NBA Summary Generator: Game Statistics to Summary Generation

Ben Shapira **Guy Gara Bagi**

Tel-Aviv University

{benshapira, guy.garabagi}@mail.tau.ac.il

https://github.com/guybagi435/nba_summary_generator

Abstract

Recent years have witnessed the rise in popularity of pretrained language models for text generation tasks. In this work, we suggest a slightly different perspective of the data-to-text generation task. Our goal is to make good use of state-of-the-art pretrained models and their great understanding of natural language, to generate NBA game summary, given the game statistics as input. Game statistics, in their nature, contain mainly numbers and therefore aren't written in natural language. We flattened the statistics table and composed a template paragraph written in natural language, expressing the game statistics numbers in words, to utilize the T5 model natural language understanding.

Another fascinating task we researched in our work is the Fusion-In-Decoder idea, taken from the Question Answering field. We suggest a new way to adjust and use it when dealing with long input text, which is well separated into parts. This idea helps the model with understanding the hierarchy of the input text.

1 Introduction

Over the past several years, neural text generation systems have shown impressive performance on tasks such as machine translation and summarization, while using transformer-based models such as T5 – a text-to-text model, pre-trained on a data-rich task before being fine-tuned on a downstream task.

A classic problem in natural-language generation involves taking structured data, such as a table, as input, and producing text that adequately and fluently describes this data as output. In our case – we get NBA game statistics as input and are required to generate a full game summary.

However, game statistics, in their nature, are structured as a table and consist mostly of numbers. This fact creates new challenges for the encoder-decoder approach, as encoding a table isn't a straightforward task and raises some interesting questions – should we address the table as a bunch of sentences? Does a cell position have a contextual meaning such as a word position in a sentence? And more practically, how can we use the powerful state-of-the-art encoder-decoder models and their amazing NL processing abilities. To suggest a way of dealing with those questions, with our specific task in mind, we tried a different approach. We flattened the data into a few natural language sentences portraying the statistics table in a verbal way, such as a sports reporter would have presented it to the viewers – and then turned the problem into a text-to-text task, so we can ask T5 NL understanding for help.

The second problem that stood in front of us was the length of the input. Converting the tabular input to NL created a long sequence after tokenization. That was an issue due to T5's input length limitations. Many have tried to solve this issue, such as Guo et al., 2021 and others, but we chose to get inspiration from Izacard et al., 2021 who came up with the Fusion-in-Decoder (FID) idea for a Question Answering task.

Our input data is naturally divided into several parts: general game statistics (who won, by how much, etc.), home team statistics, visiting team statistics, and both teams' player statistics. This fact drove us to solve the issue by dividing our inputs into parts and using the FID method.

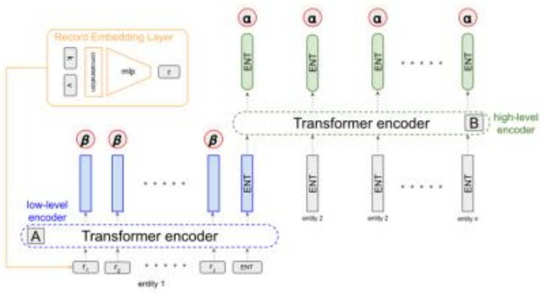
2 Prior work

Several different approaches were taken during the last few years to attack this exact task. We will introduce a couple of them very briefly. All prior works used the Rotowire dataset of NBA game stats and their corresponding summaries.

Challenges in Data-to-Document Generation (2017):

Wiseman et al. tried to attack the data to text generation through this task of game statistics to summary generation task by treating the data in its original tabular shape. Their base Architecture contains input data of comma-separated stats data, and their neural architecture consists of an embedding layer, MLP layer, and 2-layer LSTM decoder with attention and input-feeding.

