Creation of a sport newspaper articles generator as part of the study of automatic text generation by neural networks

Abstract

This report concerns the achievement of a sport newspaper generator. First, it details how the GPT-2 language model works. The sports generator is then made thanks to a transfer learning by fine-tuning on GPT-2. To measure the effectiveness of transfer learning, a first evaluation is proposed thanks to a logistic regression. An one-class SVM is also mentioned as part of the evaluation. The results show an improvement in the generated texts thanks to transfer learning. However, biases in learning indicate that evaluation methods need to be improved. The results should also be analyzed accordingly.

Keywords: text generation, sport newspaper generator, GPT-2, transfer learning, transformer architecture

Claire Robin

Internship of December 2019 to Eura Nova - Belgium

Licence 3 - Mathematics, Physics, Chemistry, Computer science-

Internship supervisor : **Thomas Peel**

Thanks

I thank Eura Nova for giving me the opportunity to do my internship with them. I thank them for the warm welcome which made this internship extraordinary. I thank Amidex for the scholarship that allowed me to do this internship. I thank Remi Eyraud for his confidence. I thank Maryam Keras and Louise Marinho for their rereading. I also thank Malian De Ron for teaching me how to use dockers correctly. Lastly, I thank Thomas Peel for all the knowledge transmitted, for his availability and for the richness of this internship.

Table des matières

Introduction

0.1. Internship objectives

The aim of this internship is to create a sport newspaper articles generator. From data or a sentence in inputs (as the score, the teams, the place etc.), a neural network should produce a text which must reproduce a sports newspaper event until potentially deceiving a human. For this, the generator must learn the structure, the grammar, the lexical fields used and the formulation of the sentences typical of a sport newspaper to produce a similar text event until potentially deceiving a human.

To address the problem, I started by studying language model with GPT-2, then I tried several approaches to design a sport news generator, especially with a Transfer Learning approach by fine-tuning of GPT-2. Lastly, I carried out an evaluation of my solution thanks a logistic regression with a TF-IDF trying to learn a classifier to detect fake news. I also considered other metrics can improve the effectiveness measurement of the Transfer Learning.

0.2. Eura Nova

Eura Nova is an Information Technology company in whose expertise is artificial intelligence and software architecture. The company has a team dedicated at research and offers its services in consulting and products. In particular, they use the latest advances in AI to meet the needs of their customers.

1. Language model studies

I started studying the state of the art in text generation -which belongs to the Natural Language Processing (NLP) domain- with the papers of two language models GPT-2 [6] and Grover [8]. Then, I carried out extensive researches to understand many notions of NLP presented afterwards.

1.1. Natural Language Processing - NLP

Natural Language Processing -NLP- is a component of AI which concerned interaction between computers and human natural language. It aimed to create an algorithm able to interpret the natural language of human people. Tasks of speech recognition, translation, natural language understanding, summarization and language model are in the hearts of research in NLP.

1.2. Language model

The language model is a probability distribution over sequences of words. A language model is defined by :
A set of words, $w_1, w_2, ..., w_n$ (usually set of world of a language),
All sentence $W = w_1, w_2, ..., w_m$ with $m \leqslant n$, we associate the probability :

$$P(W) = P(w_1, w_2, ..., w_m)$$

The most famous algorithm using a language model is probably smartphone keyboards that suggest the next word based on what you've previously typed.

Significant advances have been made possible thanks to the use of neural networks. This technology allows to better represent the mechanisms at the heart of a human's understanding of a text, like the focus in the most important words in a sentence.

1.3. GPT-2, an impressive language model

GPT-2 is a language model which generate text from a sentence in input. His architecture is based on the Transformer architecture[7], a particularly efficient architecture in the context of the translation task. The elementary structure of GPT-2 is the Decoder block [4]. A decoder block is made up 2 layers, a Multi Masked self-attention and a Feed-forward Neural Network. For GPT-2, two normalization layers are added as in the figure. 12 Decoder block are stacked with a softmax in output to form the GPT-2 model.

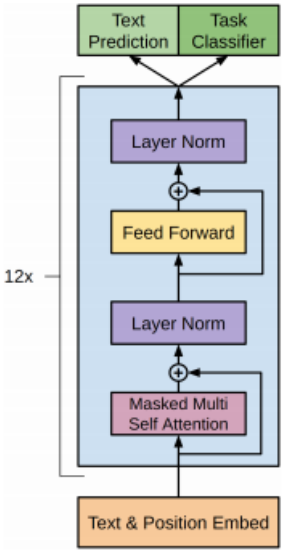

FIGURE 1: Architecture of GPT-2 [5]

1.3.1. Masked self-attention

The masked self attention process is identical to the self attention process with the words in the sentence after the word studied that are hidden. Self-attention mechanism shows the relationships between words within a sentence. It is defined as :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The self attention mechanism take three vectors, Q for query, V for value and K for key. The query is a representation of the current word and value vectors are actual word representations. The queries form a basis of d_Q dimension space.

