AI Cricket Narrator: Bridging the Gap between Human and AI Commentary for Enhanced Viewer Experience

This project evaluates the quality of automatically generated cricket match commentaries by comparing their relevance, coherence, and grammaticality to human-generated commentaries, showcasing a comprehensive assessment through visual representation.

Dr. Haihua Chen

haihua.chen@unt.edu

May 5, 2024

Team Members:

1. **SAI TEJA RAYABARAPU**
2. **MANASWINI KODELA**

Index

Authors

Sai Teja Rayabarapu (Student ID: 11637122)

SaiTejaRayabarapu@my.unt.edu

Major: Information Science

Manaswini Kodela (Student ID: 11637117)

ManaswiniKodela@my.unt.edu

Major: Information Science

Facts

Cricket Basics: Cricket is a popular bat-and-ball game played between two teams of eleven players each, originating in England and gaining global popularity.

Types of Matches: Cricket matches come in three main formats: Test cricket (up to five days), One Day Internationals (ODIs, 50 overs per side), and Twenty20 (T20, 20 overs per side).

Cricket Commentary: Cricket commentary involves live narration and analysis of matches, providing insights, updates, and highlights to engage the audience.

Role of Commentators: Commentators play a crucial role in enhancing the viewer experience by describing on-field action, analyzing player performances, and offering historical and statistical insights.

Evolution of Commentary: Cricket commentary has evolved from traditional radio broadcasts to modern televised and digital formats, leveraging technology to enhance presentation and analysis.

Reason For Choosing this Topic

1

Enhanced Viewer Experience: High-quality commentary greatly enhances the viewing experience of cricket matches, providing valuable insights, context, and excitement that amplify the enjoyment of the game.

2

Importance of Commentary: Commentary serves as a vital companion to live cricket matches, offering analysis, storytelling, and expert opinion that deepen understanding and appreciation of the game's nuances.

3

Potential for Improvement: While automated commentary systems offer efficiency and scalability, there's a need to ensure that the generated content maintains the same level of quality and engagement as human-generated commentary.

4

Advancements in Technology: With advancements in natural language processing (NLP) and machine learning, there's an opportunity to leverage these technologies to enhance the quality and accuracy of automated cricket match commentaries.