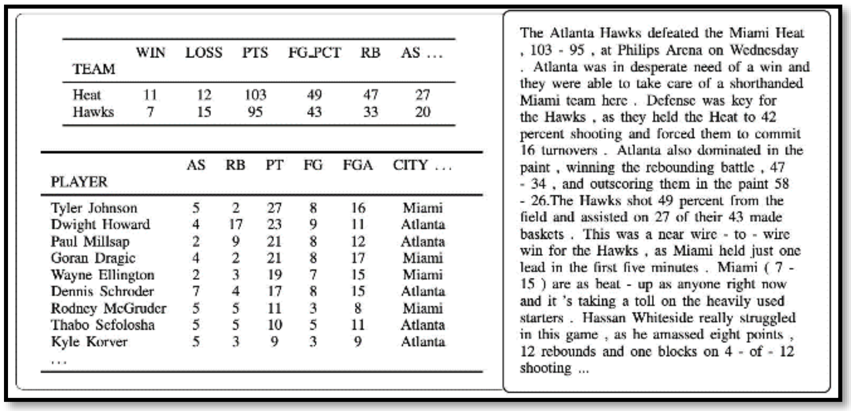
A Hierarchical Model for Data-to-Text Generation (2019):



Rebuffel et al. attacked the data-to-text task with a hierarchical model that encodes the data structure at the element level and the structure level. They evaluated their model on the Rotowire data. Their base Architecture consists of two embedding layers (to embed the record keys and values), low-level encoder, high-level encoder, hierarchical attention, and LSTM decoder.

A few more: Data-to-text Generation with Macro Planning, Data-to-text Generation with Entity Modeling

2 Data



TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS …
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY …
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
…						

The Atlanta Hawks defeated the Miami Heat , 103 - 95 , at Philips Arena on Wednesday . Atlanta was in desperate need of a win and they were able to take care of a shorthanded Miami team here . Defense was key for the Hawks , as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers . Atlanta also dominated in the paint , winning the rebounding battle , 47 - 34 , and outscoring them in the paint 58 - 26.The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets . This was a near wire - to - wire win for the Hawks , as Miami held just one lead in the first five minutes . Miami (7 - 15) are as heat - up as anyone right now and it 's taking a toll on the heavily used starters . Hassan Whiteside really struggled in this game , as he amassed eight points , 12 rebounds and one blocks on 4 - of - 12 shooting ...

Figure 1: An example NBA game statistics and summary pair from the RotoWire dataset. Taken from Weisman et al. paper.

We used the SportSett:Basketball Database

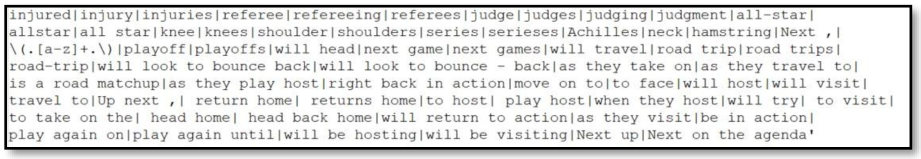
This Database (PostgreSQL) consists of (human-written) NBA basketball game summaries aligned with their corresponding box-and-line scores. There are 7202 distinct Rotowire (sports news website) summaries, covering NBA games played between 2014 and 2018

To work with text files, we had to scrape the Database's relevant data from text files using various queries.

The resulting artifacts were .txt files for both summaries and statistics.

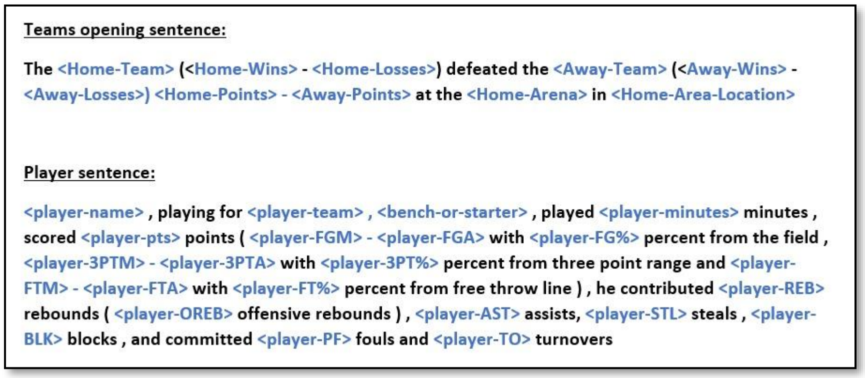
Next, we preprocessed the data in two stages:

