Machine Learning Engineer Nanodegree

Capstone Project

Eric Hulburd

4 August 2017

I. Definition and Overview

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.

Project Overview

As mentioned above, in the past 5-10 years, much progress has been made within the realm natural language processing performance, mainly due to increased computing power as well as the progress of deep learning algorithms.

This has specifically been the case for both machine translation and chatbots. These realms within MLP represent very similar problems. There are really two main tasks for these types of models. The first is to extract meaning from a verbal prompt. The second is to generate an output from that prompt - either a translation, or a response in the case of a chatbot.

The major challenge in developing good models for these tasks is to develop a model that (1) develops an adequate sense of context and (2) effectively relates inputs to outputs. Sequence to sequence models provide an appropriate architecture to meets these challenges. These models are charged with the following tasks:

- accept a tokenized input (ie word to vector),
- develop a sense of meaning of the full input,
- develop a sense of significance of each token within the broader context of the full input,
- generate a response that corresponds to the significance of the input, as well as the formulation of its individual tokens.

In order to accomplish this, the model needs a large set of input and outputs. In our case, we use a database of conversations from movie scripts. We then a set of questions and answers pertaining to the set of climate change. We will remove complexity by meta-tokenizing long, scientifically dense answers from this climate set so the model only has to recognize a class of response to climate change questions, rather than generate a response itself.

The specifics of model architecture and data processing steps are discussed in more detail below.

Problem Statement

In summary, the problem we are trying to solve is to train a chatbot to specialize in responding to a special class of prompts, namely questions about climate change.

We will build off of work currently done within the realm of machine translation. The sequence to sequence will provide a scaffolding for training a model that encodes inputs text strings and decodes generated output to text string that represent responses rather than translations. We will abstract out the complexity of long, scientifically dense responses with the use of meta-tokens.

After training on a large dataset, we should be able to create an interface where a user can input prompts, as they would to a friend over Whats App, and the chatbot should return a response that is coherent, albeit mechanical.

When the user inputs a question or prompt relating to climate change, the chatbot should identify the appropriate class of response, in the form of the aforementioned meta-token. We can then simply program the interface to detect the meta-token, and replace it with a full scientific response which it encodes.

The necessary steps for implementing this solution are as follows:

1. Gather general conversational data set for training.
2. Gather a robust set of climate change questions and answers.
3. Replace long, scientifically dense climate responses with meta-tokens.
4. Parse those inputs and outputs, and create a set of vocabulary words for both inputs and outputs. The output vocabulary must include *all* of the climate meta-tokens and include special tokens representing beginning of a response, end of a response, and unknown vocabularies.
5. Tokenize the input and output datasets. In other words, transform inputs and outputs to vectors with elements that are integers that represent a word from the source vocabulary.
6. Use this data to run a sequence to sequence model that predicts the appropriate response from a given input.
7. The model should maximize the probability that a given input would result in the source data output. Metrics to these ends are discussed in more detail below.

Metrics

Determining a metric for machine translation and chatbot learning is much more difficult that simple classification tasks. While such learning is still considered supervised learning, since models learn based on human transcribed “answers” from the real world, there are myriad different “answers” for any given input. For instance, below are valid translations and responses for the question “How are you?”:

- I’m good, how are you?
- I’ve been better.
- Doing great.
- Cómo estás?
- Qué tal?

While those are all appropriate responses and translations, they are clearly better than following responses and translations:

- The sky is blue.
- The ocean is deep.
- Latinoamérica incluye México, Centroamérica, y Sudamérica.

Ultimately, we calculate the loss using a softmax function, where categories represent a token within our generated vocabulary vocabulary. Since we use a vocabulary of 25,000 tokens, in practice we use a *sampled* softmax function, which evaluates the function only over a subset of the vocabulary for each loss calculation.

$$P(y = j \mid x) = (e^{x^T w_j}) / (\Sigma_{k=1}^{K} e^{x^T W_k})$$

This softmax function will output the probability that a given input vector will output $y = j$. This probability distribution can then be used to predict the model’s cross entropy.

Additionally of note, these losses are computed individually for different length categories of sentences. This is discussed in further detail below, but basically this means a given input-output pair is bucketed based on the maximum length of the two. This allows the model to use a separate set of weights for sentences of 5 words and 50 words, thus enabling the model to optimize more efficiently by avoiding calculations on padding symbols for short inputs and outputs.

