actions of individuals and households to reduce emissions.

# Conclusion

The goal of this project was to create a chatbot that was especially trained towards a specific ends, in this case, answering questions about climate change. The quote provided by Chomsky at the beginning of this paper points to a limitation of current machine learning methods I had to side step in order to achieve this goal. While there may come a day when we can feed deep neural networks data from disparate sources and expect it to gain a grasp of underlying concepts and structures, or deeply integrate disparate concepts such as climate change and grammar, this was not the intention of the paper.

Rather, I set out to train a chatbot on a dataset from movie dialogs in hope that it would be able to carry on a very basic conversation and both recognize and create underlying grammatical structure in conversational prompts and responses. We would then piggyback on the abilities of the encoder of our sequence to sequence architecture to recognize and categorize the significance of questions related to climate change. With this more simple ability, we are able to create a chatbot that can adequately respond to questions about climate change.

I think there are two major areas of improvement for the final model. First, I would have liked to be able to take advantage of Tensorflow's parallelization capabilities and run this model in Google cloud through the estimator interface. This may be possible, but it would require unpacking the high level methods used in this model, such as the *model_with_buckets* method.

More fundamentally, in hindsight, I think there is a more optimal structure for the objectives here. Essentially, the ability to decipher the significance of a question is the responsibility of the encoder. In turn, the decoder is responsible for creating a response based on that significance, which requires the ability to create sentences with proper grammatical structure.

For simple FAQs, with prepared responses, the decoder's responsibility is not entirely essential, since it is required to map to a single category rather than compose a sentence with multiple tokens.

With this in mind, I could very well have created separate models for the movie and climate change datasets. I could first train on a larger dataset of climate change FAQs. In addition to adding more paraphrases for the questions, I would also add prompts with null responses, ie questions and statements unrelated to climate change and map these to null values - the movie dataset may be sufficient for the input dataset, unless of course if the data set contains dialog from *Before the Flood* or *An Inconvenient Truth*. Crucially, I could then pass live prompts through both models. If the climate change model returns a null response, it would then pass the prompt to a more general chatbot, which could respond accordingly.

While this approach would indeed be more efficient from a training time standpoint, it has the benefit of enabling a model that is more efficient and performant in the task of simply categorizing questions into a limited number of buckets (ie the answers from the climate change FAQs) - ie a model architecture without the complexity of the decoder. The added benefit of separate models is that it would be able to easily swap out the generalized chatbot as chatbot technology continues to improve.

# Works Cited

arXiv:1406.1078v3 [cs.CL]. Cho et al, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." https://arxiv.org/pdf/1406.1078.pdf, 3 Sep 2014.

arXiv:1409.0473 [cs.CL]. Bahdanau et al, Machine Learning Translate By Jointly Learning to Align and Translate. https://arxiv.org/pdf/1409.0473.pdf, May 2016.

arXiv:1409.3215v3 [cs.CL]. Sutskever et al, "Sequence to Sequence Learning with Neural Networks." https://arxiv.org/pdf/1409.3215.pdf, 14 Dec 2014.

arXiv:1412.3555v1 [cs.NE]. Chung et al, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." https://arxiv.org/pdf/1412.3555v1.pdf, 11 Dec 2014.

Goodfellow et al, Deep Learning. http://www.deeplearningbook.org, MIT Press, 2016.

# Appendix A - Data Sources

Please see this Google spreadsheet for the original list of questions and answers, as well as the paraphrases:

https://docs.google.com/spreadsheets/d/1yexja22mo94y0h4Dwk4VQz9fZPrOkm779SXguQA3SL0/edit?usp=sharing

The original FAQ questions and answers were derived from the following sources:

"Climate Science FAQs." USDA Climate Hubs. Web 15 July 2017. https://www.climatehubs.oce.usda.gov/content/climate-science-faqs

"Frequently Asked Questions." Global Climate Change Vital Signs of the Planet. NASA.Web. 15 July 2017. https://climate.nasa.gov/faq/.

"FAQs About Rapid Climate Change." Utah Education Network. Web. 15 July 2017. http://www.uen.org/climate/faq.shtml.

"Climate Science FAQ." ClimatePrediction.net. Web. 15 July 2017. http://www.climateprediction.net/climate-science/faqs/.

"Five Outstanding Questions in Earth Science." Earth Magazine. Web. 15 July 2017. https://www.earthmagazine.org/article/five-outstanding-questions-earth-science.

"FAQ on Climate Models." RealClimate. Web. 15 July 2017. http://www.realclimate.org/index.php/archives/2008/11/faq-on-climate-models/.

IPCC, 2007: Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change [Solomon, S., D. Qin, M. Manning, Z. Chen, M. Marquis, K.B. Averyt, M.Tignor and H.L. Miller (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA. Web. 15 July 2017. https://www.ipcc.ch/pdf/assessment-report/ar4/wg1/ar4-wg1-faqs.pdf.

"Global Warming/ Climate Change Frequently Asked Questions." Environmental and ENerty Study Institute. Web. 15 July 2017. http://www.eesi.org/climate-change-FAQ.

"Frequently Asked Questions About Climate Change." Environmental Protection Agency. Web. 15 July 2017. https://19january2017snapshot.epa.gov/climatechange/frequently-asked-questions-about-climate-change_.html

"Frequently Asked Questions." University Corporation for Atmospheric Research. Web. 15 July 2017. https://www2.ucar.edu/contact-us/faq.

"Global Warming FAQ." Union of Concerned Scientists. Web. 15 July 2017. http://www.ucsusa.org/global_warming/science_and_impacts/science/global-warming-faq.html

Gillis, Justin. “Short Answers to Hard Questions About Climate Change.” 6 July 2017. Web. 15 July 2017. https://www.nytimes.com/interactive/2015/11/28/science/what-is-climate-change.html?_r=4

“How do human CO2 emissions compare to natural CO2 emissions?” Skeptical Science. Web. 15 July 2017. https://skepticalscience.com/human-co2-smaller-than-natural-emissions.htm.

“Could Warmer Oceans Make Atmospheric Carbon Dioxide Rise Faster Than Expected?” Science Daily. Web. 15 July 2017. https://www.sciencedaily.com/releases/2007/10/071023163953.htm.

“Global Warming Frequently Asked Questions.” Climate.gov. Web. 15 July 2017. https://www.climate.gov/news-features/understanding-climate/global-warming-frequently-asked-questions#hide4.

“Frequently Asked Questions.” National Center for Environmental Information. Web. 15 July 2017. https://www.ncdc.noaa.gov/monitoring-references/faq/.

“FAQs.” National Climate Assessment. Web. 15 July 2017. http://nca2014.globalchange.gov/report/appendices/faqs.