1) cleaning game summaries by stripping information that cannot be inferred by the input statistics. We created a blacklist containing words that we identified as non-informative and can potentially damage the model prediction abilities (for instance 'injured', 'next game', 'will host' – see figure 2.)



```
injured|injury|injuries|referee|refereeing|referees|judge|judges|judging|judgment|all-star|
allstar|all star|knee|knees|shoulder|shoulders|series|serieses|Achilles|neck|hamstring|Next ,|
\(.[a-z]+.\)|playoff|playoffs|will head|next game|next games|will travel|road trip|road trips|
road-trip|will look to bounce back|will look to bounce - back|as they take on|as they travel to|
is a road matchup|as they play host|right back in action|move on to|to face|will host|will visit|
travel to|Up next ,| return home| returns home|to host| play host|when they host|will try| to visit|
to take on the| head home| head back home|will return to action|as they visit|be in action|
play again on|play again until|will be hosting|will be visiting|Next up|Next on the agenda'
```

Figure 2: blacklist for summary cleanup


Teams opening sentence:

The <Home-Team> (<Home-Wins> - <Home-Losses>) defeated the <Away-Team> (<Away-Wins> - <Away-Losses>) <Home-Points> - <Away-Points> at the <Home-Arena> in <Home-Area-Location>

Player sentence:

<player-name> , playing for <player-team> , <bench-or-starter> , played <player-minutes> minutes , scored <player-pts> points (<player-FGM> - <player-FGA> with <player-FG%> percent from the field , <player-3PTM> - <player-3PTA> with <player-3PT%> percent from three point range and <player-FTM> - <player-FTA> with <player-FT%> percent from free throw line) , he contributed <player-REB> rebounds (<player-OREB> offensive rebounds) , <player-AST> assists, <player-STL> steals , <player-BLK> blocks , and committed <player-PF> fouls and <player-TO> turnovers

Figure 3: sentences template example

2) To utilize T5 model NL understanding we converted the statistics into plain texts by injecting stats data into generic passages (figure 3.) Which will later be used as input to the model. The final form of each game data consists of 17 passages (this data architecture will later be explained in the model section):

The first 3 passages are teams related and contain general game statistics; the other 14 passages are related to leading players from each team (leading players = 5 starters + 2 bench players who played the most minutes) (figure 5.)

4 Model

For the task of generating a large summary paragraph (a minimum of 5 sentences), we wanted to use a pre-trained NLP model, capable of generating a large portion of text. Pre-trained transformers such as T5 have shown outstanding performance in the text-to-text field and that's why they seem like a good idea for tackling our task. Our initial data was tabular-shaped, therefore we wanted to redefine the task as a sequence-to-sequence task. We had to process the data and convert it to a human-readable (or transformer-readable) shape first (further explanation can be found in the Data section).

An immediate issue we encountered was encoding long inputs. One obvious solution was to find a pre-trained model that accepts more than 512 tokens as input. Choosing this solution means adding parameters and dealing with large models which are harder to train and require more computation resources.

Another possible solution was to prune the input data to fit encoders that allow a maximum of 512 input tokens. We Investigated this approach and tried to shorten our input by eliminating both teams' and players' data, but eventually concluded that such an approach will cause a loss of necessary information.

All the above led us to find a solution that is involving encoding parts of the input separately and then somehow decoding them together.

We noticed that our data is structured in a way that can be naturally split into logical parts (general game details, each team's stats, and players' stats).

Fusion-in-Decoder is an idea that was introduced in the work of Izacard et al. (Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering). They proposed a new retriever-reader-based model for the QA domain, where the main idea is that the retriever retrieves passages from the web that are related to the question in a way, scores them, and passes a list of passages to the reader. The reader is answering the question based on the input passages.