Each key vector is linked to a word, it is obtained by training of network. The dot product following by softmax between query and key give a score which represent the importance of relationship between the two words.
At the end of masked self attention, we keep values multiplied by score for each query and then we add them up to obtain a context vector associated of the query.
In other words, we represent the query q by his context vector C_q where :

$$C_q = \beta_1 v_1 + \beta_2 v_2 + ... + \beta_n v_n$$

With β_i the scores and v_i the values.

The multi-heads attention is the concatenation of 12 self-attention mechanism -called attention head. It allows to focus -for a same query- in different positions in the text. Thus, each head give a different representation sub-spaces, this means the words with a great score will not be the same depending on the head. The output goes to the next layer.

1.3.2. Position-wise Feed-Forward Neural Network

A Feed-Forward Neural Network is just a network wherein the connections between nodes do not form a cycle. The information moves only one direction, forward. The Position-wise Feed-Forward Neural Network allows to encode the output of multi-heads attention, x (the context), into higher-level representations, it is defined as :

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

The position-wise FFN sub-layer is applied to each position separately and identically.

Finally, the final layer -a fully connected neural network- followed by a softmax takes in input each set of new tokens given by each decoder block. Indeed, each decoder block "understood" the sentence differently and offers a set of potentially different next words. The softmax gives a probability of occurrence for all the words in the dictionary.

During the internship, I used the 117M parameters model, the smaller model, in the order to reduce the calculation in the experiments phase.

2. Achievement of sport news generator

2.1. Generation by GPT-2

To carry out the sport generator, I first installed GPT-2 and I generated my first texts produced by neural network. I had to use Docker and to master Unix commands since I was working on one of Eura Nova's servers, it was at first little bit laborious. With a sentence in input -in italic in the example-, GPT-2 generate the rest of the text. We observe the high quality of the text generated by GPT-2 and the respect of the subject in the begging of the text, but GPT-2 only keeps the subject for a few lines and generates quickly the token <|endoftext|> which causes a change of subject. We search to improve the generation to obtain a complete sports article.

PSG won the football match against OM 3 points to 2, thanks to 2 goals from Mbappé and a penalty from Griezmann. The match was held on Saturday at Saint-Etienne's Nestor stadium. Mbappé scored from a penalty in the second half to increase the score to 2-1. Bruno Gibbs added a second goal from the penalty spot just before half-time, and the final whistle was whistled in the second half. Pro-Brazil coach Jesus dos Santos said the game had a "special meaning" to the nation. "For us, it's a special game with very special meaning and significance," he said. "We heard that a second goal scored in the second half was a penalty on the final whistle." The game is on TV on 8 July at 14 :00 GMT and will be streamed live on 2 August on the Pro-Brazil website.<|endoftext|>CHICAGO (AP) — The Army has seized a military vehicle and a car belonging to a fugitive indicted on charges of running a prison run by four former inmates and four members of the same family, according to four people with direct knowledge of the investigation. The seized vehicles were seized Thursday at a site near the U.S.-Mexico border where they were being processed by the agency. The vehicles were worth 1$million$to1.1 million, the people said. The Army has been looking for a former inmate accused of running a prison run by four former inmates and four members of the same family, who were arrested in 2007 in a shootout with police in the U.S.

Generated text by GPT-2 without fine-tuning

2.2. Generation by GPT-2 finetuned

During my summer internship I had make a Transfer Learning by fine-tuning a pre-trained network -VGG-19- as part of an attribute classifier. As we are in a similar case I proposed this solution to specialize GPT-2 on the generation of sport news. This consists of starting from a pre-trained network, here GPT-2, and re-training it on a specific data set. In this way we teach it a new task (transfer learning) by optimizing his parameters (fine-tuning) so that he generates texts as close as possible to the training data-set. However during my summer internship I worked on convolutional neural networks. I had no idea how to do this work on a network like this GPT-2 which is a stack of decoders. I have finally used the implementation of the data scientist Max Woolf [1] who propose to not modify the structure and only to retrain the network or to retrain specifically the transformers. I trained the network with a data-set of English news articles dealing with sport containing 150 000 texts[2]. This data set had the advantage of being large enough to allow training without over-fitting. During training I varied the following parameters :

— *batch_size* The batch size defines the number of samples that will be propagated through the network before gradient back-propagation (mechanism witch change the weights of a neural network taking into account of the loss function in the heart of the learning phase). Usually batches size turn around 256 in order to provide the network a better generalization, our batch is constrained to 2 because of the server memory. The consequence to a batch so low is a long time of convergence for our network, but in a specialization task, witch represent a small training, it is enough.