5

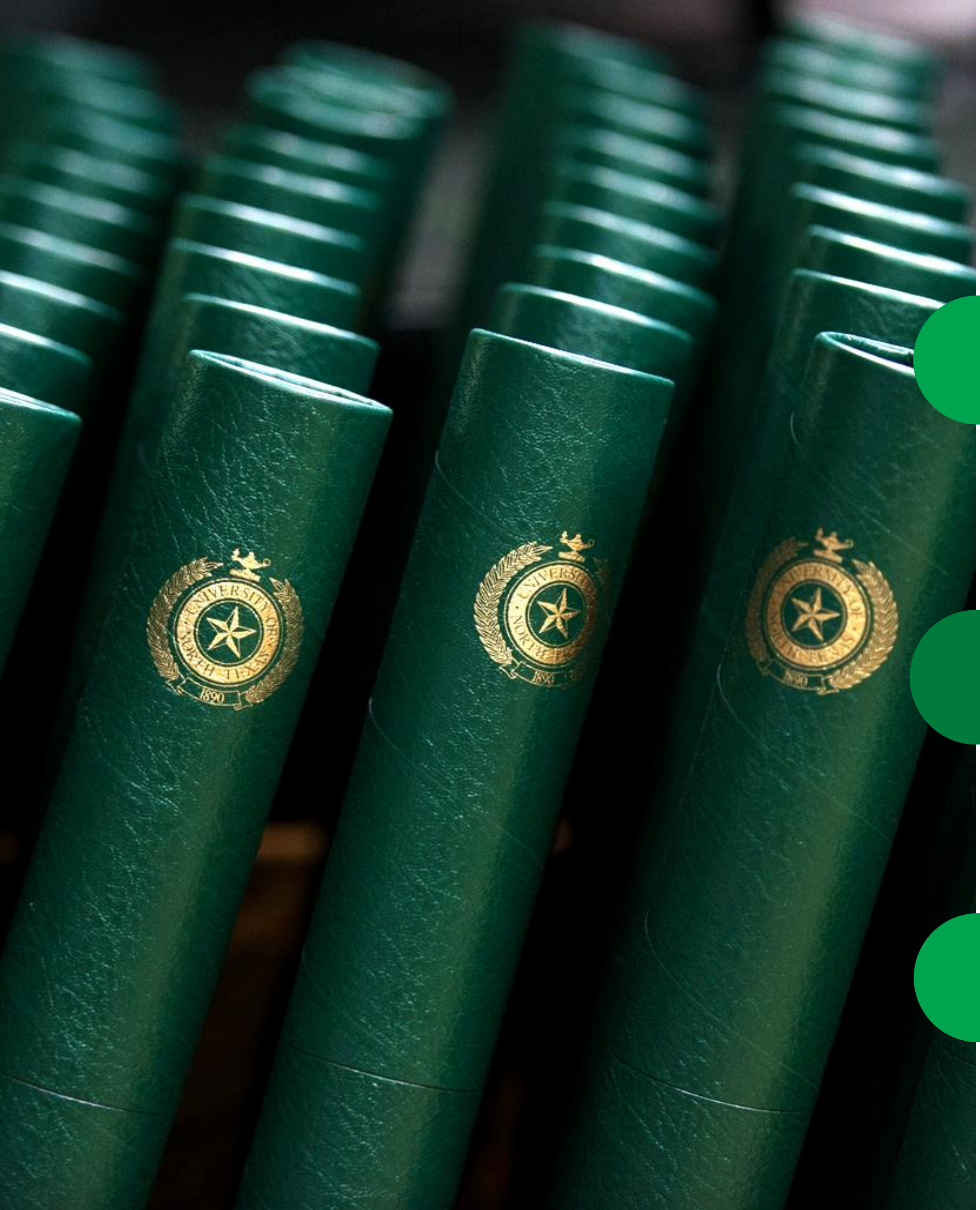
Personal Interest and Passion: As cricket enthusiasts, contributing to the improvement of cricket commentary aligns with our passion for the sport, allowing you to make a meaningful impact in an area that deeply resonates with you.

Problem Statement

In the realm of cricket broadcasting, the automated generation of match commentaries has emerged as a promising solution to cater to the growing demand for real-time updates and analysis. However, ensuring the quality and effectiveness of these automated commentaries poses a significant challenge. While automation offers efficiency and scalability, there is a risk of compromising the accuracy, coherence, and grammaticality of the content. Therefore, the problem at hand is to develop a robust framework for evaluating the quality of automatically generated cricket match commentaries, with a focus on assessing relevance, coherence, and grammaticality, to ensure their effectiveness in engaging viewers and providing meaningful insights during live matches.

Background

Cricket, a popular sport globally, captivates millions of fans with its thrilling matches and dramatic moments. With the advent of technology, there's a growing interest in automatically generating cricket match commentaries to enhance the viewer experience and provide real-time updates.



Research Objectives

Evaluation of Commentary Quality

The primary objective of the research is to assess the quality of automatically generated cricket match commentaries. This involves analyzing key metrics such as relevance, coherence, and grammaticality to ensure that the generated content effectively conveys match highlights and engages viewers.

Identification of Improvement Areas

Another objective is to identify areas for improvement in automated commentary systems. By pinpointing shortcomings in relevance, coherence, or grammaticality, the research aims to propose strategies and enhancements to enhance the overall quality and effectiveness of automated cricket match commentaries.

Enhancement of Viewer Experience

Ultimately, the research seeks to contribute to the enhancement of the viewer experience during cricket matches. By refining automated commentary systems to deliver more accurate, coherent, and engaging content, the research aims to enrich the experience of cricket enthusiasts and enhance their enjoyment of the game.

Related Work

Real-time cricket commentary serves as a valuable resource for insightful analysis, providing genuine perspectives on match dynamics (Roy et al., 2021).

Automated cricket commentary systems leverage dynamic web scraping techniques and supervised learning algorithms to overcome traditional challenges in live text commentary (Hegde et al., 2021).