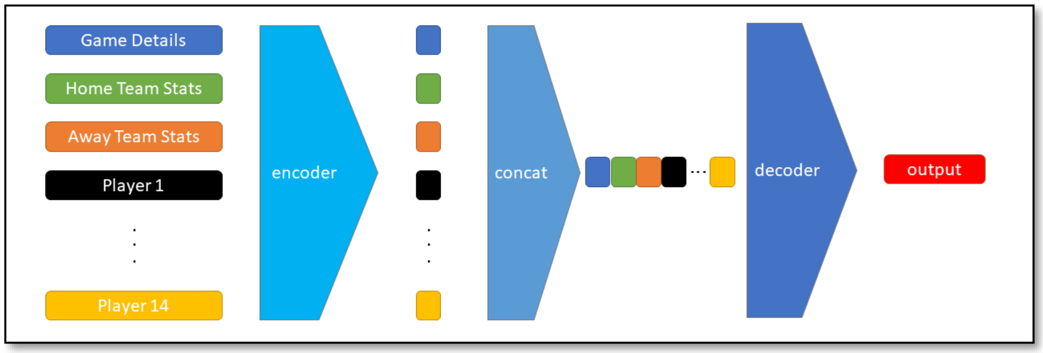


Figure 4: model architecture

In order to use the information from the passages, they introduced the Fusion-in-Decoder method, where those passages are being supplied to the encoder separately and then the encoded passages are being concatenated for the decoder to use attention over all the passages encoding together.

Although fusion-in-decoder was originally used for Question Answering tasks, where the different passages were possible answer sources to the same question and are related to each other - we used it for concatenating the encoding of unrelated passages, hoping the decoder will realize the relation between each passage and the desired summary, by applying attention over the concatenated passages.

4.1 Model Architecture

The model is based on the T5 seq2seq model. Due to limited computation resources, we used the small model size (60 million parameters).

First, each passage from the input is being tokenized using padding to max length to make all tokenized passages in the same length, so later, when concatenating the encodings, the dimension of each passage encoding will be identical.

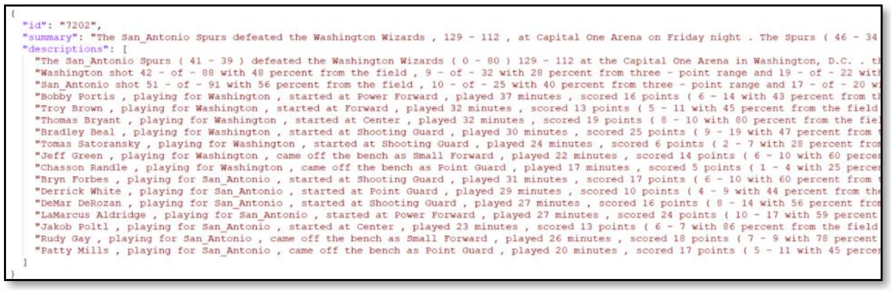
The tokenized input is then supplied to the encoder to start the forward pass. We override the T5 encoder with a new class we created called EndocerWrapper that encodes each of the passages independently and then concatenates the encodings to supply them to the decoder (based on FID by Facebook).

4.2 Training

We split the data into 3 parts: train (80%), evaluation (1%), and test (19%).

Each training session was 3 epochs long and with a batch size of 1. The training process took 2 weeks while each training session's parameters were saved (checkpointing) in order to be used in the next session. we conducted 27 epochs overall.

We chose a learning rate of 0.001 and "Adam" as the optimizer, as this method is efficient when working with large problems involving a lot of data or parameters, it requires less memory and is efficient.



```
{
  "id": "7202",
  "summary": "The San_Antonio Spurs defeated the Washington Wizards , 129 - 112 , at Capital One Arena on Friday night . The Spurs ( 46 - 34
  "descriptions": [
    "The San_Antonio Spurs ( 41 - 39 ) defeated the Washington Wizards ( 0 - 80 ) 129 - 112 at the Capital One Arena in Washington, D.C. . th
    "Washington shot 42 - of - 88 with 48 percent from the field , 9 - of - 32 with 28 percent from three - point range and 19 - of - 22 with
    "San_Antonio shot 51 - of - 91 with 56 percent from the field , 10 - of - 25 with 40 percent from three - point range and 17 - of - 20 wi
    "Bobby Portis , playing for Washington , started at Power Forward , played 37 minutes , scored 16 points ( 6 - 14 with 43 percent from th
    "Troy Brown , playing for Washington , started at Forward , played 32 minutes , scored 13 points ( 5 - 11 with 45 percent from the field
    "Thomas Bryant , playing for Washington , started at Center , played 32 minutes , scored 19 points ( 8 - 10 with 80 percent from the fiel
    "Bradley Beal , playing for Washington , started at Shooting Guard , played 30 minutes , scored 25 points ( 9 - 19 with 47 percent from t
    "Tomas Satoransky , playing for Washington , started at Shooting Guard , played 24 minutes , scored 6 points ( 2 - 7 with 28 percent from
    "Jeff Green , playing for Washington , came off the bench as Small Forward , played 22 minutes , scored 14 points ( 6 - 10 with 60 percen
    "Chasson Randle , playing for Washington , came off the bench as Point Guard , played 17 minutes , scored 5 points ( 1 - 4 with 25 percen
    "Bryn Forbes , playing for San_Antonio , started at Shooting Guard , played 31 minutes , scored 17 points ( 6 - 10 with 60 percent from t
    "Derrick White , playing for San_Antonio , started at Point Guard , played 29 minutes , scored 10 points ( 4 - 9 with 44 percent from the
    "DeMar DeRozan , playing for San_Antonio , started at Shooting Guard , played 27 minutes , scored 16 points ( 8 - 14 with 56 percent from
    "LaMarcus Aldridge , playing for San_Antonio , started at Power Forward , played 27 minutes , scored 24 points ( 10 - 17 with 59 percent
    "Jakob Poltl , playing for San_Antonio , started at Center , played 23 minutes , scored 13 points ( 6 - 7 with 86 percent from the field
    "Rudy Gay , playing for San_Antonio , came off the bench as Small Forward , played 26 minutes , scored 18 points ( 7 - 9 with 78 percent
    "Patty Mills , playing for San_Antonio , came off the bench as Point Guard , played 20 minutes , scored 17 points ( 5 - 11 with 45 percen
  ]
}
```

