John Ketterer

Capstone 2 Milestone Report

Finding a job is a job in and of itself. A determined job seeker spends hours searching for a relevant position, reading about companies and job descriptions, then crafting a cover letter and resume that are fitting to the desired position. Finding a way to maximize your time in job seeking is one way to decrease the overwhelming, discouraging and frustrating process. When a person starts to program they learn ways of automating simple tasks. When a person starts to program artificial intelligence, more complex tasks can be automated and when utilized, possibilities are endless. Without getting too far ahead of myself, we have to start walking before we can run, we will take one particular task in job seeking and automate it.

Presenting a cover letter that is clear, coherent and not overly bland will increase your likelihood in securing an interview. But presenting a cover letter that showcases your skills as a data or NLP scientist in that cover letter is a good way to gain attention and increase that likelihood even further. I think it is important to preface your letter by stating that the following cover letter was completely generated by using AI, if you are pursuing such a role. It would be good to include a personal note about the company and why you want to work there but this is another issue I would like to implement in future experiments. Just like the style of a novel writer or song writer can be perceived and generated by these frameworks used, a sense of personality would be a great addition to include in the next iterations . If not applying to a position that uses AI, a seed with a job title like “Administrative Assistant” would provide a complete cover letter or give you a solid and original foundation to build from.

There are many sites that contain samples and suggestions for “great” cover letters. Most of them are very bland and only contain dry explanations of why a person is a good fit for the position. That is why fine tuning will include some written cover letters that were from my own collection in prior job searches and of course, the most recent. But first, I had to experiment with a set of cover letters that would allow me to develop an understanding of how to utilize these transformers.

Transformers are a neural network sequence transduction model, also known as sequence to sequence, that take an input sequence and transform that into an output sequence. We can use a video of a building being constructed. When a video is set as the input sequence, which is just a collection of images in a particular sequence, the order of the elements within that sequence are processed and following as a result, a model can output what is occuring which in this example is a building being constructed. The same would apply for the video played backwards and after processing the sequence the model would output a deconstruction or destruction. Sequence transduction models are used for a variety of NLP projects to include translation and text generation.

First before explaining transformers a key idea of attention should be explained. It is analogous to being asked to describe a picture you see. If you see a picture of a landscape you would begin to identify trees, mountains and clouds in order to provide an accurate description. This is the same concept as attention within neural networks. Attention focuses on subsets of some information and uses these subsets to create context of the information. Traditionally, Recurrent Neural Networks and the like were used in these sequence to sequence tasks. For an NLP task, a model would try to break down it's input into a single vector which would then be expanded back into multiple vectors. Positions of the words were used within the computation to gain a sense of context for those words. This was a key factor of a sequence to sequence model but had many limitations. Due to the nature of the computation, the amount of time necessary was long and the ability to deal with larger amounts of text was negatively effected.

An effort by Google to improve sequence transduction models based on neural networks that include encoder-decoders introduces Transformers to the world of AI. This introduction has dramatically improved NLP tasks by increasing the quality of performance and allowing a model to be parallelizable, meaning each element is evaluated on every other element within a sequence. The result includes eliminating limits of memory and the ability to use batching for longer sequence lengths. Ultimately, it has had two crucially beneficial effects: 1. Lower computational cost and 2. Significant improvements on model performance. Like all things, transformers do have their limitations but will be discussed later.

As previously mentioned, transformers have impacted language modeling and have resulted in impressive applications of machine learning. One such application was introduced by OpenAI. GPT-2 was created by OpenAI and announced in February 2019 but the company delayed release because of its fear in being used to spread fake news, spam and other sorts of disinformation. After six months, OpenAI finally released the full version after finding no evidence that there was significant use of the model for disinformation. GPT-2 is a text generation system that provides text generation with minimal prompts. The system was trained eight million documents from the web and uses 1.5 billion parameters to generate text when provided a seed to predict what text should follow. Basically. It learns to focus attention on the previous words that are the most relevant to the topic or objective it was given. A common problem with previous language models was solving an issue with multiple meanings of words. When reading a sentence like "A man robbed a bank last Tuesday and has been apprehended by the police." the word bank can have other meanings. Such as, "I was fishing on the river bank and caught a big fish." The same word has a different meaning when used in different contexts. Attention models have been able to solve this problem by learning to focus, or adjusting the strength of, it's attention on previous words and therefore correctly apply weight to other words

within the same context. There are one hundred forty four attention patterns within the model which allow for many linguistic properties to be adhered to and contribute to the models high performance. In this project, the transformers and attention of GPT-2 will be used to generate a cover letter. But in order to utilize a state of the art transformer model for a specific task, data must be obtained to fine tune the parameters of the model to ensure you obtain the output you intended.

The first experiment to be discussed in this paper will try to use the smallest amount of data to discover how the program will react to a given prompt. Because we are using the text generator to develop cover letters, it is interesting to note what the base model will produce when given a prompt (seed): "I want to work for your company because I have all of the skills you are seeking." The result is far from the typical format of a cover letter which is to be expected, but what is interesting is noticing how the attention models are influencing the output text. The generated text reads as follows, "I want to work for your company because I have the skills you are seeking. I am the most talented programmer you could ever hope to hire. I would never leave this company. I could be here for many years. The compensation is great, but not that great. I have to ask you, though, for something... a little favour. I want to tell you I love you. I would never want to leave you for any reason. I don't want you to regret hiring me. You are my job and I want to do this for you. I need you to make me a candidate for the job. I know you will do me the honour of accepting me into the company as a candidate. I promise I will be the best candidate you will ever hire. I understand that you will want to make sure this will go very well and that it won't take you too long. I promise I will not let anything stop me." The first sentence after the given seed is promising. However, after about two sentences, the sentiment and coherence waivers and will definitely make the reader, who is the person that will be hiring you, not consider you for the position. It is impressive that so much text was generated for just a single sentence. One can infer that the generated text had identified the head noun in the sentence and used that as the subject of the paragraph that was generated. The paragraph reads like it has a huge influence from a love story and is probably a result from some of the data scraped from the web during the initial training of GPT-2. This identifies the need to introduce more data which can fine tune and adjust not only the overall coherence of the generated text but also the format in which is written. It is time for the second experiment and to dive deeper into how the model works and how it is utilizing the given data for the text being generated.