Of course, in natural language processing, for the reason mentioned above, there is no *known* probability distribution. The best we have is the input and output data from our test set. The following equation calculates the cross entropy of a given set of inputs and outputs (N), and the probability determined by the softmax function above (q(**x**)).

$$H(T, q) = -1/N \, \Sigma_{i=1}^{N} log_2 \, q(x_i)$$

Note that the higher the probability a given input outputs the corresponding output from the test set, the lower the cross entropy. This metric is taken a step further in natural language modeling by calculating the *perplexity*.

$$2^{H(T,q)}$$

Perplexity can be thought of as the number of random variables of a probability distribution. For instance, rolling a six sided die has a perplexity of 6, while choosing a number 1-10 at random has a perplexity of 10. A higher perplexity reflects a higher degree of uncertainty.

With regards to our specific model, we can interpret the perplexity as the uncertainty with which our model predicts a given output from our dataset. The negative cross entropy value above in this interpretation represents the number of bits required to encode a given word within our vocabulary. Given a particular input and a negative cross entropy of 4 (ie there is a 25% chance of the model choosing the output in the dataset), the model would have a perplexity of 16 per word.

Note, the upper bound for the 5.96 million character Brown Corpus of English text was estimated in 1992 as 1.75 bits per character or 7.95 bits per word, yielding a perplexity of about 247 (Brown et al).

II. Analysis

Data Exploration

We train our model on two different datasets. The first 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 well formatted base dataset for training the chatbot to be able to encode a wide range of input prompts and derive response from them.

Additionally, to enable the chatbot to converse intelligently about climate change, I gathered 209 question and answers about climate change from sources listed in Appendix A. Because these 209 questions and answers three orders of magnitude smaller than the general conversation, I also created five paraphrases for each question set, and generated an additional three through Amazon's Mechanical Turk.

This manual procedure replaced the original plan of generating paraphrases through Microsoft's paraphrase API, which has been deprecated. In the end, using human paraphrases worked to the advantage of the subjective performance of the chatbot. For

example, a robotic paraphrase API may easily substitute the work “weather” for “climate,” whereas the question and answer sets make critical distinctions between these two words.

Exploratory Visualization

Please see results section for visualizations of the analyzed datasets.

Algorithms and Techniques

As mentioned in the Definition and Overview section above, the fundamental challenge of machine translation and conversation is recognizing input significance, both at the sentence and word level, and generating an appropriate output. The model architecture and Tensorflow methods used in this project to address this challenge are discused below.

Recurrent Neural Network Architecture

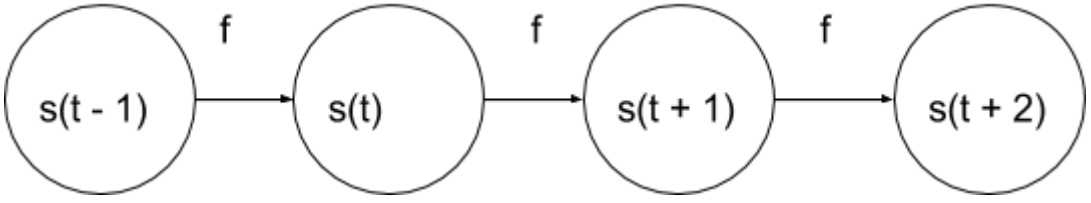
As discussed in the project proposal, Recurrent Neural Networks (RNN) represent the current paradigm of training models for development of chatbots. A critical difference between feed-forward and RNNs is that RNN units forms an acyclic computation graph. Consider the equation:

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

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.

