Automatic Generation of Formula 1 Reports

Tijmen Krijnen
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.a.krijnen@student.utwente.nl

ABSTRACT

In this work, a new language generation system is proposed. Language generation in the area of sports reports is an active interest of the NLG community. However, there has been no research into the automatic generation of Formula 1 race reports in this area. The goal of the system that will be created is to produce short Formula 1 race reports. The system will create the reports with the help of a deep learning system called GPT-2, which will be finetuned to produce short Formula 1 reports. This research shows that it is possible to generate reports with similar quality of fluency, grammaticality, and cohesiveness as human-written reports. However, it also shows the two biggest pain points for the system, which are the factuality and the repetitiveness.

Keywords

natural language generation, data-to-text, deep learning, transformer models, GPT-2

1. INTRODUCTION

Natural language generation (NLG) and natural language understanding (NLU) are both parts of natural language processing (NLP). NLP is a research field that lies at the boundary of linguistics, computer science, and artificial intelligence, dealing with natural language interaction between computers and humans. NLU is the process of understanding natural language and producing a structured representation, whereas NLG is the process of turning structured data into human-readable text.

NLG has existed since the mid-1960s when Weizenbaum developed ELIZA, the first program to make natural language conversation between humans and computers possible [1]. Further examples of NLG are the generation of weather forecasts [2], patient reports [3], and persuasive fashion product descriptions [4].

There are two types of NLG systems: template-based NLG systems and deep learning NLG systems. A template-based NLG system utilizes non-linguistic inputs and maps these to linguistic structures with gaps, where the data from the non-linguistic input will be used to fill the gaps in the linguistic structure based on rules [5].

The other possibility is to use deep learning to generate natural language. A deep learning NLG system is created by training it on an extensive data set of input-and-output data. The input does not need to be structured data. For example, the input could be images, and the NLG system could be trained to produce a caption for them, or the input could be texts, and the system must learn how to translate or summarize them.

The proposed research will focus on automatically generating Formula 1 race reports; the race reports will be generated using an existing pre-trained NLG system. The research aims to finetune the existing NLG system such that the reports generated are factual, non-repetitive, fluent, grammatically correct, and cohesive.

In this paper, firstly, the problem statement and the research question that it prompts will be described in section 2. Secondly, the related work in deep learning NLG systems will be discussed in section 3. Then, how the research has been performed will be described in section 4. After which the results of the language model and the results of the survey will be discussed in section 5. Lastly, section 6 will conclude the research and will talk about possible future work in this area of research.

2. PROBLEM STATEMENT

Automatic report generation for sports is an active topic of research inside the NLG community. In the literature there are, among others, systems for generating football reports [6], National Basketball Association (NBA) news [7], and minor baseball league news [8]. However, as of now, there is no system for the generation of Formula 1 reports. Consequently, this research will focus on creating an NLG system that can automatically generate Formula 1 reports.

Automatic report generation is an active topic inside the NLG community because NLG has multiple benefits. NLG can help media outlets publish more content and make their content more diverse, e.g., historical coverage. This is possible because NLG is faster and cheaper than writing reports. While it would be most efficient if the generated article did not need any attention, this is not always the case. NLG systems do not always generate quality reports, and sometimes they need to be edited to be published. However, this would still help improve efficiency and save time and money.

Another task which NLG could be used for potentially is live reporting and updating. As soon as data comes in, the language model could generate a minor update for people who cannot watch the event live. Another benefit of automatic reporting is the ability to translate these updates and reports to multiple languages automatically.

35th *Twente Student Conference on IT* July 2nd, 2021, Enschede, The Netherlands.

2.1 Research Question

The problem statement leads to the following research question.

How well is a deep learning NLG system able to generate race reports for Formula 1?

The research question will be answered through the following sub-questions.

RQ1: How factual are the generated reports?

RQ2: How repetitive are the generated reports?

RQ3: How fluent are the generated reports?

RQ4: How grammatically correct are the generated reports?

RQ5: How cohesive are the generated reports?

3. RELATED WORK

This section of the report will discuss the previous research performed and the available literature on this matter.

In 2019 Radford et al. created the generative pre-trained transformer 2 (GPT-2), a language model. A language model is a system which has learned to predict the probability of a word or a sequence of words, given the previous word(s). This means that when provided the start of a sentence, GPT-2 can complete it with a highly likely continuation. GPT-2 has not been trained with explicit supervision for a single task [9]. On the contrary, this system was designed as a general-purpose language model, which should be able to perform all kinds of tasks, such as question answering, translation, text generation, and summarization.