Figure 5: final input data form

5 Experiments

During the training of the model, we evaluated its outputs and after some epochs, we started to feel that the model is overfitting. We noticed a similar structure for the output, containing the same sentences describing the same summary events. For instance, if you look at Figure 6, during this epoch most of the summaries contained the same opening sentences describing the first quarter in the same words. In addition, we encountered another issue of duplicate sentences in the same output.

We realized that there exists a tradeoff between the two issues. On the one hand, we want to keep training for a few more epochs in order to get rid of the duplicate sentences, but on the other hand, more epochs can create an overfit.

We decided to add regularization to the model, by adding a dropout layer of 20% and to keep on training the model for some more time.

To test the model performance, we had to choose a scoring metric that will be suitable for the task. For starters, the intuitive way was to evaluate the test with BLEU and ROUGE, as they are lightweight and easy to implement.

Both BLUE and ROUGE are scoring metrics based on recall and precision calculations, related to the appearance of common words in the summaries against the references.

For our task, those metrics seem a bit irrelevant, as our main interest was to measure the meaning similarity between the prediction and reference, which can be achieved using smart NLP model-based scoring matrices.

BERTScore leverages the pre-trained contextual embeddings from various NLP BERT-like models and matches words in prediction and reference sentences by cosine similarity.

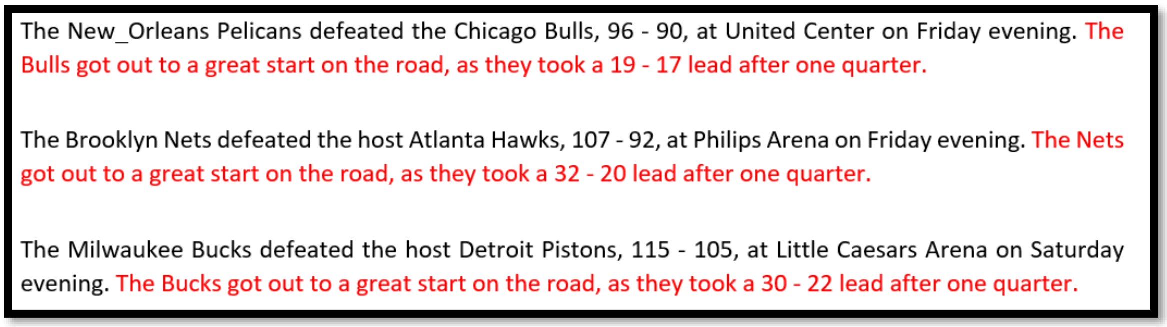
We decided to use BART-large as the model for the BERTScore to evaluate our outputs, as each output contains up to 700 words, which does not fit base-size models that accept 512 tokens.