— *step* The number of steps (also call epochs) is the number of time all data are propagated though the network. it is detailed in the next subsection.

2.2.1. Step

I first did a first search for a good step value by looking at the results generated for a range of step values. The input sentence of all texts was *"PSG won the football match against OM 3 points to 2, thanks to 2 goals from Mbappé and a penalty from Griezmann."*. For five step the first results are promising as the text generated shows many typical elements of a sport newspaper such as interview, language, time, score, team, sport celebrity.

> *PSG won the football match against OM 3 points to 2, thanks to 2 goals from Mbappé and a penalty from Griezmann.* However, even though the game's results were not as impressive as expected, the PSG players and coaches were given ample time to assess the game's performance. As a result of the PSG's win, after the game, the PSG players and coaches got to talk the players to discuss the game. The PSG coach (Robi) said : "We had a good game, so we can always work on our technical quality. "We were really good in the second half, but we lost 2-1, so we need to take this as a win. "The game is a big game for us, because our other players proved to be faster. "We are looking forward to the game, but we have to keep our focus for the next game. "We are going to get the result for us. I know we are going to get the win. But I don't think we can win every game, because we have to do everything we can." Boden, meanwhile, went on to say that he was very happy with the results of the game, and he is sure that his team will be good too.

Generated text by GPT-2-FT with 5 epochs

For more than a few tens of steps we observe a large number of repetitions which I interpret as over-fitting. An example generated by 50 epochs is in annex 3.3.

During the construction of data-set of generated texts -detailed in the appendix 3.3-, we used the first line of texts of the human data-set in input and this latter are less explicit. The words are rarer and the results are less impressive. But, comparing for same input GPT-2-fine-tuned (abbreviated below GPT-2-FT) text and GPT-2 text, we note an improvement. Indeed, there are many repetitions in the texts of GPT-2-FT but the improvement of the generation shows that network learns and it is less lost face to sport rare words. I initially interpreted the first repetitions as over-fitting but they may be caused by a context too far from learning. As the context is new, the probability of occurrence of news words is very low and and we observe artifacts and repetitions. I trained then the network with 30 epochs. The results seem less good, with more repetitions, however the quality depends on the input sentences and the texts are heterogeneous. We need a rigorous comparison and valuation method. The analysis of results and valuation method are detailed in section 3.

2.3. GAN with Grover

A second solution was considered to make the generator of sport newspaper. The solution was to make a Generative Adversarial Network -GAN- with GPT-2 as generator and the discriminator of Grover which determine for each text if it is written by a human or a machine. I used an

implementation of Grover, which contain Grover's Generator and Grover's discriminator. Studying the paper, we discovered Grover's discriminator was trained (in a GAN) against Grover's Generator but it was also trained against GPT-2. On the one hand the results against GPT-2 were worse, on the other hand, nothing indicated re-trained Grover's generator and GPT-2 in specific data-set will give better results for our task. Lastly, many errors have appeared when using Grover. I therefore did not explore this solution further.

3. Metric

The first results expose the question of evaluations of our results. We want to measure how much texts generated by GPT-2-FT are closer to sport news than whose generated by GPT-2. In order to do so, we can compare the rate of generated text by GPT-2-FT classified as "human sport news" by a classifier against the rate of GPT-2. We started by using a logistic regression to differentiate the generated texts and the human texts.

3.1. TF-IDF with Logistic Regression

For the logistic regression, we will compare words distribution. We assume that the word distribution of a sports article is characteristic with an over-representation of sport words. We hope that from this logistic regression can separate the generated texts from human texts.

As the words that appear the most in the text are the determinants, the adverbs and the commons verbs (as to be, to have etc.) measure the frequencies of occurrence of all words in a data-set is irrelevant. Instead, we will focus on the rare words -words that have a low probability of appearing in a text. We can employed for each text the Term Frequency–Inverse Document Frequency - TF-IDF.

