MIT SLOAN SPORTS ANALYTICS CONFERENCE presented by ESPN 42 ANALYTICS MARCH 4-5, 2022

Sports Narrative Enhancement with Natural Language Generation

Henry Wang, Amazon ML Solutions Lab, yuanhenw@amazon.com
Saman Sarraf, Amazon ML Solutions Lab, ssarraf@amazon.com,
Arbi Tamrazian, Fox Sports, Arbi.Tamrazian@fox.com

1. Introduction

Sports broadcasters are increasingly sharing statistical insights throughout the game to tell a richer story for the audience. Thanks to abundant data and advanced statistics, broadcasters can quickly tell stories and make comparisons between teams and players to keep viewers engaged. To keep up with the fast-paced nature of many games, broadcasters rely on template-generated narratives to speak about in-game stats in real time. When milestone event happens, these rule-based templates "stitch" relevant tabular information and create narratives with fixed sentence structures.

Because of the fixed structure, however, these narratives often sound rigid and are hard to understand, especially when lots of information is concatenated into long sentences. Commentators may choose to ignore these narratives if their meanings are hard to grasp. As a result, exciting stats may not come through to the audience. Additionally, as data volume rises, the amounts of efforts required on building and maintaining templates also increase. They have to be manually updated constantly to reflect the changes.

To address this issue, we design and build an end-to-end machine learning pipeline using natural language generation, a technique to generate natural language descriptions from structured data. The pipeline is trained to understand the semantic meaning of inputs, and can be expanded to include new statistics and applied to other sports through fine-tuning with a few hundred samples. This enables broadcasters to produce more natural-sounding narratives and easily scale narrative-generation engines. The generated narratives can also be used in social media and push notifications. By coupling narratives with the highlighted game clips, broadcasters can ensure fans do not miss exciting moments from their favorite teams and players.

The rest of the paper is organized as follows: in Section 2 we describe the two-step modeling approach, the dataset and the evaluation metrics; Section 3 highlights the sample results achieved with the solution; in Section 4 we summarize the contributions followed by discussions on future improvements.

2. Methodology

Our method consists of two stages. Figure 1 describes the overview of the two-stage solution. In first stage, we convert tabular data into structured sentences by leveraging large language models. In second stage, we rewrite the sentences and enhance their readabilities through various natural language generation techniques.

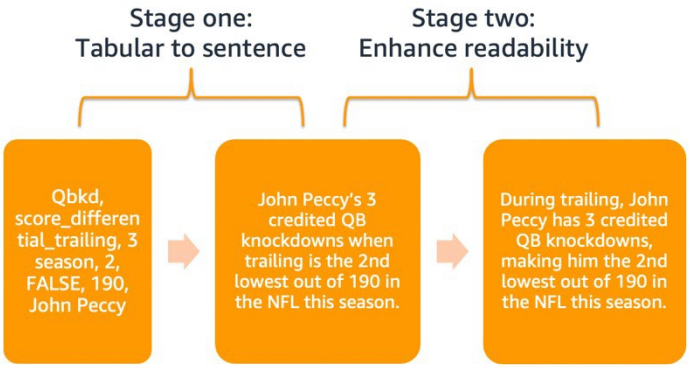


Figure 1: Two-stage solution overview

2.1. Template to ML

Natural language generation (NLG) is the use of machine learning to produce written or spoken narratives from a dataset, which is usually in a numerical or a structured tabular form that is difficult for humans to directly interpret. The technology has been used in area such as voice assistants (Alexa, Siri), chatbots and email/text auto-completion [1]. We base our solution on NLG because of its capability to generate human-readable narratives.

The first phase of the NLG-based narrative generation solution leverages tabular features, including player and team names, metrics, and game situations. These features are paired with their target sequences, which are generated using rule-based templates. The goal here is to take the tabular features and generate candidate narratives containing all the relevant information.

2.1.1 Dataset

We synthetically generate a dataset using rule-based methodology. The dataset is generated by permuting different statistics, feature values, and team and player names, and includes more than 57,000 samples composed of eight features. For each tabular sample, we generate the corresponding narrative from a rule-based template as target. We randomly shuffle and divide the dataset into training, validation, and testing sets based on an 80/10/10 split for training and fine-tuning our model.