5.1 Results

Scoring metric / Scores	**Precision**	**Recall**	**F1-Score**
ROUGE	50.03	34.9	**39.42**
BLEU	-	-	**7.27**
BERTScore	65.97	59.54	**62.37**

Values*100

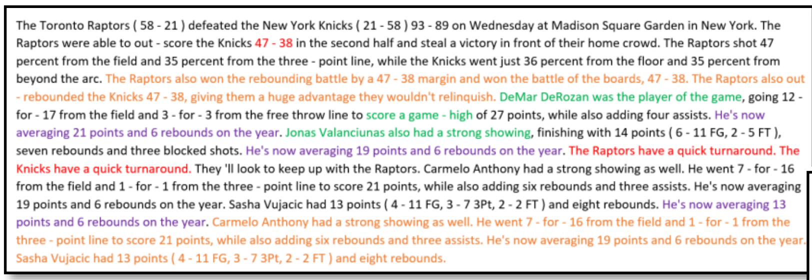
As we anticipated, BERTScore produced the highest score, as it has some knowledge of the overall meaning.


The New_Orleans Pelicans defeated the Chicago Bulls, 96 - 90, at United Center on Friday evening. The Bulls got out to a great start on the road, as they took a 19 - 17 lead after one quarter.

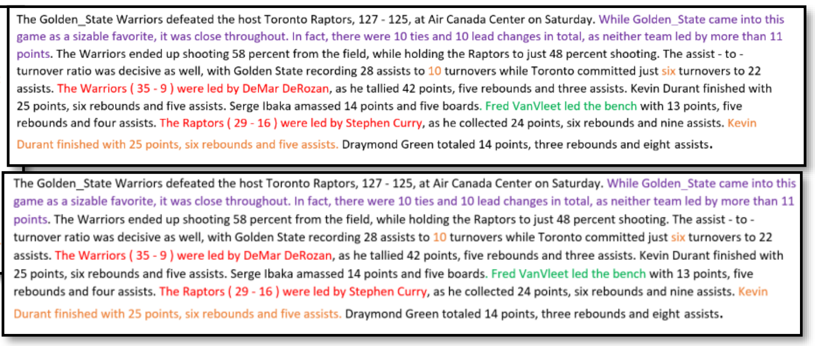
The Brooklyn Nets defeated the host Atlanta Hawks, 107 - 92, at Philips Arena on Friday evening. The Nets got out to a great start on the road, as they took a 32 - 20 lead after one quarter.

The Milwaukee Bucks defeated the host Detroit Pistons, 115 - 105, at Little Caesars Arena on Saturday evening. The Bucks got out to a great start on the road, as they took a 30 - 22 lead after one quarter.

Figure 6: recurring sentences. Example for overfitting.



The Toronto Raptors (58 - 21) defeated the New York Knicks (21 - 58) 93 - 89 on Wednesday at Madison Square Garden in New York. The Raptors were able to out - score the Knicks 47 - 38 in the second half and steal a victory in front of their home crowd. The Raptors shot 47 percent from the field and 35 percent from the three - point line, while the Knicks went just 36 percent from the floor and 35 percent from beyond the arc. The Raptors also won the rebounding battle by a 47 - 38 margin and won the battle of the boards, 47 - 38. The Raptors also out - rebounded the Knicks 47 - 38, giving them a huge advantage they wouldn't relinquish. DeMar DeRozan was the player of the game, going 12 - for - 17 from the field and 3 - for - 3 from the free throw line to score a game - high of 27 points, while also adding four assists. He's now averaging 21 points and 6 rebounds on the year. Jonas Valanciunas also had a strong showing, finishing with 14 points (6 - 11 FG, 2 - 5 FT), seven rebounds and three blocked shots. He's now averaging 19 points and 6 rebounds on the year. The Raptors have a quick turnaround. The Knicks have a quick turnaround. They 'll look to keep up with the Raptors. Carmelo Anthony had a strong showing as well. He went 7 - for - 16 from the field and 1 - for - 1 from the three - point line to score 21 points, while also adding six rebounds and three assists. He's now averaging 19 points and 6 rebounds on the year. Sasha Vujacic had 13 points (4 - 11 FG, 3 - 7 3Pt, 2 - 2 FT) and eight rebounds. He's now averaging 13 points and 6 rebounds on the year. Carmelo Anthony had a strong showing as well. He went 7 - for - 16 from the field and 1 - for - 1 from the three - point line to score 21 points, while also adding six rebounds and three assists. He's now averaging 19 points and 6 rebounds on the year. Sasha Vujacic had 13 points (4 - 11 FG, 3 - 7 3Pt, 2 - 2 FT) and eight rebounds.