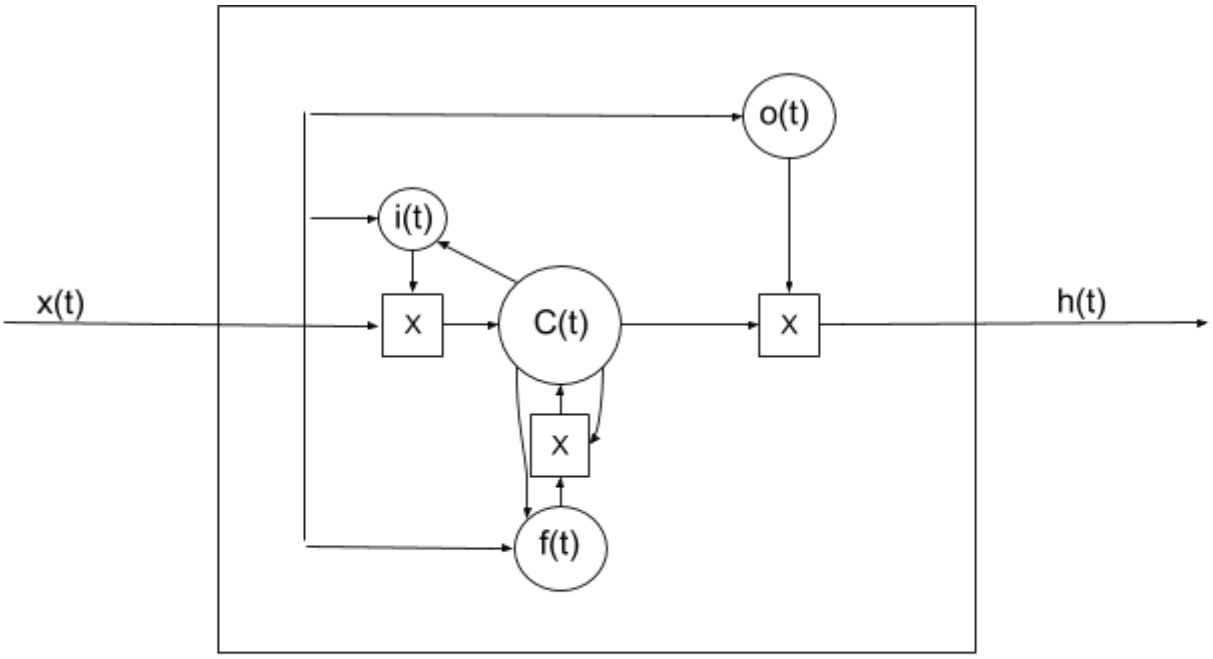
The reason for this shortcoming is a problem within other neural network architectures, namely the problem of vanishing or exploding gradients. Consider the following abstraction from the Recurrent Neural Network chapter from Goodfellow et al 2016:

$$h^{(t)} = (W^t)^\top h^{(0)}$$
$$W = Q\Lambda Q^\top$$
$$h^{(t)} = Q\Lambda^t Q h^{(0)}$$

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

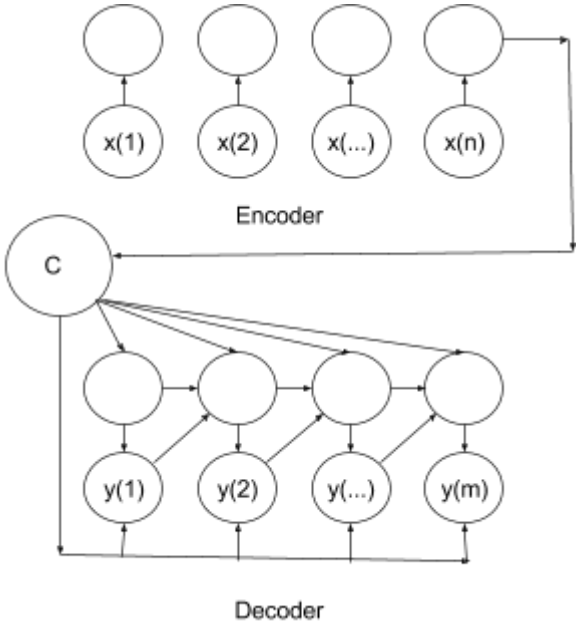
Another strategy RNN models can adopt, which can be used in combination with the gated recurrence networks above, is to clip the gradient norm before updating cells. This approach guarantees that if the gradients explode, update steps will be limited when gradients begin to explode, but can proceed at 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