In 2019 Budzianowski and Vulić researched the possibility of using GPT-2 for task-oriented dialogue systems [10]. They proposed a task-oriented dialogue model that only uses text input. The model has not been finetuned further for the purpose of a task-oriented dialogue system. They found that the model performed just as well as a model trained explicitly for this purpose.

In 2020 Lee and Hsiang researched the possibility of using GPT-2 to generate patent claims [11]. They found that the generated patent claims could be coherent on the surface form. However, they also found that a more considerable challenge was how to measure the semantic quality of the generated text.

The work by Mishra et al. shows the possibility of using GPT-2 to automatically generate titles for papers based on the paper's abstract [12]. The model employs three phases; generation, selection, and refinement followed by a scoring function. Using these three phases, the model can generate accurate titles that represent the given text and are semantically and syntactically accurate.

4. METHODOLOGY

The research was divided into three phases. The first phase focused on collecting data, after which the second phase consisted of training and finetuning the language model. Finally, in the third phase the reports were evaluated.

4.1 Phase One: Collecting data

In order to finetune the language model, a considerable amount of input and output data was required. In the first phase, the research focused on gathering the input and output data. For the input data, a dataset or an API with race information was required. This dataset or API needed to supply information about the race. At first the requirements for the API were quite extensive, the API needed to provide information such as race winner, race duration, number of laps, lap times, location, race name, and pit stops. Furthermore, the drivers and constructors standings were also required. However, during the second phase it became clear that not all of this data could be generated. There were a couple of potential API's such as Sportradar[1], Sportmonks[2], api-sports[3], and Ergast API[4]. In the end the Ergast API was used, since it had all of the required data, and it did not require a subscription or any kind of payment. Furthermore, there also exists a wrapper package for the Ergast API called fastf1 in Python which makes it very easy to work with the data. Using this package the data required for the second phase was acquired, the data that was acquired was:

- winning driver's name (e.g. ”Michael Schumacher”)
- second place driver's name (e.g. ”Mika Hakkinen”)
- third place driver's name (e.g. ”Ayrton Senna)
- winning constructor's name (e.g. ”Ferrari”)
- race name (e.g ”Italian Grand Prix”)
- race location (e.g. ”Monza”)
- year (e.g. ”2000”)

For the output data, Formula 1 race reports were required. The reports could be from any source, however, they do all need to be in English. There are of course lots of news sites providing Formula 1 race reports, however, to train the language model, as many reports as possible were required. So, eventhough sites like BBC Sport[5] and the official Formula 1 website[6] provide quality reports, they either only provided a couple reports, or the website was a bit more difficult to scrape. In the end a website called Crash.net[7] was used, it was chosen because it contained Formula 1 reports dating all the way back to the year 2000 and because it was structured such that scraping it would not take too much time. On average each year starting from the year 2000 until 2020 had about 20 races, in the end this resulted in a total of 397 reports scraped. An example of a report from crash.net can be seen in **Example: Crash.net** in subsection 4.3.

4.2 Phase Two: Finetuning

In order to understand what happens in phase two of the research, first two important terms have to be explained. These terms are GPT-2 and finetuning and will be explained in the following section after which phase two will be discussed.

4.2.1 Background

As mentioned in section 3, GPT-2 is a pre-trained generative model that can predict the next word (or sequence of words) in a given sentence. The GPT-2 model was trained unsupervised to predict the next word in a sentence. This means that the model was given a heap of raw text and trained to figure out the statistical features in order to

[1] https://developer.sportradar.com/docs/read/racing/Formula_1_v2
[2] https://www.sportmonks.com/formula-one-api/
[3] https://api-sports.io/documentation/formula-1/v1
[4] http://ergast.com/mrd/
[5] https://www.bbc.com/sport/formula1/results
[6] https://www.formula1.com/en/latest.html
[7] https://www.crash.net/f1/reports

generate text. Pre-trained, in this case, means that the model has already learned how to complete text, and only needs to be finetuned for specific tasks later.

GPT-2 has been trained on a dataset of 8 million general web pages, however, in order for the model to learn how to better represent Formula 1 race reports it has to be finetuned. When finetuning the language model is trained in the same way as it is trained in the pre-training stage, however, it is now trained on a dataset which is specifically targeted on Formula 1. The knowledge that the model has gained from pre-training is kept as much of the language features stay the same, but the language model now learns more specifically how F1 reports are written, including how to use technical jargon, driver names, team names, track names, etc.