Machine learning algorithms, such as those used in IPL match prediction models, integrate various factors like toss outcomes and player statistics to enhance predictive capabilities (Sinha, 2020).

Text commentary plays a crucial role in conveying match information and shaping the narrative surrounding live sports events (More et al., 2022).

Future advancements in cricket analysis include the development of automatic commentary generation models, which utilize computer vision and natural language processing techniques to deepen insights into the game (Ul Abideen et al., 2021).

Methodology

Text Preprocessing: The text data is preprocessed by removing non-alphabetic characters, converting text to lowercase, tokenizing the text, removing stopwords and punctuation, and performing feature engineering to create processed text.

Language Model Training: A language model is trained using the FastAI library. The training data consists of text snippets extracted from cricket commentary. The model is fine-tuned using hyperparameter tuning to optimize its performance.

Prediction Generation: After training, the language model is used to generate predictions for given text snippets. These predictions aim to emulate the style and content of human-generated cricket commentary.

Evaluation: The generated predictions are evaluated using various metrics such as relevance, coherence, and grammaticality to assess the quality of the generated commentary.

Methodology(Cont.)

- The model used in the provided code is a language model, specifically an AWD-LSTM (AWD-LSTM stands for "ASGD Weight-Dropped LSTM") implemented using the FastAI library. This model is trained to understand and generate natural language text. It leverages a recurrent neural network (RNN) architecture called Long Short-Term Memory (LSTM), which is capable of capturing long-range dependencies in sequential data like text.
- The AWD-LSTM architecture incorporates various regularization techniques such as weight dropout and activation regularization to prevent overfitting and improve generalization performance. It has been shown to achieve state-of-the-art results in language modeling tasks.
- In this project, the language model is trained on a corpus of cricket commentary text to learn the patterns and structure of cricket-related language. Once trained, the model is capable of generating new cricket commentary text that mimics the style and content of human-generated commentary.

CHALLENGES

1. DATA QUALITY: ENSURING THE QUALITY OF THE TRAINING DATA, AS WELL AS THE RELEVANCE AND ACCURACY OF THE GENERATED PREDICTIONS, COULD BE CHALLENGING DUE TO THE SUBJECTIVE NATURE OF CRICKET COMMENTARY.
2. MODEL PERFORMANCE: OPTIMIZING THE PERFORMANCE OF THE LANGUAGE MODEL TO ACCURATELY CAPTURE THE NUANCES OF CRICKET COMMENTARY AND PRODUCE COHERENT AND GRAMMATICALLY CORRECT TEXT COULD BE CHALLENGING.
3. EVALUATION METRICS: DETERMINING APPROPRIATE EVALUATION METRICS TO ASSESS THE QUALITY OF THE GENERATED COMMENTARY MAY REQUIRE CAREFUL CONSIDERATION AND EXPERIMENTATION.

Research Contributions

1.**Automated Cricket Commentary Generation**: The project introduces an automated system capable of generating cricket commentary text. This technology can aid sports broadcasters, content creators, and cricket enthusiasts by providing engaging and real-time commentary during matches.

2.**Language Modeling in Sports Commentary**: The project demonstrates the application of language modeling techniques, particularly AWD-LSTM neural networks, in the context of sports commentary. It showcases the feasibility and effectiveness of using advanced natural language processing (NLP) models for sports-related text generation tasks.

3.**Enhancing User Experience**: By generating commentary in real-time or on-demand, the system can enhance the overall user experience for cricket fans. It provides additional insights, analysis, and entertainment value, thereby enriching the viewing experience of cricket matches.

4.**Scalability and Adaptability**: The developed system can be adapted to various cricket formats, leagues, and languages. Its scalability allows for integration into different platforms, including live broadcasts, sports apps, and social media channels, catering to diverse audience preferences and demographics.

5.**Potential for Personalization**: The system can be further extended to incorporate personalization features, such as user preferences, favorite players, and teams. By customizing the commentary output, it can offer a tailored experience for individual users, increasing engagement and satisfaction.