Table 1 shows examples of the raw data used – each row represents a sample, and each column represents the relevant information associated with the sample, including the statistic name, value for the statistic, situation that the statistic is calculated upon, etc. For this manuscript, we replace actual team and player names with generic names: team Bobcats and player John Peccy.

Statistic	Situation	Value	Time frame	Rank	Rank Order	Population	Team name / Player name
rec_td	stadium_retractable_dome	5	season	7	True	32	Bobcats
qbkd	score_differential_trailing	3	season	2	False	190	John Peccy

Table 1: example of tabular features

For each row, the raw tabular features are concatenated to form a text sequence. Table 2 shows examples of the text sequences used as inputs and the associated narratives from the rule-based template as outputs.

Template input	Template output
rec_td stadium_retractable_dome 5 season 7 TRUE 32 Bobcats	Bobcats' 5 caught passes for touchdowns when playing in a retractable roof is the 7th highest out of 32 in the NFL this season.
qbkd score_differential_trailing 3 season 2 FALSE 190 John Peccy	John Peccy's 3 credited QB knockdowns when trailing is the 2nd lowest out of 190 in the NFL this season.

Table 2: example of tabular inputs to template narratives

2.1.2 Methods

One effective strategy in training machine learning models is called transfer learning [2]. It focuses on reusing and improving an existing model developed for a different but related task. For example, a model trained to recognize sedans can be used as a starting point for training a sports car-detection model. The technique is popular in deep learning domains such as computer vision (CV) and natural language processing (NLP). By leveraging the generalizable features learned by prior models, it is significantly faster for the new models to achieve satisfying results with limited amounts of resources.

Because transfer learning has been proved effective, we utilize a language model called Text-To-Text Transfer Transformer (T5) [3], which was pretrained on the open-source dataset Colossal Clean Crawled Corpus (C4). T5 achieves state-of-the-art results on many NLP benchmarks and is flexible to be fine-tuned to different custom NLP use cases. To fine-tune the T5 model for our task, we concatenate tabular features into text sequences as inputs, and use the template-generated statements as labels. For example, table 3 is translated into the text sequence *"Team Bobcats, prss, 4, score_differential_leading, 7"*.

Team name	Metric	Value	Situation	Rank
Bobcats	prss	4	score_differential_leading	7

Table 3: example of tabular features

The corresponding template statement - *"The Bobcats' 4 total times of pressuring the quarterback when leading is the 7th highest in the NFL this season"* - is passed in as the target output. With thousands of such examples, the T5 model can be fine-tuned to generate statements similar to the template. Since it learns the positional meaning of the input, the model is able to generalize to previously unseen data, making it extensible to fresh players and newly created metrics.

2.1.3 Evaluation metric

We use bilingual evaluation understudy (BLEU) [4], a popular metric for machine-generated text quality evaluation, to quantitatively measure the similarity between model outputs (candidates) and template outputs (references).

BLEU score ranges from 0 to 100. The more words from candidate text match the words from reference text, the higher is the BLEU score. We use standard implementation of sentence BLEU score, which is an average of 1-gram, 2-gram, 3-gram and 4-gram BLEU scores. An individual N-gram score evaluates gram-matching of a specific order, such as single words (1-gram) or word pairs (2-gram). This way we ensure two sentences match not just on single words but also phrases.

After fine-tuning on a few thousand sentences, the T5 model is able to achieve a BLEU score above 99 on the test set, an indication that most of the generated sentences are identical to template-generated sentences. This again echoes the usefulness of leveraging pre-trained models trained on abundantly available unlabeled-text for different downstream tasks.

2.2. Improve comprehensibility

While the fine-tuned T5 model is able to mimic and generalize the templates rules, it still suffers from the same drawback – the resulting narratives are sometimes verbose and awkward to read since they follow the same pre-defined sentence structure. This can lead to confusion for the broadcasters and fans. To address this, our second phase of modeling employs language models to enhance the readability of narratives from first phase modeling. The goal is to make the narratives sound more natural, hence making it easier for live commentating and for fans to digest.