3.1.1. Term Frequency–Inverse Document Frequency

The TF-IDF is defined as :

$$TF\_IDF(t, d, D) = tf(t, d) \cdot idf(t, D)$$

Where :

— the tf (for term frequency) is the number of times a term occurs in a given document

— The idf (for inverse term frequency) of a given term (t) in a set of documents (D) is the logarithm of the number of text (N) by the number of text where the word t appears (n_t) :

$$idf(t, D) = \log\left(\frac{N+1}{n_t+1}\right)$$

So, idf tends toward zero when n_t tends toward N and idf tends toward log(N) when n_t tends toward 1, the common words will have a small score while the rare words will be put forward,

especially if its appear more than once in the same document. Constants 1 are added to prevent divergence.

3.1.2. Logistic Regression

The logistic regression is an algorithm commonly used in the case of binary classification. Our starting hypothesis is that there is a correlation between occurrence of rare words and an human sport newspaper. The details of a logistic regression is in the appendix 3.3. I use Sckit-learn for the code, it is available in the appendix 3.3.

The data-set of training is made up of 600 texts, half Webhose texts for human class, a quarter GPT-2-FT with 5 epochs texts and a quarter GPT-2 texts for generated class. The construction of the training data-set is detailed in the appendix 3.3. I shared the training data in order to set aside 25% of the data-set to constitute a test data-set. After training, we evaluate the learning on the test data set. We observe a score of 93 % of good prediction.

first line	GPT-2-FT 5 epochs	GPT-2-FT 30 epochs	GPT-2 (No training)
webhose	29 % ±7%	43 % ±7%	6 % ±7%

TABLE 1: **Texts rate classified as human with first training data-set**

The results are very interesting. Firstly, we observe the fine-tuning works, the texts generated by GPT-2 classified as human are in the order of the precision, while a third of generated texts by GPT-2-FT are considered as human. The result of GPT-2-FT with 30 epochs is very encouraging but it benefits of a bias since the training data-set does not contain example of this type, I must generate a new training data-set with generated texts by GPT-2-FT with 30 epochs. When we look the texts generated by GPT-FT we observe a lot of artifacts, however the sentences in input of Webhose.io are often off-topic. I use a second data-set (for the training data-set and test data-set) the bbcsport football [3], which does not contain off-topics.

first line	GPT-2-FT 5 epochs	GPT-2-FT 30 epochs	GPT-2 (No training)
webhose	18 % ±11%	28 % ±11%	6 % ±11%
bbcsport football	14 % ±11%	26% ±11%	10% ±11%

TABLE 2: **Text rate classified as human with second training data-set**

Our network seems to better detect the generated texts, the trend is confirmed with a better score for GPT-FT with 30 epochs. There is no significant difference between the two data-sets.

These results must, however, be qualified. Firstly, the score obtained on the test data-set means the classifier has learned correctly and it is able to differentiate generated text from human text in most cases. But, when we look the GPT-2 30 epochs text classification, we observe text classified

as human with lot of artifacts and good quality text classified as generated. The simple addition of '/n' in a text allows its classification as human. This means that our classifier has learned bias. To improve it, we need to build a more heterogeneous data-set with more good quality generated text. Indeed, a significant part of the texts generated from the training data set are of poor quality with many artifacts. But faced with the time necessary to generate new data-sets, we turned to an alternative classifier based on the detection of anomalies thanks to the one-class SVM.

3.2. One class - SVM

I was interested in the one class SVM in order to have a metric without training data-set containing texts generated that take a long time to produce.

An one-class SVM search an hyper-plan which separates in two categories our data, the inliers and the outliers. The training is made in the human text and the generated text are treated as anomaly. the code is available in the appendix 3.3. However I ran out of time to explore this track, the results obtained cannot be interpreted because I did not find the appropriate parameters. With 11,000 texts in training, the classifier displays an accuracy of 87% and indicates the following results.

first line	GPT-2-FT 30 epochs	GPT-2 (No training)
webhose	52 %	71 %
bbcsport	62 %	81 %

TABLE 3: **Text rate classified as human by one-class SVM**

3.3. Achievement of a discriminator

A solution studied was to create a discriminator from GPT-2. The idea consist to modify this one for delete the generator section and keep only the model, add a layer of two output with a softmax and re-train GPT-2 modified. Like this, we create an discriminator. Our discriminator use benefits frameworks of GPT-2, with self-attention mechanism for focus in the most important words in the text and intuitively we can imagined that a discriminator with a framework closer whose the generator will be better (intuition proved within Grover).
This track seems very promising, it must be continued. It was however particularly complex at my level.