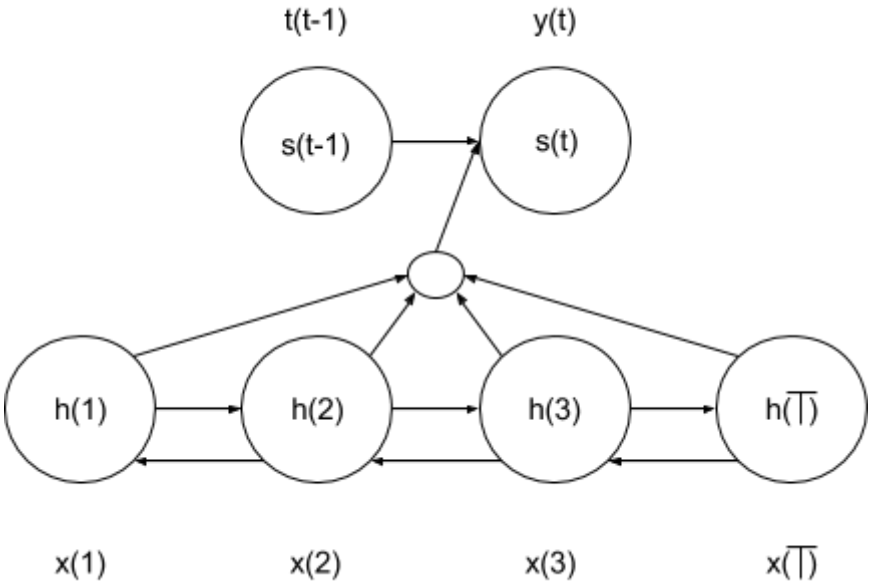
While the encoder-decoder structure enables the model to operate in more real world scenarios with variable input and output lengths, information from the original input may be lost as it is abstracted into a single context variable.

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):



Benchmark

The original idea for the implementation used in this project was Cho et al's encoder-decoder model for French-English translation. This model was subsequently applied in Tensorflow and detailed in its Sequence to Sequence tutorial.

This tutorial suggests using a similar framework for creating a chatbot. This inspired an online competition through Youtube by Siraj Raval in his video "How to Make an Amazing Tensorflow Chatbot Easily" (https://www.youtube.com/watch?v=SJDEOWLHYVo&t=55s). The chatbot in this paper builds upon the model adopted from Tensorflow's Sequence to Sequence tutorial in this video - see github.com/llSourcell/tensorflow_chatbot.

The competition organizer of the adapted model from 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

perplexities over 4-6 after 12+ hours of training (issues/48). Competitors did not report validation set perplexity.

Additionally, as a higher level point of reference, Montemurro et al 2011 reported an entropy of 5.7 bits per word for the English language and 9.1 for shuffled text. These numbers correspond to perplexities of 52 and 549, respectively. The limited dataset used in this project, as well as the data pre-processing methods, greatly reduce perplexity of the final model results.

III. Methodology

Data Preprocessing

As mentioned earlier, the Cornell Movie conversation dataset provides the base data for training the model in general conversation structure. 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).

Including the climate data provided additional pre-processing challenges. 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 conversational data, but also 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. Data pre-processing procedures for the climate FAQs were as follows:

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 training data is not representative of conversational grammar. However, given that the climate data set was smaller, this seemed like a much more reasonable approach than to expect our model to learn long, scientifically dense output responses.

Implementation

Implementation followed the Tensorflow Sequence to Sequence tutorial for French-English transation. This model reflected the encoder-decoder architecture discussed in Cho et al 2014, as well as the attention mechanism described in Bahdanau et al 2016 that is discussed above.

The original tutorial was written in Tensorflow r0.12 and used the following built in Tensorflow functions.

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.

Refinement

There were four main parameters that I explored while refining the model:

- number of layers
- units per layer
- maximum gradient norm

- vocabulary size.

Unfortunately, hardware restrictions limited the model size to a 3 256-unit layers. Configurations above this threshold exhausted GPU memory. A preliminary project goal was to refactor the code to implement Tensorflow's Estimator interface, so that it could run on Google cloud. However, this refactorization was not possible due to the complexities of the model architecture, specifically those related to recurrence, discussed above.

Additionally, in order to ensure that the model was adequately exposed to all relevant climate vocabulary, we increased the vocabulary size to 25,000 from an original size of 20,000 and included all climate input vocabulary and all response meta-tokens, without regard to the overall frequency of those tokens within the whole dataset.

IV. Results

Final Perplexity

The model was run on a Macbook Pro 2.5 GHz i7, 16 GB memory, Nvidia GeForce GT 750M 2048 MB.

