How does a computer describe a movie?

Lana Elauria

UC Berkeley MIDS W266 Final Project

Abstract

Most of the developments in data-to-text generation have been focused on generating uniform, factual summary sentences from highly structured inputs (e.g., “The Atlanta Hawks (46-12) defeated the Orlando Magic (19-41) 95-88 on Monday” or “The three star coffee shop, The Eagle, gives families a mid-priced dining experience”). But what happens when the structured data input describes something that can be more ambiguous, like a movie? Can a data-to-text model give short descriptions of a movie based on its genre and more specific keywords related to its content? After all, much of the data that a model may need to summarize for different use cases may not be as fact-based as a sports game recap. For this project, I used the popular Movies Dataset from Kaggle to fine-tune Google’s T5 language generation model and evaluate its performance in summarizing movie plots based on the information fed into it from the dataset. Results indicate fair performance on factual summarization of movie production details, but trouble with more detailed plot descriptions.

Introduction

Data-to-text generation is an important field in Natural Language Processing (NLP). The ability to translate raw structured data into more human-readable sentences is crucial as the sheer volume of data available to data scientists continues to grow and data collection methods become more automated. Data-to-text models are used in a wide variety of industries to create short, descriptive sentences for human users. The overall goal of data-to-text generation is a sentence that is easy to parse by human readers, and as such, it is important that the information contained in the sentence is accurate and relevant to the user. Advances have been made in data-to-text generation for recollecting factual information in short summary sentences, such as basketball game recaps or descriptions of restaurants. However, in some use cases, a model may need to generate descriptions of more abstract concepts instead of statistics and facts. Real-world data is messy, and countless applications and industries collect data related to less concrete events or subjects, such as artistic expression, literary meaning, or musical appeal. In these cases, can a data-to-text model generate a text representation of the input that captures those more ambiguous targets? How will those text representations compare to a description written by a human? These are the questions that I investigated in this final project paper, evaluating Google’s T5 language model’s performance in describing movies. A movie is a complex work of collaborative art, with abstract themes and unique characters in every script, but the movie industry is also incredibly well documented, with extensive data on credits and movie content to filter through and feed into a model. I believed this combination of art with data could be an interesting playground to experiment with the new T5 model, so I decided to center this project around training the T5 model to describe movies based on structured input data from the Movies Dataset on Kaggle.

Background

The cutting edge advances in data-to-text generation recently have utilized sequence-to-sequence encoder-decoder architecture (Dušek and Jurčíček 2016) to generate strings of readable text from “meaning representations” that tell the model what to include in the string, as well as where to include it. Improvements have since been made to this architecture, such as inclusion of higher-level sentence planning and content selection (Puduppully, Dong, and Lapata 2019), imposition of pragmatic information preservation techniques and metrics (Shen et al. 2019), and employment of hierarchical encodings for the structured input data (Rebuffel et al. 2020). The cutting-edge in data-to-text generation has focused on honing and improving models that generate text representations of statistical data that is readable by human users. These advancements have mainly been focused on generating factual descriptions of events or places, such as a restaurant or basketball game, and they have been utilizing RNN’s and LSTM’s for their purposes. The models in these papers were specifically built for data-to-text generation, taking inputs from structured datasets and generating formatted summary sentences or paragraphs from scratch.

With Google’s new T5 language generation model (Raffel et al. 2019), the possibility of fine-tuning a pre-trained model for data-to-text generation becomes an interesting avenue for experimentation. The T5 model is pre-trained for various text-to-text tasks, such as machine translation or summarization, with the ability to fine-tune the base models to better suit the user’s specific needs. It is fairly straightforward to fine-tune the T5 model for a new prompt, and it has already been pre-trained on prompts similar to my task (“summarize”, for example). There have been online articles fine-tuning T5 for data-to-text generation, but I was not able to find much academic literature for fine-tuning this sequence-to-sequence model for data-to-text generation, leaving the door open for an enthusiastic graduate student to fill that void for a final project.