2.2.1 Methods – back translation

One way to replace unnatural words in sentences is through back translation [5]. Back translation is a two-step translation method. It first translates a sentence into another language and then translates the sentence back to its original language. It is a technique used mostly for text data augmentation, i.e., increasing the variety of original text. For our use case, we find the pre-trained translation models can help fix certain rule-induced mistakes in the original sentence, e.g., a singular noun may be corrected to a plural. The models may also choose more natural-sounding phrases in place of jargons and puns. This approach gives us an automated way to improve readability for our generated sentences. Figure 2 shows what a narrative can look like after back translation. The original long sentence is split to two parts, making it easier to read and understand.

For back translation, we used pre-trained neural machine translation models from fairseq [6], an open-source toolkit for sequence and language modeling. It is important to note that the choice of intermediate language matters for this approach. Languages that share similar syntax and word roots are usually good candidates as they tend to preserve the meaning of the original language. For example, we found German and Spanish generate more meaningful and accurate back translation results compared to Chinese and Russian. This is likely due to the reason that American Football contains some unique terms that are hard to find in languages from other cultures. We also notice undesirable occasions where some information is lost during the back translation process. For example, the “when leading” context is left out in the resulting narrative. We discuss the solution in section 2.3.

Figure 2: sample result of a narrative going through back translation.

2.2.1 Methods - paraphrasing

An alternative NLP approach is called paraphrasing - a technique that aims to express semantically similar narratives in different forms [7]. We employ a pretrained T5 model [8], which is fine-tuned for paraphrasing purposes using open-sourced paraphraser dataset - PAWS [9]. PAWS contains over 100,000 human labels on whether a phrase pair is paragraph with each other. Since the model is auto-regressive and we want to have a set of diversified texts, we employ Top-K and Top-P sampling techniques [10, 11], which allows the top most probable candidate words to be sampled during text generation. This ensures that our paraphrasing model generates several candidates for a given narrative with slightly different contents. We then choose the candidate that best fits business requirements. Figure 3 shows an example of the paraphrasing outputs against a sample narrative.

Figure 3: Sample of candidate narratives generated from the paraphraser model. Dark texts highlight the differences in generated narratives against the original narrative.

2.2.2. Model evaluation

Quantitatively evaluating how "natural" a sentence sounds is an ongoing challenge in NLP community. While there are classic methods such as Flesch-Kincaid test and Dale-Chall score [12], which measure the text lengths and difficulty of words, they don't take into account of sentence structures, hence less suitable to measure which rewrite is more effective. For our work, we leverage metric called Perplexity [13], which is commonly used for evaluating language models.

Given a tokenized sentence $X = (x_1, x_2, \ldots, x_t)$, the perplexity of X is,

$$PPL(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

Where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i-th token conditioned on the preceding tokens. Intuitively it measures the model's ability to predict the sentence, or as a proxy measure of how "surprised" a language model is at a sentence. In other words, it measures how common an evaluation sentence is among text corpus used to train a language model, which can be used to compare the quality of different sentences. For language models such as GPT2 [14], it typically assigns low perplexity score to real and syntactically correct sentences and high perplexity to fake, incorrect, or awkwardly-structured sentences. For example, GPT2 will assign a lower perplexity score to a sentence like "Can you do it?" and assign a higher perplexity score to a sentence like "Can you does it?". Using perplexity score, we are able to compare the quality of generated sentences sharing similar semantic meanings and output the one with the lowest perplexity score.

2.3. Solution Architecture

In order to productionize the machine learning solution, we need to ensure the models meet certain criteria. First, the final narratives must contain the key information specified in the original sentences. Secondly, the final narratives shouldn't be harder to read than the original ones. Because of the probabilistic nature of text generation with NLG models and possibilities that information gets left out during back translation or paraphrasing, we propose an end-to-end workflow that consists of two major components: 1) replacing the current ruled-based approach with the fine-tuned T5 model and 2) enhancing the generated narratives through a multistep ML-based approach.