Conclusion

During this internship I discovered the text generation problematic topic, I had to automatically generate sports news. For this purpose, I started studying state-of-the-art of language model then I carried out a Transfer Learning by fine-tuning of GPT-2. Then, to measure the efficiency of this task I carried out a classifier thanks to a transforming the text into a TF-IDF representation with a Logistic Regression. The obtained results must be improved, we observe a huge variation in the quality of the generated texts. A better training data-set with a GAN constitute an interesting track to improve the quality of texts generation, for that I need to improve my skills in neural networks.

In the longer time, we could modify the input of the algorithm so that it takes data -like the teams and the score- instead of a sentence. This way it will no longer depend on an input sentence. Moreover, it would have constraints on the data instead of just generated text following the input sentence. We can also imagine training it on match descriptions. He will thus learn the progress of a match and perhaps the rules and the teams. In this way his texts will gain consistency.

During this internship I also discovered Tensorflow, regression methods and SVM methods, I learn to use Docker and Tmux. It is thanks to the knowledge gained at LIS that I was able to make connections and propose solutions. It was also a first step in the IT business world.

Bibliography

Références

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

[2] `https://webhose.io/free-datasets/sports-news-articles/`.

[3] `http://mlg.ucd.ie/datasets/bbc.html`.

[4] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. *Generating Wikipedia by Summarizing Long Sequences*. CoRR, abs/1801.10198, 2018.

[5] Alec Radford. *Improving Language Understanding by Generative Pre-Training*. 2018.

[6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. *Language Models are Unsupervised Multitask Learners*. 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention is All you Need*. In NIPS, 2017.

[8] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. *Defending Against Neural Fake News*. CoRR, abs/1905.12616, 2019.

Generated text appendix

generated text dataset

For train logistic regression algorithm we need to create a generated texts data-set with half GPT-2-FT texts half GPT-2 texts. But to generate sport text, the network need a first sentence. Indeed, GPT-2-FT improve the words sports perplexity and the dependencies in the sport lexical fields, but if the first word is Iraq it will never generate a sport news. So, I write a program which collect the n firsts words for each text of human data-set. After a first generation of a set of text with 15 words in input, I observed a large number of repetitions in the texts of the new data-set. I supposed GPT-2 needed more context, I tested with 25 words and I noted a big improvement. I note also a weakness in long-dependencies with a shift of subject between the beginning and the end of the text if it is too long. I produced five texts from first words of each human text and I produced 300 texts in total with GPT-2-FT. I reproduce the same operation for GPT-2.

I then built the test data-set with the same method.

Logistic regression

Démonstration. We want to determine the value of the variable Y by the TF-IDF.

We note + : Be human sport newspaper, and - : Be generated sport newspaper

Let Y a new text whose we want to determine the class.

Let $X = x_1, ..., x_n$ the TF-IDF linked to Y.

We want to determine the probability that X belongs to the human class knowing X, in other words we search P(Y = + | X).

We know that

$$P(Y = y|X) = \frac{P(Y = y) \cdot P(X|Y = y)}{P(X)}$$

So, we have :

$$\frac{P(Y = +|X)}{P(Y = -|X)} = \frac{P(Y = +)}{P(Y = -)} \cdot \frac{P(X|Y = +)}{P(X|Y = -)}$$

If the ratio is superior to 1, we classify Y in the class +, else Y in the class -.

Our data-set is made up equal parts of human text and generated text, so $\frac{P(Y=+)}{P(Y=-)} = 1$.

The logistic regression introduce the hypothesis following :

$$ln\left(\frac{P(X|Y = +)}{P(X|Y = -)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

where $\beta_0, \beta_1, ..., \beta_n$ are the weights that the network will learn. As we are in the binary classification, we use the Bernoulli law.

From where :

By noting $P = P(X|Y = +)$

and $\beta X = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$

$$ln\left(\frac{P}{1-P}\right) = \beta X = logit(P)$$

which is equivalent to :

$$P = \frac{exp(\beta X)}{1 + exp(\beta X)} = \frac{1}{1 + exp(-\beta X)} = sigmoid(\beta X)$$

□

text generated by GPT-2-FT

text generated by GPT-2-FT with 5 epochs

PSG football game against OM gave the PSG winner 3 points 2 points thanks to 2 goals from Mbappé and a shot on goal from Griezmann. However, even though the game's results were not as impressive as expected, the PSG players and coaches were given ample time to assess the game's performance. As a result of the PSG's win, after the game, the PSG players and coaches got to talk the players to discuss the game. The PSG coach (Robi) said : "We had a good game, so we can always work on our technical quality. "We were really good in the second half, but we lost 2-1, so we need to take this as a win. "The game is a big game for us, because our other players proved to be faster. "We are looking forward to the game, but we have to keep our focus for the next game. "We are going to get the result for us. I know we are going to get the win. But I don't think we can win every game, because we have to do everything we can." Boden, meanwhile, went on to say that he was very happy with the results of the game, and he is sure that his team will be good too.