The Golden_State Warriors defeated the host Toronto Raptors, 127 - 125, at Air Canada Center on Saturday. While Golden_State came into this game as a sizable favorite, it was close throughout. In fact, there were 10 ties and 10 lead changes in total, as neither team led by more than 11 points. The Warriors ended up shooting 58 percent from the field, while holding the Raptors to just 48 percent shooting. The assist - to - turnover ratio was decisive as well, with Golden State recording 28 assists to 10 turnovers while Toronto committed just six turnovers to 22 assists. The Warriors (35 - 9) were led by DeMar DeRozan, as he tallied 42 points, five rebounds and three assists. Kevin Durant finished with 25 points, six rebounds and five assists. Serge Ibaka amassed 14 points and five boards. Fred VanVleet led the bench with 13 points, five rebounds and four assists. The Raptors (29 - 16) were led by Stephen Curry, as he collected 24 points, six rebounds and nine assists. Kevin Durant finished with 25 points, six rebounds and five assists. Draymond Green totaled 14 points, three rebounds and eight assists.

The Golden_State Warriors defeated the host Toronto Raptors, 127 - 125, at Air Canada Center on Saturday. While Golden_State came into this game as a sizable favorite, it was close throughout. In fact, there were 10 ties and 10 lead changes in total, as neither team led by more than 11 points. The Warriors ended up shooting 58 percent from the field, while holding the Raptors to just 48 percent shooting. The assist - to - turnover ratio was decisive as well, with Golden State recording 28 assists to 10 turnovers while Toronto committed just six turnovers to 22 assists. The Warriors (35 - 9) were led by DeMar DeRozan, as he tallied 42 points, five rebounds and three assists. Kevin Durant finished with 25 points, six rebounds and five assists. Serge Ibaka amassed 14 points and five boards. Fred VanVleet led the bench with 13 points, five rebounds and four assists. The Raptors (29 - 16) were led by Stephen Curry, as he collected 24 points, six rebounds and nine assists. Kevin Durant finished with 25 points, six rebounds and five assists. Draymond Green totaled 14 points, three rebounds and eight assists.

Figure 7: model output example 1.

Figure 7 shows the model outputs. We colored some of the sentences to highlight interesting outliers.

Red - the correct data existed in the input, but the model outputted the wrong prediction.

Purple – output that cannot be inferred from the input data.

Orange – data issues – duplicate sentences, unreadable sentences, etc.

Green – special insights the model had on the data.

Black – correct data.

5.2 Output Analysis

The overall output the model generates is very informative and human-readable. The fusion-in-decoder seems to have worked in the sense that the decoder outputs well-structured sentences similar to the input passages.

The model had nice insights into the data, as it inferred hidden information, that was not explicitly given in the input data, for instance, the model inferred triple-doubles and double-doubles for players that had scored ten or more in three/two different categories. It also inferred the player of the game and the leader from the bench.

Common issues – the model sometimes miss-identify the player's team, switches between the team in terms of quarter scores, duplicates sentences, and mentions the same player for both teams. Some of the repeating sentences appear at the end of the summary, so a possible solution to that would be to limit the maximum output size to a smaller size.

The model outputs some information that is not backed by the input data. Probably due to information that exists, during the training, in the reference data but not in the input data.

6 Conclusion & Further Work

To conclude, we find the experiment successful.

From our perspective adding a fusion-in-decoder layer for SOTA models such as T5 can help in tasks where the input data is longer than 512 tokens. We showed that fusion-in-decoder can work in other fields rather than the QA field.

Also, tabular data can be flattened into the human-readable text to make use of models that expect NL inputs.

6.1 Further Work

We would recommend trying and using larger models, as the task of generating long summaries might be hard for 60 million parameters-based models.

In addition, try using models that were pre-trained on sports-related data, such as Microsoft's SportsBERT.

Rich the input data with play-by-play statistics for the model to "better understand" the reference data.