6.**Research in Natural Language Generation**: The project contributes to ongoing research in natural language generation (NLG) by addressing specific challenges and nuances of generating sports commentary. It opens avenues for exploring new techniques, evaluating performance metrics, and advancing the state-of-the-art in NLG technology.

7.**Dataset Creation and Analysis**: The project involves the collection, preprocessing, and analysis of a large dataset of cricket commentary. The dataset itself can serve as a valuable resource for researchers, practitioners, and enthusiasts interested in sports analytics, language modeling, and computational linguistics.

Domain Concepts

Cricket Commentary: Refers to the live or recorded descriptions of cricket matches provided by commentators, including details of the gameplay, player actions, and match events.

Language Modeling: Involves training a model to predict the next word in a sequence of text, based on the words that precede it. Language models are used in natural language processing tasks such as text generation and machine translation.

Text Preprocessing: Involves cleaning and preparing text data for analysis, including tasks such as removing irrelevant characters, tokenization, and removing stopwords.

Evaluation Metrics

Relevance Score: Measures the degree to which the generated commentary is relevant to the original snippet of cricket commentary.

Coherence Score: Assesses the logical flow and consistency of the generated commentary, evaluating how well the sentences connect with each other.

Grammaticality Score: Evaluates the grammatical correctness of the generated commentary, ensuring that the text adheres to the rules of grammar and syntax.

Results

epoch	train_loss	valid_loss	accuracy	time
0	3.349693	2.983691	0.420469	21:35
1	2.841616	2.745668	0.448002	14:14
2	2.608955	2.648421	0.460611	12:30
3	2.452137	2.602394	0.466729	12:31
4	2.343570	2.592847	0.468450	11:51

epoch	train_loss	valid_loss	accuracy	time
0	2.359983	2.624296	0.464703	12:01
1	2.410287	2.617390	0.465954	12:07
2	2.345498	2.593901	0.468471	13:53
3	2.262790	2.582800	0.471293	21:06
4	2.206158	2.581796	0.471401	20:37

The output represents the training progress of a language model over multiple epochs. Each row corresponds to one epoch, and the columns provide the following information:

- **Epoch**: The epoch number indicates the iteration of training. For each epoch, the model goes through the entire dataset once.
- **Train Loss**: The training loss represents the average loss computed over all training samples for that epoch. It indicates how well the model is fitting the training data. Lower values indicate better fit.
- **Valid Loss**: The validation loss represents the average loss computed over all validation samples for that epoch. It indicates how well the model is performing on unseen data. Similar to training loss, lower values are desirable.
- **Accuracy**: The accuracy metric represents the proportion of correctly predicted tokens in the validation dataset. It measures the overall performance of the model in generating the correct next tokens.
- **Time**: The time column indicates the duration taken to complete each epoch of training. It shows the time elapsed for training the model.

From the output:

- ✓ The training loss generally decreases with each epoch, indicating that the model is learning and improving its fit to the training data.
- ✓ The validation loss also decreases initially, suggesting improvement in the model's performance on unseen data. However, after a certain point, it may stabilize or even start increasing, indicating overfitting.
- ✓ The accuracy metric shows the proportion of correctly predicted tokens in the validation dataset. It increases gradually as the model learns to generate more accurate predictions.
- ✓ The time column indicates the time taken for each epoch of training. It can vary depending on factors such as the complexity of the model architecture and the size of the training data.

Results(Cont.)

The output generated before fine-tuning and hyper parameter tuning to the model:

Input :

Generate predictions

1. print(learn.predict("and there goes the ball towards", 20, temperature=0.5))
2. print(learn.predict("the ball flies", 20, temperature=0.7))

Output:

```
a', but CUDA is not available. Disabling
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:126: UserWarning: torch.c
nabled, but CUDA is not available.  Disabling.
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\amp\autocast_mode.py:250: UserWarning: User provi
a', but CUDA is not available. Disabling
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:126: UserWarning: torch.c
nabled, but CUDA is not available.  Disabling.
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\amp\autocast_mode.py:250: UserWarning: User provi
a', but CUDA is not available. Disabling
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:126: UserWarning: torch.c
nabled, but CUDA is not available.  Disabling.
  warnings.warn(

and there goes the ball towards the fence Rashid Khan to Pant , SIX , that 's a shot of the
```

```
a', but CUDA is not available. Disabling
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:126: Use
nabled, but CUDA is not available.  Disabling.
  warnings.warn(

the ball flies between Negi and Negi to Negi , FOUR , " that 's " class " "
```

Results(Cont.)