Methods

The T5 model's flexibility allows users to fine-tune the pre-trained model on new tasks. By feeding a new prompt in with a text input and providing a few examples of target outputs, the T5 model can be trained to respond to novel prompts specific to the user's goals. I utilized this functionality for this experiment, and I fine-tuned the T5 model to respond to a new keyword, "describe". The data used for this fine-tuning was the Movies Dataset, publicly available on Kaggle. The Movies Dataset contains data of thousands of movies, including title, crew, keywords, and a short plot overview. The Kaggle dataset is sourced from The Movie Database (TMDb), a website that collects production information for movies and television shows. The dataset on Kaggle is split into several files, and I joined and filtered the relevant files to create the data that I fed into T5. Since T5 expects text inputs, I formatted the data in a "fill-in-the-blank" sentence format, with a "describe:" prompt prepended. The final input text contained the movie's title, genre, director, main actor, keywords, year of release, and some keywords that describe the movie's themes. For example, *"describe: John Wick is a action thriller film directed by David Leitch. John Wick was released in 2014. John Wick stars Keanu Reeves. John Wick contains hitman, russian mafia, revenge, murder, gangster, dog, retired, widower."*

Baseline Model: Plot Overviews

The baseline model used short plot overviews within the dataset as target outputs for training. These plot overviews were highly varied in both length and writing style, and they often reference character names or plot points in the films that are not contained in the input data. These baseline plot overviews also do not often reference the title of the film, and it would be difficult for a human who has not seen a certain film to write these overviews with the data used as inputs. However, the plot overviews touch on a movie's main themes or story beats, so there could be words in the overview that are related to the keywords that are fed into the model in the input text. Even though the model may have trouble parsing these plot overviews and finding patterns to learn from them, it is still interesting to evaluate its performance in converting data to text in this way. We will be able to see how much abstraction a T5 data-to-text model can perform, similar to abstractive summarization from other NLP models. The text generated may also give insight into patterns that the pre-trained T5 model has learned about a movie's themes generally, based on the novel sequences it creates to describe it. Some movies have also permeated our culture, so writings about those movies in T5's pre-training corpus may influence how T5 describes movies in general, which would also be interesting to observe.

Improved Model: Wikipedia Descriptions

In addition to the plot overviews included in the Movies Dataset, most movies also have a Wikipedia page, and the first sentence of a movie's Wikipedia page contains a more factual description of its production and themes. The Wikipedia descriptions are often in similar formats, and they usually reference the title of the movie at the beginning of the description. Many Wikipedia descriptions also cite various cast and crew involved in the production, who are also included in the input data from the Movies Dataset. The Wikipedia descriptions tend to be more focused on the facts surrounding a movie's production than the plot overviews from the Movies Dataset, but can also include short descriptions of a movie's plot or themes as well. It seemed as though these Wikipedia descriptions would be much easier for the T5 model to recreate, so I explored using these descriptions as another set of target outputs for fine-tuning the model. As an extension of the baseline, I used the Wikipedia API to retrieve descriptions of about 1,000 movies from the Movies Dataset to use as separate training, validation, and test datasets. The 1,000 descriptions were split into a 500-movie test set and a 500-movie validation set, which I used to fine-tune the T5 model with few-shot training. I used these Wikipedia descriptions to train another T5 model for comparison to the baseline, to see if it was better able to generate factual descriptions of a movie's production rather than more abstract descriptions of its themes and story.

Fine-Tuning T5

To fine-tune the model, I created input-output pairs with the input text generated from the movie data and the target output text. I compared model performance when training on two different training sets; the input text was the same for both sets, and both sets contained the same movie data, but the target output for one training set was the plot overview contained within the Movies Dataset, and the target output for the other was the Wikipedia description of the movie from the Wikipedia API. For a good balance between model performance and computational efficiency, I used few-shot training to tune the T5 model to respond to the "describe" prompt. For each training set, I sampled 15 input-output pairs to train the model in five epochs. Hyper-parameters such as batch size and number of epochs were tuned using a 500-movie validation set sampled from the full dataset. Computational restrictions on the Wikipedia API and the T5 model itself limited the validation and test set sizes, so the model was evaluated on 500-movie samples of the full Movies Dataset.

Evaluation Metrics

The models were evaluated using the ROUGE metric, with ROUGE-1 as the primary evaluation metric. After training, the model was fed text inputs from the test set to generate novel descriptions of several movies. I used a beam search to generate more fluent sentences, and the value of the beam search was tuned with the validation set to balance fluency with relevance of the generated tokens to the actual content of the movies. The best beam search value for these models was 9, erring on the side of fluency over repetition of words from the input text. Each model generated three candidate descriptions for the movies in the test set, and I then calculated the ROUGE scores between the T5 model's descriptions and the descriptions from both the Movies Dataset and Wikipedia. The highest ROUGE-1 F-score between the three candidates was recorded for each movie in the test set, and then the ROUGE-1 scores for the entire test set were averaged together. I calculated the ROUGE scores against both plot overview and Wikipedia description for each model to see if improved performance from one training set also improved performance on the other. Improved performance in describing a movie should presumably reflect in higher ROUGE scores for both target outputs, regardless of training data, since both target output sentences describe the same movie.