4.2.2 Finetuning

The second phase focused on finetuning the language model with the collected data. This is where the collected race data and race reports were used. However, before the race data and race reports could be used to finetune GPT-2 it first had to be encoded and aligned. This meant that for each race a file was created where the race data and race report were combined. The beginning of the file contained the race data, which was encoded with tags, the tags are prepended to the race report such that the language model does not just generat any Formula 1 report, but a Formula 1 report that reflect what happened in the race. After the tags have been added the beginning of a file looks like: ”<|first|>Michael Schumacher<|second|>Sebastian Vettel<|third|> Lewis Hamilton”. After the race data, the beginning of the report was indicated with the tag: ”<|report|>”, this has been done since there are only opening tags and no closing tags. If this tag was not added it might not be clear where the tags end and where the report starts.

After the report tag the actual report starts, then after the headline and the first paragraph an end tag (”<|end|>”) was added, this was done to later be able to truncate the generated report. This way the model could be finetuned on the entire report, while only a shorter report could be generated by truncating everything after the end tag. Since the data in the Ergast API is not sufficient to represent everything that happened in the race and should be in the report, the decision was made to generate short reports consisting of a headline and a brief summary as a brief summary does not require a lot of knowledge to be written. After all the reports were combined with the tags, they were all put into a single file with each race being separated by a newline. They were all put together as this is the way the input has to be structured for GPT-2 to use it.

To finetune GPT-2 Google Colab was used, Google Colab allows anybody to write and execute Python code in a Jupyter notebook environment. Google Colab was chosen because it has a couple of advantages. First of all, it has a more powerful GPU compared to my desktop at home. Secondly, it can be run continously for up to 12 hours, which should be plenty of time for the finetuning process. The created file with reports and tags was uploaded in Google Drive and then loaded into Google Colab. In Python the package gpt-2-simple[8] was used to then import the loaded file into a tensorflow session. After the file was imported GPT-2 was finetuned, different amounts of steps were tried to create the best possible finetuned language model. The different steps that were tried were 1000, 2000, and 5000 steps.

4.3 Phase Three: Evaluating

In the third phase, the research focused on evaluating the generated race reports. To assess the performance of the NLG system, the following dimensions were established (see subsection 2.1): catchyness, factuality, repetitiveness, fluency, grammaticality, and cohesiveness.

To evaluate the quality of the generated reports a survey has been composed. The survey first asks the respondent whether they are a Formula 1 fan. Then the survey shows two headlines one of them written by a human and the other generated by a computer, the respondent is then asked what they think of the headline based on catchyness, repetitiveness, fluency, and grammaticality. After the headline, the first paragraph of the reports are shown to the respondent, once again they are asked what they think of the quality. However, this time the metrics have been changed a little, instead of catchyness, the metric cohesiveness is now used. The metrics have been changed since the metric catchyness is more appropriate for a headline, and the metric cohesiveness is more appropriate for a small report. All the metrics used in the survey are recorded on a 5 point Likert scale ranging from extremely bad to extremely good. Then, the respondent is asked which report they preferred. This is done three times for three different races from the year 2000, the 2000 Australian Grand Prix, the 2000 Brazilian Grand Prix, and the 2000 Belgian Grand Prix.

The human-written reports are taken from the aforementioned website called crash.net. The following is an example of the headline and the first paragraph of a race report from crash.net, for the rest of the report please see the link in the footnote.

Example: Crash.net
”*Schumi wins as history repeats.* Michael Schumacher romped to victory in Melbourne, making the most of a double retirement for the McLaren team to lead home a Ferrari one-two. Michael Schumacher romped to victory in Melbourne, making the most of a double retirement for the McLaren team to lead home a Ferrari one-two.”[9]

For the computer generated reports the data for the race was fed to the language model as a prompt, which then generated five race reports of which the best one was selected and used in the survey. The following is an example of a race report generated by the finetuned language model.

Example: GPT-2
”*Australia GP 2000 - Schumi back in the saddle.* Michael Schumacher won the inaugural Australian Grand Prix, following Ferrari team-mate Rubens Barrichello’s (15th) defeat by the Spaniard at the Nurburgring on lap two. Michael Schumacher won the inaugural Australian Grand Prix, following Ferrari team-mate Rubens Barrichello’s (15th) defeat by the Spaniard at the Nurburgring on lap two.”