As illustrated in the Figure 3, the fine-tuned T5 ML model generates the narratives (green blocks). Next, the narratives are passed through the backtranslation model as an attempt to produce enhanced narratives. A fixed tabular feature to keywords dictionary is passed to check if the resulting narratives contain the keywords. If the back-translated results include the necessary keywords and their perplexity scores are lower compared to the T5 model outputs, they are used as the final outputs. Otherwise, we pass the T5 model outputs through the paraphrasing model and apply the same condition check. If none of our enhancement models reduces the perplexity score, we simply output the T5 model outputs. Through this workflow, we ensure all the required features are captured and improve the readability of the sentence when appropriate, maximizing the benefit ML can bring to the existing solution.

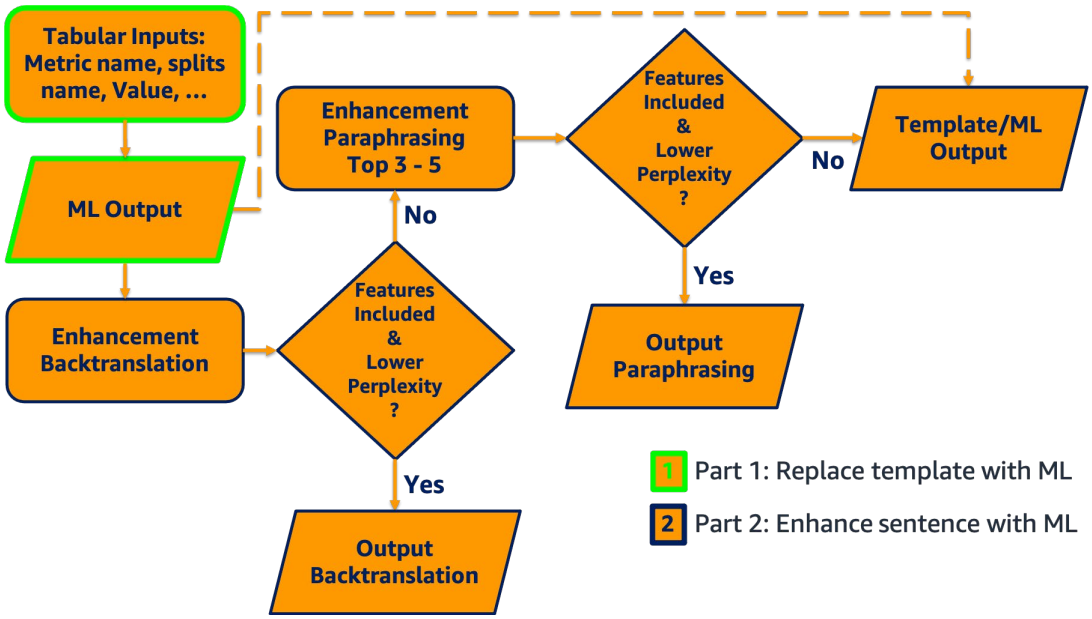
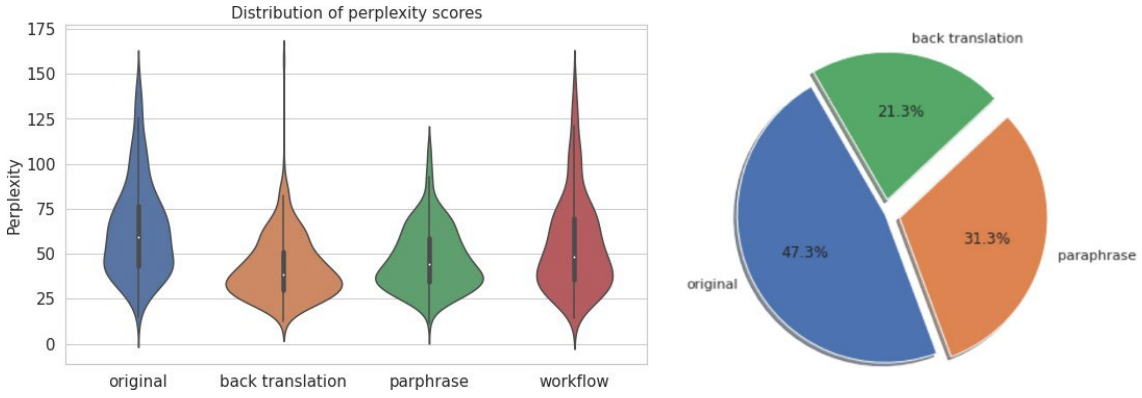