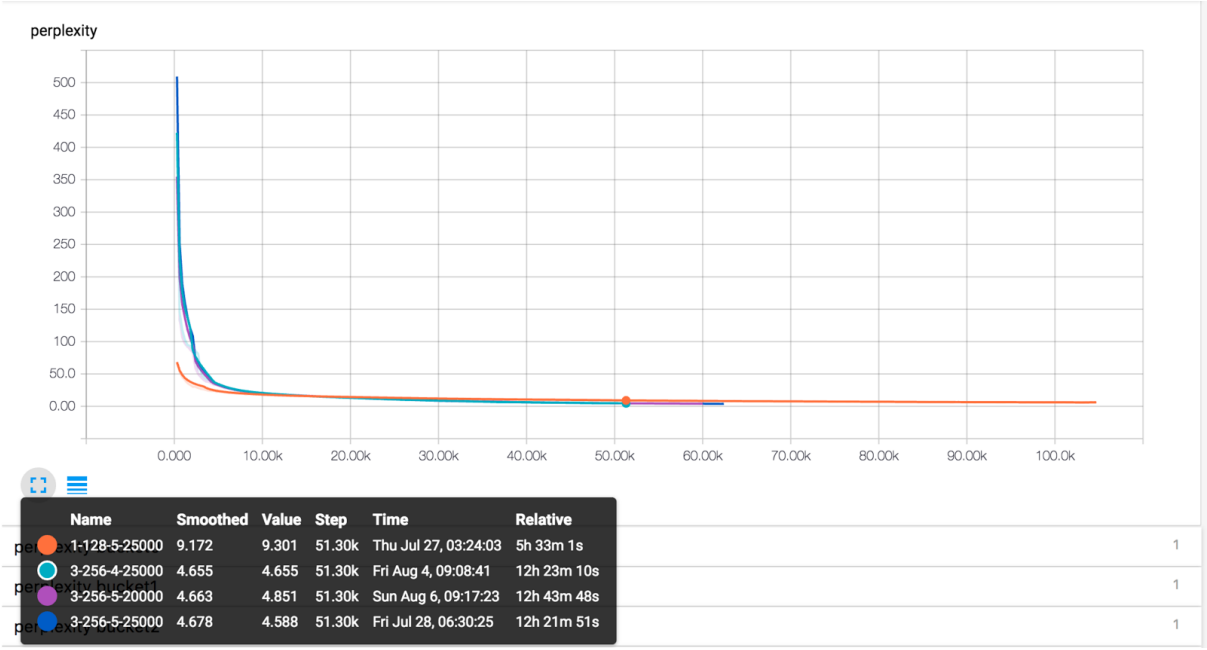
I ran the model on several different configurations, mostly varying the number of layers and the number of units per layer. I also changed the max gradient parameter. Ultimately, I found that a max gradient of 5.0 on a 3 layer, 256 unit network produced the best results, although results were not very sensitive to variations in the max gradient.

Given the large improvement of the 256 unit layers over the 128 unit layers, I would like to have also tested a 512 unit layer, however, my system memory could not effectively run that model.

No. Layers	1	3	3	3
Units per Layer	128	256	256	256
Max Gradient	5.0	5.0	4.0	5.0
Vocab Size	25,000	25,000	25,000	20,000
At Step 51,300	9.30	4.59	4.66	4.85
At 11 hours of training	6.04	5.19	5.36	5.55
Final Perplexity	6.23, 104.7k steps	3.82, 62.4k steps	4.66, 51.3k steps	4.04, 60k steps