Through some searches, a website called jobhero.com was identified for scraping. The site consists of 1670 job titles and a corresponding cover letter sample for that specific job title. Because of the wide range of positions and the consistent formatting of the letters, this data can

be used for a second experiment to investigate how and what will change by the new data's influence. The web crawling framework used to obtain the data was Scrapy. Scrapy was fast and efficient in scraping the text from the jobhero.com website. In less than 5 seconds, it was able to scrape 1663 cover letters from the complete 1670 cover letters on the website and used a csv format. After the data was scraped, Jupyter notebook was used to import the data into a pandas dataframe. For practical purposes I had two columns from the scraped content. The first column includes the title of the job being applied for in the cover letter and the second column included the actual cover letter. Preprocessing of the data was implemented to make the data accessible and clean to ensure maximum utility of the data. First the data had to be split according to new lines. When the data was scraped it included many new line markers and commas used to separate sentences to have a specific format on the web page. The data was structured and cleaned according to sentences and words. Therefore, after cleaning the data to enable analysis empty strings and spaces were cleaned up within the sentences after splitting each sentence. Next, there were some sentences that needed to be eliminated. Had they remained in the data, each text generation would include these sentences. Such as, "Dear <name_of_person>," "thank you...," and "Sincerely, <name_of_jobseeker>." I will include these in the final product to maintain a typical format of the cover letter but in this experiment it is not relevant and will therefore be dismissed for the time being. After these sentences were eliminated from the text, proper punctuation ensured a regular paragraph form for the data. Next, the words were cleaned within the sentences. First, all the words were put into a lowercase form and contractions were expanded. After reading about some of the data collected, GPT-2 does recognize contractions but to ensure error is maximally reduced, contractions, to include an apostrophe 's', were expanded by using a dictionary from aravindpai on Github. Parenthesis were eliminated and any text outside was extracted as well as removing all numbers and special characters. Numbers within the cover letters often referred to the amount of years of experience a candidate had or referred to a quantifiable amount of success they provided in their past at previous companies. Because I wanted to make a model that was generalized and would probably be used more by recent graduates of a data scientist course, these numbers would not prove to be of any significant relevance. Another column was created that did not include stopwords. This assisted in exploratory data analysis and allowed a more focused picture of the corpus.

With clean data, analysis of the text can be performed. One of the first discoveries made during EDA included identifying some frequently occurring words that needed to be added to the stopword list. Some of those additional 20 stopwords within the cover letter included appreciate, consideration, position, and resume. This allowed the fine tuned model to not repeatedly generate these words in the output. Among this initial corpus there were a total of 7021 words within the

corpus excluding all stopwords. There was one word that was left in that occurred almost three times more frequently than the next frequent word; this word was experience. I decided to allow that word to remain in the text because ultimately, this is the main objective of a cover letter: to communicate the skills developed from experience. I will exclude this word from an experiment to decide on the influence on which it has. But to help influence context of the generated text it will remain in this second experiment. The next most frequent words include, skills followed by background, highlights, team and success. Deciding to keep the word experience will contribute to these frequent words and keep the coherence of the cover letter that is generated. The majority of words only occured at a frequency between 1 and 25. The distribution is highly skewed but will help contribute to the context of the text to be generated. After looking at the corpus at a word level, the amount of words within a cover letter was analyzed. An average of 156 words were used within the cover letters. This was found by including all of the stop words within the text but not including the salutation and closing of the letter. This is to limit the amount of text generated and ensure that the generated cover letter does not drag on or lose context as it did in the initial experiment. This is a limitation to be considered during text generation. The average number of words within a sentence was twenty two and the number of sentences within a cover letter was seven. This indicates that sentences often contain a lot of information and can be adjusted within the model to reduce the risk of run on sentences.

After EDA, the second experiment with the current data was conducted and the results were surprisingly decent. The text that included all of the punctuation and stopwords, excluding the cover letter stop words was input into the model. The following is an output from the second experiment: "`I am prepared to contribute to your goals and objectives. From developing and implementing complex software applications to implementing sophisticated statistical models, my background has prepared me to excel in this position. With a solid foundation in statistical analysis and application, along with my ability to collaborate effectively with project teams, I am confident that my skills, background and passion for statistical analysis will significantly benefit your organization.Highlights of my background include:,With my expertise in statistical analysis and application, along with my ability to collaborate effectively with project teams, I am confident that my skills, background and passion for statistical analysis will significantly benefit your organization. I look forward to discussing the position and my qualifications in further detail.<|endoftext|>`" This data has made a great boost in performance compared to the first experiment but a lot needs to be done to continue to improve the model. There is some repetition and when it states that "highlights of my background include:" there is nothing further and the claim is unfinished.

To improve the model, more data will be collected and job descriptions will be used to help develop a more focused text generation. This can be used as a metric and will have even more impact on improving the overall performance.