Figure 4: Workflow of final solution

3. Results

To test model performances, we randomly select 300 sample narratives and apply back translation, paraphrasing and solution pipeline separately. Figure 4.a shows distribution of perplexity scores for resulted narratives after each method. Original narratives have perplexity score of 63.9(±26.3), back translation reduces perplexity to 42.5(±17.4), paraphrasing reduces perplexity to 47.3(±17.4) and final narratives out of workflow have perplexity score of 55.7(±26.5).



(a) *(b)*

Figure 5: (a) Perplexity distribution of output narratives after each method. Original represents narratives generated from Part1 of the workflow; back translation represents narratives after passing original to back translation; paraphrase represents narratives after passing original to paraphraser model and choosing the candidate with lowest perplexity; workflow represents the output after passing through the final solution. (b) Final output proportion from each method after going through the solution workflow for the 300 sample narratives. Back translation and paraphrase combined rewrite over half of the narratives while containing original key information.

While back translation and paraphrasing reduce perplexity score the most, as mentioned in Section 2, they potentially suffer from leaving out or misplacing information during the rewrite. For example, *"pressuring the quarterbacks"* can sometimes be rephrased to *"hitting the quarterbacks"* or *"pushing the quarterbacks"*, which distort the meanings of original narratives. By leveraging the workflow, we make sure the keywords are contained in the final output narratives. Figure 4.b shows the breakdown of techniques used for final narratives from the workflow. We notice that a significant portion of selections comprise original narratives. Closer examination reveals that although some of the rewrites are semantically correct, they don't contain the specified keywords, which turn out to be too strict of constraints. For example, "*pass incomplete*" feature maps to keywords including "*fail, complete, pass*". Although "*incomplete passes*" means the same, it doesn't contain all keywords, hence doesn't get selected as the final output. One way to improve is to include synonyms of keywords and features, and the final narrative will meet the requirement as long as one of the synonyms exist. We leave this as part of the future work.

Table 4 compares some of the narratives generated from rule-based template and from our ML pipeline. With a 13% average reduction in perplexity, broadcasters can use these narratives live during the games and automatically send notifications to fans!

Template Narrative	Solution Narrative
The Bobcats 8 secured receptions in September is the 7th lowest out of 32 in the NFL this season.	In September, the Bobcats secured 8 receptions -- the 7th lowest of 139 in the NFL this season.
The Bobcats 10 catches for first downs when trailing inside two minutes is the 7th highest out of 32 in the NFL this season.	The 10 catches of the Bobcats for first downs when trailing inside two minutes is the 7th most in the NFL out of 105 this season.
The Bobcats has blitzed the passer 12 times against the Bears is the 7th highest out of 32 in the NFL this season.	The Bobcats have blitzed the passer 12 times against the Bears, which is the 7th highest out of 139 in the NFL this season.

Table 4: Comparisons of template-generated narratives and solution-generated narratives