5. RESULTS

The evaluation of the generated reports is done in two ways. Firstly, a qualitative reflection was done, here the factuality of the reports was checked. Secondly, a survey was held to determine what people think of the generated reports in terms of repetitiveness, fluency, grammaticality, and cohesiveness and how they stack up against human-written reports.

[8] https://github.com/minimaxir/gpt-2-simple

[9] https://www.crash.net/f1/race-report/35433/1/schumi-wins-as-history-repeats

5.1 Evaluation

One of the problems with generating reports is that one report might be really well written in all aspects, however, the next might be very badly written. There is no guarantee that the quality of all generated reports will be equal. So, somebody will always have to check the generated reports to find out if the quality of the report is up to par.

For the finetuning different amounts of steps were tried to see if they resulted in different quality of generated reports. As mentioned in subsection 4.3, 1000, 2000, and 5000 steps have been tested with GPT-2. However, they all produced rather similar quality of reports and there was no real disparity in the generated reports. This does not mean that there is no difference between different amounts of training, however, for these specific amount of steps the quality was very similar.

RQ1: How factual are the generated reports?
To answer this subquestion, a quick look at a couple of generated reports is all it takes. In example 1 down below it becomes clear that the model might have learned how a race report is written, however, it does not know what actually happened in the race, what racing is, or what the driver's nationality is. In this example Mika Hakkinen won the Belgian Grand Prix and is called a Brazilian, however, Mika Hakkinen is actually Finnish.

Example 1
"*Belgium GP 2000 - Hakkinen's runaway.* Brazilian star Mika Hakkinen defied the experts to claim his second career win at the Belgian Grand Prix and snatch the 1-2 advantage inside the top eight. Brazilian star Mika Hakkinen defied the experts to claim his second career win at the Belgian Grand Prix and snatch the 1-2 advantage inside the top eight."

Since it does not know what actually happened, it can often generate reports which do not tell the truth. Another example where the report was not factual is example 2, eventhough the input prompt clearly stated that Sergio Perez had won the race, the language model generated that Sebastian Vettel had won the race since this was way more likely to happen according to history. Furthermore, again the driver's nationalities are incorrect, as Sebastian Vettel is German and not Azerbaijani as the report indicates by generating home soil, and Sergio Perez is Mexican and not Brazillian.

Example 2
"*Vettel wins Azerbaijan GP as Perez star leaves with injury.* Sebastian Vettel marched to his fifth Azerbaijan Grand Prix victory by being the last driver to finish on home soil after a late safety car period and a shock turn of pace from Perez. Sebastian Vettel celebrated his fifth place in Bashkortostan F1 team Red Bull's fifth victory of the season, as a late safety car period and a shock turn of pace from Perez saw the Brazilian escape without any damage."

5.2 Survey Results

From analysing the survey results it became clear that the quality of the generated reports is very different between metrics. The metrics catchyness, fluency, grammaticality, and cohesiveness were some of the strong points for GPT-2, however, GPT-2 scored very poorly on repetitiveness.

It was decided to not analyse the data from the F1 fan question due to the limited amount of data. The catchyness of the headlines was not one of the subquestions, nevertheless, it is a very important metric for a headline. From the survey it became clear that on this metric GPT-2 did perform decent, sometimes very interesting titles were generated, but other times the titles were very simple, an example of a simple headline is this headline from the **Example 1** in subsection 5.1: "*Belgium GP 2000 - Hakkinen's runaway.*".

	Crash.net	GPT-2
Catchyness	3.48	3.39
Repetitiveness	3.45	3.58
Fluency	3.55	3.82
Grammaticality	3.21	3.64

Table 1: Mean scores for the headlines

	Crash.net	GPT-2
Repetitiveness	3.39	2.21
Fluency	3.45	3.61
Grammaticality	3.39	3.76
Cohesiveness	3.67	3.67

Table 2: Mean scores for the reports

	Crash.net	GPT-2
Repetitiveness	3.42	2.90
Fluency	3.50	3.72
Grammaticality	3.30	3.70

Table 3: Mean scores for headlines and reports combined

RQ2: How repetitive are the generated reports?
The repetitiveness is one of the weak points of the system (2.90, see Table 3), sometimes the model generates the same sentence twice as can be seen in **Example 1** in subsection 5.1. This is also what probably caused the low score for repetitiveness for the generated reports as can be seen in Table 2. However, the score for the repetitiveness of the headlines is actually very comparable to the human-written headlines of crash.net.