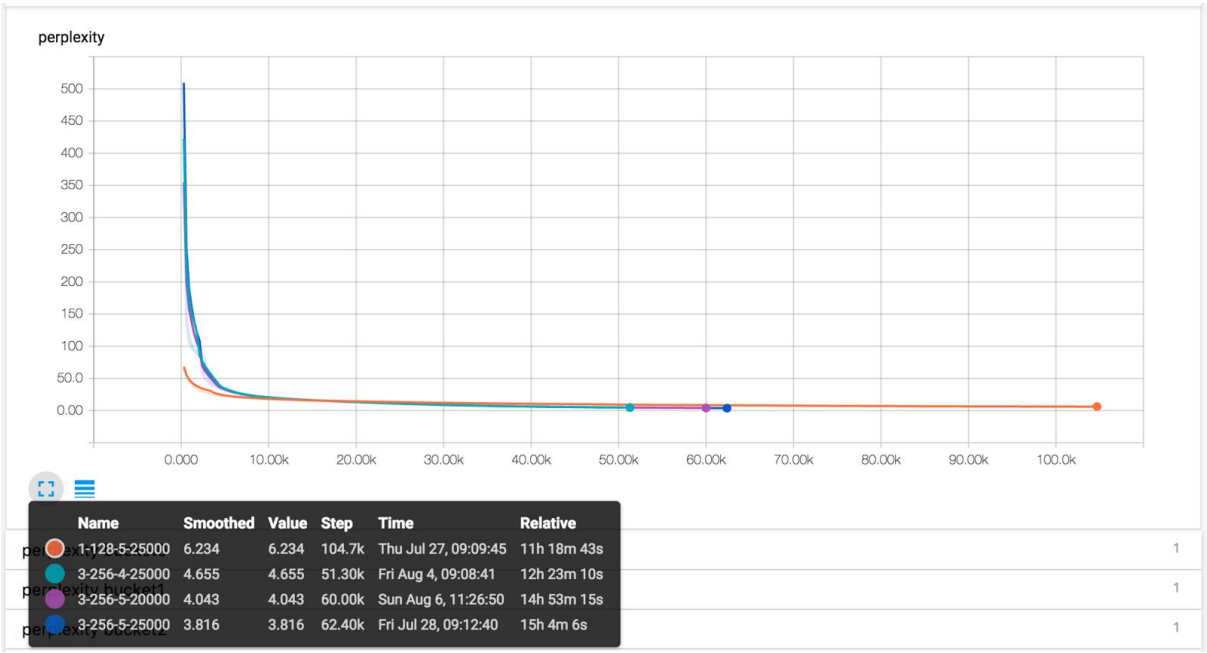
(arbitrary)				

Step 51,300



Note, that we will provide a further corollary to these results in the validation section.

Final Step



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.