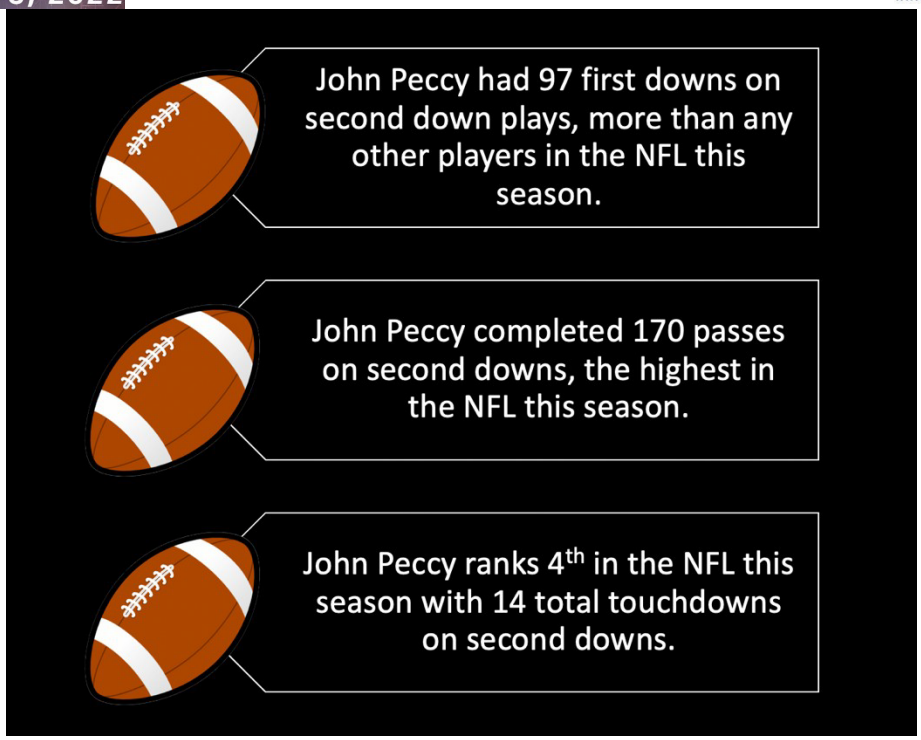


Figure 6: Sample narratives to be sent through newsfeed or Twitter

4. Conclusion

In this work, we describe how to build an end-to-end Machine Learning solution that is able to convert in-game tabular statistics into natural sounding narratives. Compared to the template-based approach, our solution significantly improves the readability of sentences while ensuring the key information is preserved. Because the solution is built on top of language models that are pre-trained on huge text corpus, additional metrics and game situations can get passed in directly to generate the desired outputs. The extensibility also enables solution to be transferred to other sports by simply fine-tuning the model with sample narratives. These capabilities allow the model to scale and adapt to broadcasters' future business needs.

References

[1] Gatt, Albert, and Emiel Krahmer. "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation." *Journal of Artificial Intelligence Research* 61 (2018): 65-170.
[2] Tan, Chuanqi, et al. "A survey on deep transfer learning." *International conference on artificial neural networks*. Springer, Cham, 2018.
[3] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *arXiv preprint arXiv:1910.10683* (2019).
[4] Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002.
[5] Edunov, Sergey, et al. "Understanding back-translation at scale." *arXiv preprint arXiv:1808.09381* (2018).
[6] Ott, Myle, et al. "fairseq: A fast, extensible toolkit for sequence modeling." *arXiv preprint arXiv:1904.01038* (2019).
[7] Bannard, Colin, and Chris Callison-Burch. "Paraphrasing with bilingual parallel corpora." *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 2005.
[8] Sai Vamsi Alisetti, Paraphrase Generator, 2021, Github, https://github.com/Vamsi995/Paraphrase-Generator
[9] Zhang, Yuan, Jason Baldridge, and Luheng He. "PAWS: Paraphrase adversaries from word scrambling." *arXiv preprint arXiv:1904.01130* (2019).
[10] Fan, Angela, Mike Lewis, and Yann Dauphin. "Hierarchical neural story generation." *arXiv preprint arXiv:1805.04833* (2018).
[11] Holtzman, Ari, et al. "The curious case of neural text degeneration." *arXiv preprint arXiv:1904.09751* (2019).
[12] Fry, Edward. "A readability formula that saves time." *Journal of reading* 11.7 (1968): 513-578.
[13] Jelinek, Fred, et al. "Perplexity—a measure of the difficulty of speech recognition tasks." *The Journal of the Acoustical Society of America* 62.S1 (1977): S63-S63.
[14] Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.