text generated by GPT-2-FT with 5 epochs

PSG football game against OM gave the PSG winner 3 points 2 points thanks to 2 goals from Mbappé and a shot on goal from Griezmann. The PSG are now 2-0 up with 1 :00 left in the game, but the first goal of the opening half was a clear winner for PSG. The two points give the PSG an extra point after Brie took a pass from Mbappé, but the ball was targeted instead. However, the PSG counter-attacked and lifted the lead by three goals after a superb save from Kaka. The goal was again Mbappé's 65th of the season, but he was up against wide areas and it was in fact 6-0. The PSG striker replaced Mbappé in the 30th minute, but the ball went away and PSG had their chance to score. The lead was extended soon after, with the ball he made to Mbappé was deflected into the net. The ball was then put into the net from the left, but the ball was promptly ruled out. The PSG man made a great save and the ball went out of the net. The PSG man then was spotted by some of the fans behind the goal.

text generated by GPT-2-FT with 30 epochs

PSG won the football match against OM 3 points to 2, thanks to 2 goals from Mbappé and a penalty from Griezmann. The tie was also decided by a first leg draw with Bournemouth, who had won 3-2 on the back of a penalty by Dzeko. The result was the first time the two clubs had met since the 1970s. Real Sociedad became the first club to win both matches. Bruno Rodríguez, a former defender for the club, was the match's referee. He described the game as "good football" in a statement. Related Articles Neymar admits he has some confidence in coach Antonio Conte after PSG win 3-1 victory over Bournemouth

PSG head coach, Jose Mourinho, 'really good'

Mourinho : I can see people playing this game for the first time in years'

Mourinho : I'll go back to my old team and try and make sure that they don't get a bad result" The match was also decided by a 2-1 scoreline. The Bournemouth side were one of three sides still in the middle of the table. The other two sides were Dybala and Leganes. The match was also decided by a tie for the tie-breaker. PSG lost 2-1 at Bournemouth and a 2-1 defeat at Inter Milan. Bournemouth won 3-2 to the Eredivisie side at home and fell to Bournemouth 4-2 at home. The Eredivisie side won 2-1 to the Bournemouth side at home and fell to Bournemouth 3-0 at home. Bournemouth lost 4-1 at Inter Milan. The match was also decided by a shootout. PSG won 2-1 at Inter Milan and fell to Bournemouth 3-2 at home. The match was also decided by a tiebreaker. PSG lost 4-2 to Bournemouth and fell to Bournemouth 3-0 at home. The Bournemouth side won 2-1 to the Eredivisie side. The match was also decided by a shootout. PSG won 2-1 to Inter Milan and fell to Bournemouth 3-2 at home. The match was also decided by a shootout. PSG won 2-1 to Bournemouth and fell to Bournemouth 3-0 at home. The match was also decided by a shootout. PSG lost 3-2 to Inter Milan and fell to Bournemouth 3-2 at home. The match was also decided by a shootout. PSG won 2-1 to Bournemouth and fell to Bournemouth 3-2 at home. The match was also decided by a shootout. The other two clubs were not directly involved in the match. The Bournemouth side won 3-1 to the Eredivisie side and fell to Bournemouth 2-0 at home. The match was also decided by a shootout. PSG lost 3-2 to the Bournemouth side. The match was also decided by a shootout. The other two clubs were not directly involved in the match. The Bournemouth team won 3-1 to the Eredivisie side and fell to Bournemouth 1-2 at home. The match was also decided by a shootout. The other two clubs won 3-0 to the Bournemouth side. The match was also decided by a shootout. The other two clubs were not directly involved in the match. The match was also decided by a shootout. PSG won 2-2 to the Bournemouth side and fell to Bournemouth 2-0 at home. The match was also decided by a shootout. The other two clubs lost 2-0 to the Eredivisie side and fell to Bournemouth 1-1 at home. The match was also decided by a shootout. The other two clubs lost 1-2 to the Eredivisie side and fell to Bournemouth 1-0 at home. The match was also decided by a shootout. PSG won 2-1 to the Eredivisie

side and fell to Bournemouth 2-0 at home. The match was also decided by a shootout. The other two clubs won 2-1 to the Bournemouth side. The match was also decided by a shootout. The other two clubs lost 2-0 to Inter Milan. The match was also decided by a shootout. PSG won 3-0 to the Eredivisie side and fell to Bournemouth 1-1 at home. The match was also decided by a shootout. PSG won 3-0 to the Eredivisie side and fell to Bournemouth 0-2 at home. The match was also decided by a shootout. The other two clubs lost 2-0 to Inter Milan. The match was also decided by a shootout. The other two clubs lost 2

text generated by GPT-2-FT with 50 epochs

sports eickhoff dominates mets phillies win 3 0 philadelphia phillies' darin ruf hits a two run home run off new york mets relief pitcher sean izazi