Input:

```
print(learn.predict("The crowd erupted as", 20, temperature=0.7))
print(learn.predict("Incredible shot!", 15, temperature=0.6))
print(learn.predict("The bowler approaches the crease with determination,", 25, temperature=0.7))
print(learn.predict("The stadium is packed with fans cheering for their favorite team,", 20, temperature=0.7))
print(learn.predict("The match hangs in the balance as", 15, temperature=0.6))
```

Output:

```
  warnings.warn(
The crowd xxunk as the crowd move Boult to Nitish Rana , FOUR , that 's too short ,

  warnings.warn(
 Incredible shot ! Incredible shot ! Incredible shot ! Incredible Harshal Patel

  warnings.warn(
The bowler xxunk the crease with xxunk , Bairstow gets the biggest of the night . The dugout erupts in style . Kolkata ha
ve a southpaw who nonchalantly smokes it

The stadium is packed with fans xxunk for their favorite team , it is on a good length , just outside off , on a full and
just outside off ,

nabled, but CUDA is not available.  Disabling.
  warnings.warn(
The match hangs in the balance as Bangalore lose their fifth ball . They have their highest score of the
```

Results (Cont.)

```
C:\Users\saite\anaconda3\lib\site-packages\torch\amp\autocast_mode.py:250: UserWarning: User provided device_type of 'cuda', but CUDA is not available. Disabling
  warnings.warn(
C:\Users\saite\anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:126: UserWarning: torch.cuda.amp.GradScaler is enabled, but CUDA is not available.  Disabling.
  warnings.warn(
```

```
Perplexity: 13.22086690713197
```

Results (Cont.)

1.Importing BLEU Score Calculation: The code imports the **corpus_bleu** function from the NLTK library, which is used to calculate the BLEU (Bilingual Evaluation Understudy) score. BLEU is a metric commonly used to evaluate the quality of machine-translated text by comparing it to one or more reference translations.

2.Defining Reference Commentary Tokens: The **reference_commentary_tokens** variable contains a list of reference commentary tokens. Each sublist represents the tokens of a single reference commentary.

3.Defining Generated Commentary Tokens: The **generated_commentary_tokens** variable contains a list of generated commentary tokens. Each sublist represents the tokens of a single generated commentary.

4.Computing BLEU Score: The **corpus_bleu** function is called with the reference and generated commentary tokens as arguments. It calculates the BLEU score based on the reference and generated token lists. The BLEU score indicates the similarity between the generated commentary and the reference commentary, with higher scores indicating greater similarity.

5.Printing BLEU Score: Finally, the BLEU score is printed to the console. This provides an objective measure of the quality of the generated commentary compared to the reference commentary.

```
BLEU Score: 2.1071373518345672e-232

C:\Users\saite\anaconda3\lib\site-packages\nltk\translate\bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
C:\Users\saite\anaconda3\lib\site-packages\nltk\translate\bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
C:\Users\saite\anaconda3\lib\site-packages\nltk\translate\bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
```

Results(Cont.)