Results and Discussion

Figure 1 (included in the Appendix for visual sizing purposes) shows the comparative performance between the baseline and improved models, with varying numbers of training examples for few-shot training during fine-tuning. Improved performance with one target output did not lead to improved performance for the other, however the improved model trained on 15 Wikipedia descriptions was the highest-performing model overall.

Baseline Model Performance

The average ROUGE-1 scores from the baseline model are just over 0.2 for the baseline plot overviews, and around 0.3 for the Wikipedia descriptions. The plot overviews given in the Movies Dataset are incredibly varied, and much of the information used to write the overviews is not present in the input data presented to the model. For example, many overviews include character names, descriptions, and goals left vague enough to create intrigue from a potential viewer. These words are usually not included in the keywords in the dataset, which often indicate higher-level themes or concrete nouns that are contained in the movie. The plot overviews are also highly varied in length and content, which could confuse the model when trying to optimize for specific patterns based on the structured input text. With all of these considerations in mind, it is understandable that the T5 model does not perform well when fine-tuned with these plot overviews as target outputs. A future experiment could use movie scripts instead of production data as the input text in order to create the summaries included in the dataset, but the computational power to process that much text is completely out of the scope of this project.

The generated descriptions from the baseline model were also largely irrelevant to the plot of the movie, most likely due to pre-trained associations between keywords or names referenced in the input data. The generated text still read like fluent English, however they often did not accurately describe basic aspects of the movie's story or characters. These inaccuracies are not fully captured in the ROUGE-1 score, since the same words can be used to describe very different scenarios in different sentences. However, comparing descriptions generated by the model to the actual plot overview from the Movies Dataset reveals that the T5 model often resorts to the most well-known or generic interpretation of the film's keywords, or even nonsensical sequences of text seemingly ignoring the keywords or context of the movie. This could have been due to the beam search hyper-parameter, but running the model with a lower beam-search runs into a different problem. Generating descriptions with a lower beam search parameter inhibits the model's ability to string together fluent sentences; the model ends up repeating text from the input verbatim, which renders the model's generation trivial.

Improved Model Performance

On the other hand, the extended model, fine-tuned on Wikipedia descriptions of each movie, achieves ROUGE-1 scores of approximately 0.15 for the plot overviews, but over 0.4 for the Wikipedia descriptions, significantly increasing the effectiveness of the baseline model when describing production details. The Wikipedia description of a movie is usually more concrete and factual than the plot overview included in the dataset, citing the movie title, year of release, genre, and director(s). These factual data are usually shared by both descriptions, indicating that this T5 model was able to learn how to describe a movie's production details more accurately than the baseline model. Some Wikipedia descriptions contain just these production facts, but some also include extra information about the movie, such as greater themes or a simple explanation of the movie's plot or cultural context. Some Wikipedia descriptions also contain different, more complex relations between cast and crew members (e.g., "written, directed by, and starring..."), or companies that produced the movie. This is important to note, since the fine-tuning picked up on these patterns quite often, generalizing them to more movies where they did not belong. This overfitting was mitigated through hyper-parameter tuning, but remains as an artifact of having only a few examples for the model to learn from. This overfitting could be addressed in future work by experimenting with the level of detail included in the input text, such as the number and specificity of keywords, crew (director, producer, special effects, etc.), and cast roles (possibly including character names as well). Including more of this data could allow the model to pick up more patterns in the Wikipedia overviews, reducing confusion when there are multiple roles associated with one person or references in the description that are absent in the input text.

Conclusion

Overall, fine-tuning Google's T5 language generation model achieved fair performance on factual descriptions of movie productions, but struggled with more detailed summaries of movie plots based on the data included in the input. The model's performance demonstrates the difficulty of creating accurate, holistic representations of art in data-to-text settings: a baseline T5 model generating descriptions of movies often could not learn the nuances of character or themes, but a secondary model trained to recreate Wikipedia descriptions greatly improved the baseline accuracy in summarizing movie production details. Future work can definitely be done to improve both models, either by adjusting or extending the input data or further fine-tuning the model parameters with more computational power. As data continues to be collected and used in new contexts, experimentation with data-to-text generation such as this will prove to be valuable for data scientists and users to create human-readable summaries of whatever that data describes. Grappling with the difficulties of capturing abstract, distinctly "human" concepts, such as artistic expression, in deep learning models will enable us to explore the limits of machine learning and understand what cannot be reduced to embedding matrices.