The Sixers were trailing 3-0 in the second inning of Game 6 of the Eastern Conference finals and trailed 3-2 in the third. Philadelphia's Yoan Moncada hit a two-run home run off the left-field wall in the top of the third and a pair of home runs off the right-field wall in the fourth. Andrew Bynum, who walked again in the fifth, got the Sixers ahead. Ramona Shelburne, who stole second base in the eighth, stole second base in the ninth and stole home in the 10th.

After a run in the top of the fifth, David Lee doubled the lead with a two-run single in the bottom of the sixth. Trevor Ariza reached on a single off Anthony Hollis to right on a double play in the botto m of the eighth. Salah Mejri stole second base in the top of the eighth, then stole third base in the bottom of the ninth.

The Sixers were trailing 3-0 in the third inning of Game 6 of the Eastern Conference finals and trailed 3-2 in the third. Philadelphia's Yoan Moncada hit a two-run home run off the left-field wall in the top of the third and a pair of home runs off the right-field wall in the fourth. Andrew Bynum, who walked again in the fifth, got the Sixers ahead. Ramona Shelburne, who stole second base in the eighth, s tole second base in the ninth and stole home in the 10th. Andrew Bynum, who stole second base in the eighth, stole second base in the ninth and stole home in the 10th. David Lee doubled the lead with a tw o-run single in the bottom of the seventh. Salah Mejri stole second base in the top of the eighth, then stole third base in the bottom of the ninth. Trevor Ariza reached on a single off Anthony Hollis to right on a double play in the bottom of the ninth. Salah Mejri stole second base in the bottom of the ninth, then stole third base in the bottom of the ninth. David Lee doubled the lead with a two-run sin gle in the bottom of the ninth. Salah Mejri stole second base in the top of the ninth, then stole third base in the bottom of the ninth. David Lee singled to center field, then walked out to center to cen ter to win it. David Lee singled to center field, then walked out to center to win it. David Lee scored a run in the top of the seventh to tie it 2-2.

The Sixers were trailing 3-0 in the second inning of Game 6 of the Eastern Conference finals and trailed 3-2 in the third. Philadelphia's Yoan Moncada hit a two-run home

run off the left-field wall in the top of the seventh. Salah Mejri stole second base in the top of the eighth, then stole third base in the bottom of the ninth. David Lee struck out in the top of the eighth to take it 3-0 in the top of th e seventh. David Lee singled to center field, then walked out to center to win it. David Lee scored a run in the top of the eighth, then walked out to center to win it. David Lee scored a run in the top o f the eighth, then walked out to center to win it. David Lee walked out to center to win it. David Lee scored a run in the top of the eighth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the eighth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center t o win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top of the ninth, then walked out to center to win it. David Lee scored a run in the top

text generated classified as human

bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour-bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjourbonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$ bonjour bonjour bonjour bonjour$\backslash n$

bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour

Code appendix

python program for logistic regression

```python
import os
from sklearn.datasets.base import Bunch
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

def load_corpus(path):
    categories = [folder for folder in os.listdir(path)
        if os.path.isdir(os.path.join(path, folder))]
    print(categories)

    files  = [] # holds the file names relative to the root
    data   = [] # holds the text read from the file
    target = [] # holds the string of the category

    # Load the data from the files in the corpus
    for folder in categories:
        for name in os.listdir(os.path.join(path, folder)):
            files.append(os.path.join(path, folder, name))
            target.append(folder)

            with open(os.path.join(path, folder, name), 'r') as f:
                data.append(f.read())


    # Return the data bunch for use similar to the newsgroups example
    return Bunch(
        categories=categories,
        files=files,
        data=data,
        target=target,
        shuffle=True
    )
path  = '/gpt-2-simple/gpt-2_finetuning/data/data_train'

#to carry out Bunch
X_train = load_corpus(path)

#--------- TF-IDF
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train.data)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)

#parse data for training
x_train, x_test, y_train, y_test = train_test_split(X_train_tfidf, X_train.target,
    test_size=0.25, random_state=0)

#Training
logisticRegr = LogisticRegression()
logisticRegr.fit(x_train, y_train)