RQ3: How fluent are the generated reports?
The fluency of the reports is a strong point for the GPT-2 system (3.72, see Table 3). The fluency of the generated reports was actually even perceived a little bit better than the fluency of crash.net reports. However, this is not a significant difference, but it shows that the generated reports and headlines are very good in terms of fluency and can be compared to the human-written reports.

RQ4: How grammatically correct are the generated reports?
The grammaticality of the reports is another strong point for the GPT-2 system (3.70, see Table 3). On this metric GPT-2 also scored a little bit better than crash.net. However, again this is not a significant difference, but the grammaticality of both the generated headlines and the reports are very comparable to the human-written headlines and reports.

RQ5: How cohesive are the generated reports?
The cohesiveness was not measured for the headlines, as the headlines are very short and thus cohesiveness would not be a good metric for the quality of the headlines. Nevertheless, the generated reports scored exactly the same as the human-written reports (3.67, see Table 2). Which also indicates that the cohesiveness is very comparable to the

human-written reports and can be seen as another strong point for GPT-2.

6. CONCLUSION AND FUTURE WORK

This research aimed to finetune an existing language model called GPT-2 to generate Formula 1. This research has shown that it is possible to generate reports using a pretrained language model after it has been finetuned. However, it also shows that it is difficult to produce factual reports, as the language model does not really know what racing is or what happened in the race.

In the end, the main research question: "how well is a deep learning NLG system able to generate race reports for Formula 1?" can be answered by looking at what is important for a good F1 report. For a good report a primary condition is that the report has to be factual, otherwise the reports are just fictional stories that did not happen. As it is not possible to create factual reports with the current system it can be concluded that the current system is not capable of generating race reports for Formula 1.

However, there are also some positive takeaways from this research. Mainly that generating reports using a finetuned language model is very feasible and that the quality of these generated reports is very comparable in terms of fluency, grammaticality, and cohesiveness.

Due to a restricted time frame, this research had some big limitations. One of the big limitations was the size of the training data. If this research were to be continued, future work could focus on gathering more race reports, such that the language model could be finetuned further with a bigger training set. The training data set could be expanded with reports from multiple websites mentioned in subsection 4.1. Furthermore, there could be more experiments with more information in the prepended tags such that the model could possibly be conditioned to generate factual reports.

7. ACKNOWLEDGMENTS

The author would like to thank Lorenzo Gatti for providing guidance and feedback throughout this research.

8. REFERENCES

[1] J. Weizenbaum. ELIZA — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 26(1):23–28, jan 1983. (Back to section 1)

[2] E. Goldberg, N. Driedger, and R. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53, apr 1994. (Back to section 1)

[3] F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, may 2009. (Back to section 1)

[4] V. Munigala, A. Mishra, S. G. Tamilselvam, S. Khare, R. Dasgupta, and A. Sankaran. PersuAIDE ! An Adaptive Persuasive Text Generation System for Fashion Domain. *CEUR Workshop Proceedings*, pages 1–9, 2018. (Back to section 1)

[5] K. van Deemter, M. Theune, and E. Krahmer. Real versus Template-Based Natural Language Generation: A False Opposition? *Computational Linguistics*, 31(1):15–24, mar 2005. (Back to section 1)

[6] C. van der Lee, E. Krahmer, and S. Wubben. PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104, Stroudsburg, PA, USA, 2017. Association for Computational Linguistics. (Back to section 2)

[7] C. Li, Y. Su, J. Qi, and M. Xiao. Using GAN to Generate Sport News from Live Game Stats. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11518 LNCS, pages 102–116. Springer Nature Switzerland, 2019. (Back to section 2)

[8] B. Mullin. The Associated Press will use automated writing to cover the minor leagues. *Poynter Institute*, 2016. (Back to section 2)

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI*, 2019. (Back to section 3)

[10] P. Budzianowski and I. Vulić. Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *arXiv*, pages 15–22. Association for Computational Linguistics, jul 2019. (Back to section 3)

[11] J.-S. Lee and J. Hsiang. Patent claim generation by fine-tuning OpenAI GPT-2. *World Patent Information*, 62(June), sep 2020. (Back to section 3)

[12] P. Mishra, C. Diwan, S. Srinivasa, and G. Srinivasaraghavan. Automatic Title Generation for Text with Pre-trained Transformer Language Model. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 17–24. IEEE, jan 2021. (Back to section 3)