References

[1] Dušek, O.; and Jurčíček, F. 2016. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 45–51. Berlin, Germany: Association for Computational Linguistics.

[2] Puduppully, R.; Dong, L.; and Lapata, M. 2019. Data-to-Text Generation with Content Selection and Planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 6908–6915.

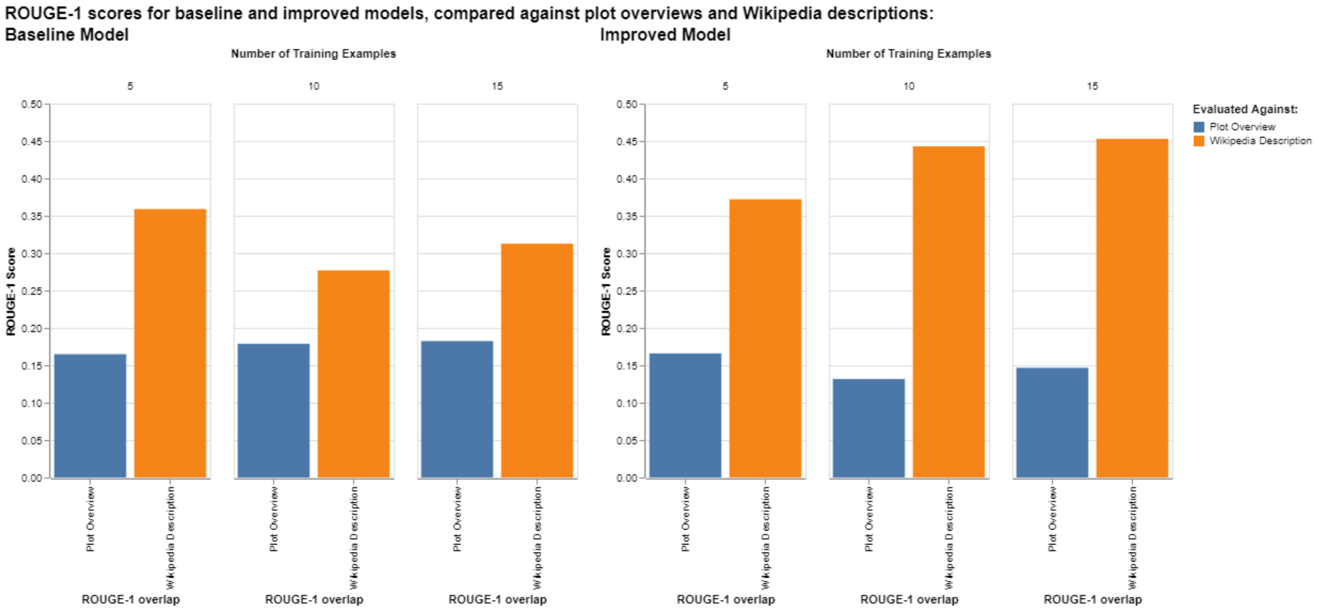
[3] Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

[4] Rebuffel, C.; Soulier, L.; Scoutheeten, G.; and Gallinari, P. 2020. A Hierarchical Model for Data-to-Text Generation. In Jose, J. M.; Yilmaz, E.; Magalhães, J.; Castells, P.; Ferro, N.; Silva, M. J.; and Martins, F., eds., *Advances in Information Retrieval*, 65–80. Cham: Springer International Publishing. ISBN 978-3-030-45439-5.

[5] Shen, S.; Fried, D.; Andreas, J.; and Klein, D. 2019. Pragmatically Informative Text Generation.

Appendix

Figure 1: ROUGE-1 model scores, for baseline and improved models.


ROUGE-1 scores for baseline and improved models, compared against plot overviews and Wikipedia descriptions:
Baseline Model
Improved Model
Number of Training Examples
ROUGE-1 Score
Plot Overview
Wikipedia Description
ROUGE-1 overlap
Evaluated Against:
Plot Overview
Wikipedia Description