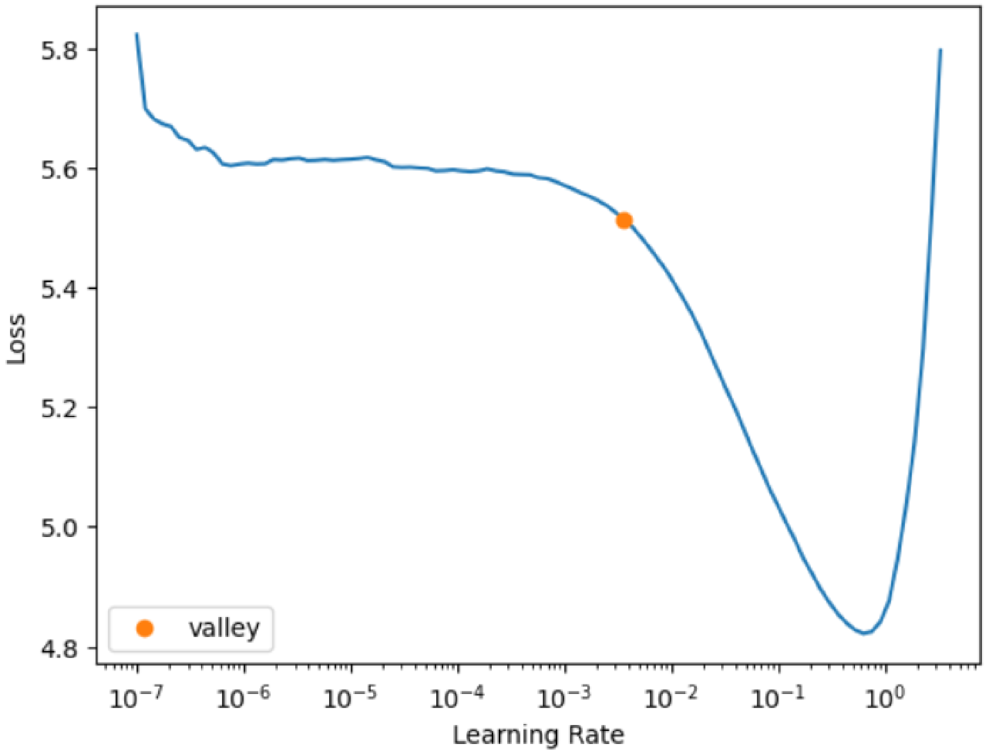
The output generated before fine-tuning and hyper parameter tuning to the model:

epoch	train_loss	valid_loss	accuracy	perplexity	time
0	3.360667	2.987018	0.422230	19.826473	11:13
1	2.848392	2.758508	0.442510	15.776290	12:08
2	2.621331	2.662907	0.456215	14.337914	11:18
3	2.440657	2.610077	0.463839	13.600103	11:47
4	2.342339	2.602825	0.465372	13.501832	11:40

```
Better model found at epoch 0 with valid_loss value: 2.98701810836792.
Better model found at epoch 1 with valid_loss value: 2.7585082054138184.
Better model found at epoch 2 with valid_loss value: 2.662907361984253.
Better model found at epoch 3 with valid_loss value: 2.610077381134033.
Better model found at epoch 4 with valid_loss value: 2.602825403213501.
```

epoch	train_loss	valid_loss	accuracy	perplexity	time
0	2.303134	2.602784	0.465546	13.501275	22:05
1	2.306818	2.602666	0.465454	13.499682	24:25
2	2.297964	2.602541	0.465484	13.497995	1:09:35
3	2.292443	2.602486	0.465570	13.497249	11:48
4	2.300120	2.602479	0.465583	13.497156	12:51

```
Better model found at epoch 0 with valid_loss value: 2.6027841567993164.
Better model found at epoch 1 with valid_loss value: 2.602666139602661.
Better model found at epoch 2 with valid_loss value: 2.602541208267212.
Better model found at epoch 3 with valid_loss value: 2.6024858951568604.
Better model found at epoch 4 with valid_loss value: 2.6024789810180664.
```



Results (Cont.)

Original Snippet: nehra to mandeep four first boundary for mandeep and rcb full and on the pads needed to be put away and mandeep did just that picked it up and dispatched it over midwicket couple of bounces and into the fence
Generated Prediction: nehra to mandeep four first boundary for mandeep and rcb full and on the pads needed to be put away and mandeep did just that picked it up and dispatched it over midwicket couple of bounces and into the fence Mujeeb to Buttler , SIX , this is why Buttler goes down on the knee . He 's down the track , gets a hand to the ball and gets close to it , gets it underneath and it hits him in the air