```

```
#prediction in the test data-set
predictions = logisticRegr.predict(x_test)
score = logisticRegr.score(x_test, y_test)
print(score)

        #EVALUATION

#SCORE for GPT-2-FT
path2 = '/gpt-2-simple/gpt-2_finetuning/data/data_test/gen_FT'
val_FT  = load_corpus(path2)
FT_counts = count_vect.transform(val_FT.data)
FT_tfidf = tfidf_transformer.transform(FT_counts)
score = logisticRegr.score(FT_tfidf, val_FT.target)
print(score)

#SCORE for GPT-2
path3 = '/gpt-2-simple/gpt-2_finetuning/data/data_test/gen_FT50ep'
val_NOFT = load_corpus(path3)
NOFT_counts = count_vect.transform(val_NOFT.data)
NOFT_tfidf = tfidf_transformer.transform(NOFT_counts)
#predicted = logisticRegr.predict(NOFT_tfidf)
score = logisticRegr.score(NOFT_tfidf, val_NOFT.target)
print(score)
```

Listing 1: logistic regression

python program for one class SVM

```

import os
from sklearn.datasets.base import Bunch
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import numpy as np

import matplotlib.font_manager
from sklearn import svm

def load_corpus(path):

    categories = [
        folder for folder in os.listdir(path)
        if os.path.isdir(os.path.join(path, folder))
    ]
    print(categories)

    files  = [] # holds the file names relative to the root
    data   = [] # holds the text read from the file
    target = [] # holds the string of the category
```

```
24
25      # Load the data from the files in the corpus
26      for folder in categories:
27          for name in os.listdir(os.path.join(path, folder)):
28              files.append(os.path.join(path, folder, name))
29              target.append(folder)
30
31              with open(os.path.join(path, folder, name), 'r') as f:
32                  data.append(f.read())
33
34
35      # Return the data bunch for use similar to the newsgroups example
36      return Bunch(
37          categories=categories,
38          files=files,
39          data=data,
40          target=target,
41          shuffle=True
42      )
43
44
45  path  = '/gpt-2-simple/gpt-2_finetuning/data/test'
46
47  #faire un Bunch
48  X_train = load_corpus(path)
49
50  #r alisation du Tf-IDF
51  count_vect = CountVectorizer()
52  X_train_counts = count_vect.fit_transform(X_train.data)
53  tfidf_transformer = TfidfTransformer()
54  X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
55
56
57  #Entrainement
58  clf = svm.OneClassSVM(nu=0.1, kernel="rbf", gamma=0.1)
59  x_train, x_test, y_train, y_test = train_test_split(X_train_tfidf, X_train.target,
        test_size=0.25, random_state=0)
60  clf.fit(x_train)
61  print('tout va bien')
62  y_pred_train = clf.decision_function(x_test, x_test.target)
63  print(y_pred_train)
64  n_error_train = y_pred_train[y_pred_train == 1].size
65
66  print('rappel', n_error_train/len(y_pred_train))
67  print(len(y_pred_train), n_error_train)
68  #ANNALYSE SUR L'EVALUATION
69
70  #SCORE pour GPT-2-FT
71  path2 = '/gpt-2-simple/gpt-2_finetuning/data/data_test/gen_FT30'
```

```
val_FT  = load_corpus(path2)

FT_counts = count_vect.transform(val_FT.data)
FT_tfidf = tfidf_transformer.transform(FT_counts)

y_pred_test = clf.score_samples(FT_tfidf, vat_FT.target)

print(y_pred_test)
n_error_test = y_pred_test[y_pred_test ==1].size
print(n_error_test/len(y_pred_test))

print(len(y_pred_test), n_error_test)


#SCORE pour GPT-2
path3 = '/gpt-2-simple/gpt-2_finetuning/data/data_test/gen_NOFT'
val_NOFT = load_corpus(path3)

NOFT_counts = count_vect.transform(val_NOFT.data)
NOFT_tfidf = tfidf_transformer.transform(NOFT_counts)

y_pred_outliers = clf.score(NOFT_tfidf, val_NOFT.target)

print(y_pred_outliers)
n_error_outliers = y_pred_outliers[y_pred_outliers ==1].size
print(n_error_outliers/len(y_pred_outliers))
print(len(y_pred_outliers), n_error_outliers)
```

Listing 2: one class SVM