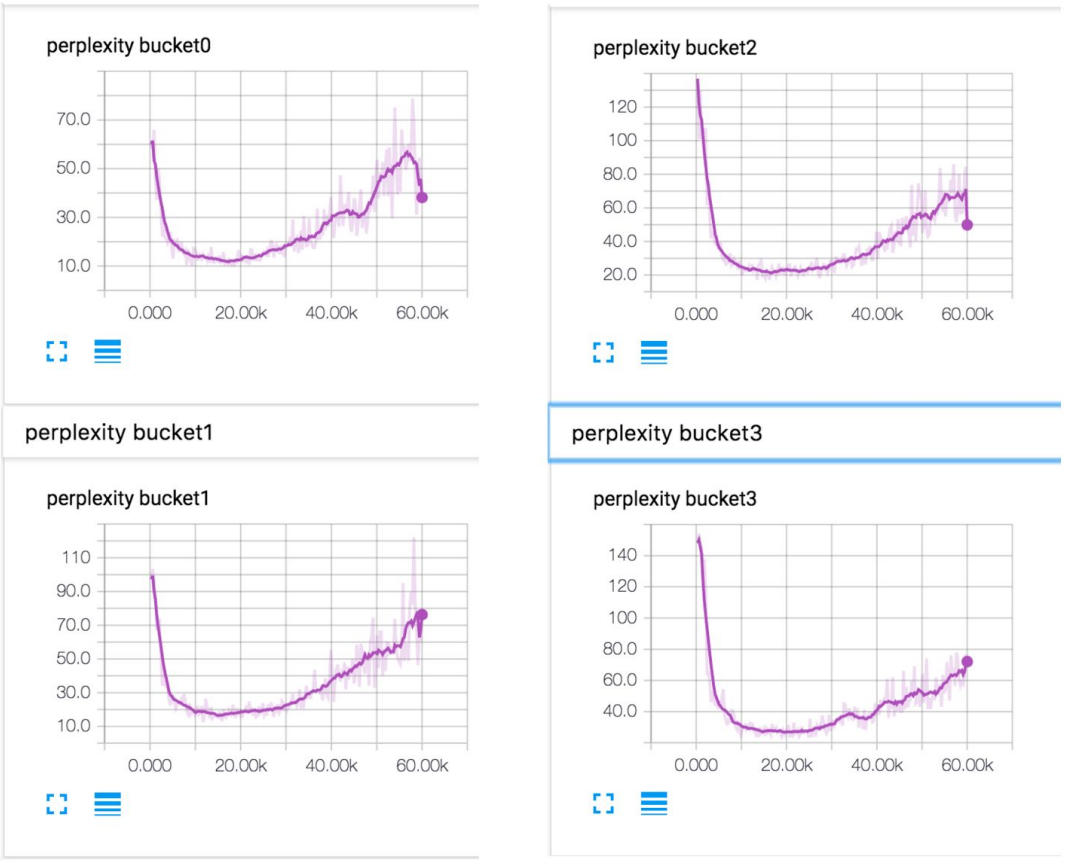
```
> What is climate change?
The Earth’s average temperature has risen by 1.5°F over the past century, and climate scientists estimate it will rise another 0.5 to 8.6°F by the end of this century, depending, in part, on future emissions.[1] That may not sound like much to worry about, since most of 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.

Model Evaluation and Validation

Note: Due to an error in the configuration file, I do not have validation perplexities for the other model configurations.

During the training process in addition to evaluating the training set perplexity, the system also evaluated the perplexity of a test set for each of the included bucket sizes. For the 3 256-unit layer model with vocabulary size 20,000, the perplexities for the four buckets developed as follows:



Interestingly, the minimum occurred around step 16.2k. This suggests further training created overfitting, as validation perplexity began to rise and training perplexity continued to fall.

	bucket0	**bucket1**	**bucket2**	**bucket3**
Max Token Length	input 5, output 10	input 10, output 15	input 20, output 25	input 40, output 50
Perplexity at step 16.2k	15.71	16.75	17.24	27.70

Note, that at this step, the *training* perplexity was 15.77. This demonstrates, that the model was well generalized at this point. While these do not do not match the *final* perplexity of training data, they are within a factor of 3-6, well within an order of magnitude. This suggests that the model was effectively learning and indeed able to generalize grammatical structure, as well as vocabulary significance.

Justification

In the benchmark section, we noted that the lowest training perplexity obtained by competitors in the Github Issues of the chatbot competition (see Benchmark section above), was 1.35, though most were within the range of 4-6. The final training perplexities obtained here were close to these figures, ranging from 3.5-6. Unfortunately, the validations perplexities were not reported in Github, so it is hard to tell if these users were over training their models.

For the 3 256-unit layer model with a 20,000 word vocabulary, we obtained validation perplexities ranging from 15-30 for different buckets. This range corresponds to a chance of 3.3-7% for predicting the target output and about 2.9-4.9 bits per word. Though this percentage may seem low, or inaccurate, evaluating natural language processing is very different than evaluating the accuracy of a more simple classifier. There are a multitude of of words that may be appropriate for the next position in a given sentence.

In the Benchmark section above, we also mentioned that Montemurro et al 2011 reported an entropy of 5.7, or a perplexity of about 52. There are a few reasons why our perplexity is much smaller than these figures:

1. Our dataset was much smaller. We used a dataset based on movie dialogs for both training and validation. The grammatical and vocabularies in movies tend to be simple and short, whereas a dataset representing the entirety of the English language, such as the Brown Corpus, is much larger and more complex, especially if it includes scientific literature.
2. We limited our vocabulary to the 20-25k most frequent word identified in our dataset.
3. We normalized input text to reduce the number of ambiguous characters and words. For instance, we ignored numerals within the text.

V. 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.

Free-Form Visualization

See results section above.

Reflection

This project adequately demonstrated the feasibility of building a chatbot with specialized knowledge, capable of recognizing subject relevant prompts and classifying the appropriate response given sufficient input data and employing a

strategy meta-tokenization to mitigate the complexities of generating intelligible responses for complex outputs.

The improvement section below discusses how an improved architecture that better separates the responsibilities of generalized conversation from specialized input recognition to response category could potentially be designed.

Improvement

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.

Additionally, with regard to generalized conversation, Salesforce found that training a chatbot on conversational data with a model that has already learned translation can provide improvements in performance (Knight 2017). This is likely due to the fact that translation

provides a more well defined task, whereas loss functions for conversation are more ambiguous, given a wider array of responses that may in reality be considered acceptable. This may help the chatbot more clearly identify grammatical structure and thereby significance in input sentences, in turn facilitating the encoder's responsibility of deriving a response.

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.

Brown et al, 'An Estimate of an Upper Bound for the Entropy of English'. http://www.cs.cmu.edu/~roni/11761/PreviousYearsHandouts/gauntlet.pdf, CMU, March 1992.

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

Knight, Will. 'To Build a Smarter Chatbot, First Teach It a Second Language.' MIT Technology Review, July 31, 2017.

Montemurro et al, 'Universal Entropy of Word Ordering Across Linguistic Families.' http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0019875#s4, PLOS.org, May 13, 2011.

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.