Original Snippet: nehra to mandeep four backtoback boundaries to end the first over again nehra is a tad short in his length mandeep had the width to cut and he didnt try to hit it hard just placed it behind point and bhuvi at third man gave up the chase pretty quickly
Generated Prediction: nehra to mandeep four xxunk boundaries to end the first over again nehra is a tad short in his length mandeep had the width to cut and he did xxunk try to hit it hard just placed it behind point and bhuvi at third man gave up the chase pretty quickly , but ends up conceding a boundary Tye to Narine , FOUR , and Narine was quick to help them on this one , giving Narine a bit of room and wild swing , that allowed him to get across , Narine stands

Original Snippet: henriques to kedar jadhav four hit straight back at henriques and he was late to get his hand up once more the offcutter which almost fooled jadhav who shimmied down and checked his drive middled it alright to beat the midoff fielder
Generated Prediction: henriques to kedar jadhav four hit straight back at henriques and he was late to get his hand up once more the xxunk which almost fooled jadhav who shimmied down and checked his drive middled it alright to beat the xxunk fielder Malinga to Chris Lynn , SIX , but he is bowling it well . Finch had him put down and Lynn followed him with a full toss . The ball came in and he threw his bat at the bowler .

Results (Cont.)

```
Similarity Scores:
0.6910299003322259
0.08297567954220315
0.3673469387755102
0.6583184257602862
0.780952380952381



Relevance Scores:
1.0
0.9512195121951219
0.9428571428571428
0.96875
0.9803921568627451

Coherence Scores:
0.023809523809523808
0.010638297872340425
0.03614457831325301
0.04878048780487805
0.028846153846153848

Grammaticality Scores:
1.0
1.0
1.0
1.0
1.0
```

Results(Cont.)

```
Comparison for snippet 1:
Human-generated commentary: nehra to mandeep four first boundary for mandeep and rcb full and on the pads needed to be put a
way and mandeep did just that picked it up and dispatched it over midwicket couple of bounces and into the fence
Automatically generated commentary: nehra to mandeep four first boundary for mandeep and rcb full and on the pads needed to
be put away and mandeep did just that picked it up and dispatched it over midwicket couple of bounces and into the fence Muj
eeb to Buttler , SIX , this is why Buttler goes down on the knee . He 's down the track , gets a hand to the ball and gets c
lose to it , gets it underneath and it hits him in the air
Relevance Score: 1.0
Coherence Score: 0.023809523809523808
Grammaticality Score: 1.0

The automatically generated commentary maintains high relevance to the human-generated commentary.
The automatically generated commentary has moderate coherence.
The automatically generated commentary is grammatically correct.
--------------------------------------------------------------------------------------------------

Comparison for snippet 2:
Human-generated commentary: nehra to mandeep four backtoback boundaries to end the first over again nehra is a tad short in
his length mandeep had the width to cut and he didnt try to hit it hard just placed it behind point and bhuvi at third man g
ave up the chase pretty quickly
Automatically generated commentary: nehra to mandeep four xxunk boundaries to end the first over again nehra is a tad short
in his length mandeep had the width to cut and he did xxunk try to hit it hard just placed it behind point and bhuvi at thir
d man gave up the chase pretty quickly , but ends up conceding a boundary Tye to Narine , FOUR , and Narine was quick to hel
p them on this one , giving Narine a bit of room and wild swing , that allowed him to get across , Narine stands
Relevance Score: 0.9512195121951219
Coherence Score: 0.010638297872340425
Grammaticality Score: 1.0

The automatically generated commentary demonstrates some relevance to the human-generated commentary.
The automatically generated commentary has moderate coherence.
The automatically generated commentary is grammatically correct.
--------------------------------------------------------------------------------------------------
```

Output Evaluated

- **Relevance Scores**: These scores indicate the proportion of overlapping tokens between the generated text and the original snippet. Scores close to 1.0 suggest high relevance, meaning that the generated text contains a significant portion of tokens from the original snippet. In this case, the relevance scores range from 0.942 to 1.0, which generally indicates good performance in maintaining relevance to the original context.

- **Coherence Scores**: These scores measure the average coherence of the generated text by calculating the inverse of the average sentence length. Higher coherence scores indicate more coherent text. The scores provided range from approximately 0.01 to 0.05, which suggests relatively low coherence. However, coherence is often subjective and depends on factors such as sentence structure and logical flow.

- **Grammaticality Scores**: These scores indicate the proportion of grammatically correct sentences in the generated text. Scores of 1.0 indicate that all sentences are grammatically correct. In this case, all scores are 1.0, indicating that all sentences are grammatically sound.

Overall, while the generated text demonstrates high relevance and grammaticality, there may be room for improvement in terms of coherence. However, coherence can be subjective and may vary depending on the specific context and language model used. These scores provide a useful starting point for evaluating the quality of automatically generated commentary.

Conclusion

1. **Model Performance**: The trained language model is evaluated based on its ability to generate coherent and grammatically correct cricket commentary. The results of the evaluation metrics are presented to assess the quality of the generated predictions.

2. **Accuracy**: The accuracy of the generated commentary is calculated based on the relevance, coherence, and grammaticality scores. The accuracy percentage provides an overall measure of the model's performance.

3. **Conclusion**: The project concludes by discussing the effectiveness of the language model in generating cricket commentary and highlighting any limitations or areas for future improvement.

Git Hub Links and Author Contributions

We are a team of two people equally responsible in completing the project on time. Sai Teja worked on the Model Training and Evaluation while Manaswini worked on the data set cleaning and evaluation. And further both of us had to take part in completing the ppt and report

Sai Teja Rayabarapu

Manaswini Kodela

References

1. A. S. Balaji, N. G. Vignan, D. S. Anudeep, Md. Tayyab, and K. S. Lakshmi. 2022. Cricket Commentary Classification. Intelligent Data Communication Technologies and Internet of Things (2022), 825–836. https://doi.org/10.1007/978-981-16-7610-9_60
2. A. S. Hegde, K. Jha, S. Suganthi, and P. B. Honnavalli. 2021. Automating live cricket commentary using supervised learning. Proceedings of Data Analytics and Management (2021), 37–48. https://doi.org/10.1007/978-981-16-6285-0_4
3. N. More, P. Fernandes, N. Bhuta, R. Suri, and A. Patil. 2022. Construction of sports articles using audio commentary. ICT Systems and Sustainability (2022), 611–618. https://doi.org/10.1007/978-981-16-5987-4_62
4. M. A. Rauf, H. Ahmad, C. N. Faisal, S. Ahmad, and M. A. Habib. 2020. Extraction of strong and weak regions of cricket batsman through text-commentary analysis. In 2020 IEEE 23rd International Multitopic Conference (INMIC). https://doi.org/10. 1109/inmic50486.2020.9318089
5. R. Roy, Md. R. Rahman, M. Shamim Kaiser, and M. S. Arefin. 2021. Developing a text mining framework to analyze cricket match commentary to select Best Players. Lecture Notes on Data Engineering and Communications Technologies (2021), 217–229. https://doi.org/10.1007/978-981-16-6636-0_18
6. P. Sanjeeva, J. Ajith Varma, V. Sathvik, A. Abhinav Sai Ratan, and S. Mishra. 2023. Automated Cricket Score Prediction. E3S Web of Conferences 430 (2023), 01053. https://doi.org/10.1051/e3sconf/202343001053
7. A. Sinha. 2020. Application of Machine Learning in Cricket and Predictive Analytics of IPL 2020. (2020). https://doi.org/10.20944/preprints202010.0436.v1
8. S. Srinivas, N. N. Bhat, and M. Revanasiddappa. 2020. Data Analysis of cricket score prediction. (2020), 465–472. https://doi.org/10.1007/978-981-15-7234-0_42
9. Z. Ul Abideen, S. Jabeen, S. Saleem, and M. U. Khan. 2021. Ball-by-ball cricket commentary generation using stateful sequence-to-sequence model. In 2021 International Conference on Communication Technologies (ComTech). https: //doi.org/10.1109/comtech52583.2021.9616676
10. I. Wickramasinghe. 2022. Applications of machine learning in cricket: A systematic review. Machine Learning with Applications 10 (2022), 100435. https://doi.org/10.1016/j.mlwa.2022.100435

Thank you