ErrorSpotter: A Visual Analytics System for Spotting Errors in AI Generated Sports Summaries

Zhehao Yang, Ken Wakita

近年の自然言語生成技術の発展に伴い、ニューラルネットワークモデルを用いて事実を表すデータから，叙述的内容を適切な順番で生成できるようになった．しかし，生成されたテキストは、事実に基づかない記述を含んだり，特筆すべき事実に触れないなどの課題がある．本論文は，AI が生成したスポーツの試合についての要約記事の誤りを発見するための視覚的分析システムを提案する．ニューラルネットワークモデルに入力した試合のデータとそこから生成された記事内の語句を関連づけるため，文書に attention 値を組み合わせて表示することで，試合のデータとニューラルネットワークが生成した単語列の関係をわかりやすく提示する．このようなインタラクションを通じて，フィルタリング，誤りの検出，関連するデータについての調査を可能にし，記事の潜在的な誤りを指摘し，試合の事実に沿って検証できる．

Recent improvements in natural language generation have enabled neural network models to generate descriptive texts with modeling what to say and in what order. However, there are still erroneous texts remaining and other remarkable data facts missing in the texts. In this paper, we present one erroneous text spotting visualization system for AI generated sports summaries. We combine intuitive visualizations for attention-augmented documents, which illustrates the connection between the input game data and the output words generated by neural network model. Through interactions, our system enables attention filtering, error spotting, and data facts exploring that assist users in locating and verifying potential erroneous texts.

1 Introduction

Over the past several years, neural text generation systems have shown impressive performance on tasks such as machine translation and text generation. Meantime, more and more advanced summary generation models were developed [9, 13] where attention mechanism [1] has been widely used in. Attention mechanism has been the most intuitive intermediate delivery from neural network models: by exporting vectors of attention weights corresponding to the encoding units in encoder-decoder models [2, 11], attention approximates where the models are paying attention to, hence assists in debugging common problems like repetition and copying in black-box models [4]. In text generation tasks, attention visualizations have effectively helped researchers to verify alignments and explore attention correlation in displaying attention distribution, amplifying tiny differences and detecting potential erroneous output.

Despite producing overall fluent text, up-to-date neural systems still have difficulty capturing all the key points and generating a perfect summary that satisfies everyone. In other words, there are still erroneous text remaining and mentionable data facts missing in neural model generated text. Now that the machine is not smart enough to fix this problem by itself, we can take advantage of attention mechanism and visualization to fix the text error

ErrorSpotter: 自動生成されるスポーツ記事中の誤りを発見するための視覚的分析システム

楊 哲皓　脇田 建, 東京工業大学情報理工学院, School of Computing, Tokyo Institute of Technology.

issue on our own.

Therefore, we design a visual analytics system for spotting errors in NLG tasks. We extend text heatmap to intuitively display documents as well as potential erroneous text and utilize 2D heatmap to reveal the correlation between attention weight and each generated token. Through interaction, we enable the users to explore reliable game data as reference to verify spotted erroneous text.

We conclude our study with discussions on how the visual design can be improved and extended for other domains and analysis tasks.

2 Related Work

2. 1 Natural Language Generation

As the process of generating phrases and sentences in the form of natural language, Natural Language Generation (NLG) attracts a growing interest with the rapid development of Artificial Intelligence. There are several types of tasks that NLG can contribute to, such as question answering, summarizing [3], and even the computational humor generating [10]. The ErrorSpotter is targeting at the field of generating a natural language text that describes and summarizes sports data. Recent achievements in sports game summary generation [13] indicate the significant impact of Natural Language Generation on descriptive summary generation. Followed work [9] improved generation quality by explicitly modeling content selection (*what to say*) and planning (*in what order*) within a neural architecture.

2. 2 Attention Visualization for NLG Tasks

Since *attention mechanism* was first proposed in the NLG field [1], researchers never stop their pace to improve and utilize this emphasis embedded mechanism. Attention visualization plays an essential role in helping analysts explore and evaluate their results in NLG tasks. Text Heatmap presents the overview of generated output text with each text token colored in terms of its attention value [6]. 2D Heatmap shows a correlation between input and output tokens by mapping these tokens with different color weights in a tabular space [1]. It is easy to cross-compare the relationships between two tokens. However, a disadvantage with such 2D matrices is that they take too much space. Our work takes advantage of multiple conventional visualization methods to provide an interactive, intuitive, and easily interpretable visualizer of attention model.

2. 3 Error Spotting System

In this section, we discuss common types of issue occurring in NLG texts and corresponding error spotting systems. Here are the three main issue types we classify :

Grammatical Issues include, but are not limited to: spelling errors, unwanted words, missing words, prepositional errors, punctuation errors, and many of the grammatical errors. Grammatical Error Correction is typically formulated as a sentence correction task of fixing such grammatical issues. Recent Grammatical Error Correction Systems [8] are showing great performance on the test of CoNLL-2014 Shared Task which is the most widely used database to benchmark Grammatical Error Correction [7].

Artificial Issues represent the problem of having the indistinguishable generated text from human-written text to non-expert readers such as fake comments, false articles, and misleading reviews, which might cause the abuse of text generation system. One visual detection tool, GLTR [5], was developed by applying a suite of baseline statistical methods to support humans in detecting generation artifacts which are the texts generated by a model.

Factual Issues refers to the problem of having unfaithful data facts within generated textural output which are inconsistent with the truth. Factual issues might also occur along with grammatical issues and artificial issues. Besides, a minor erroneous data fact hidden in long sentence tends to be neglected without the corresponding check from source data set. These characteristics potentially increase the difficulty of detecting factual issue related errors. The information extraction system which can achieve the goal of distinguishing erroneous data facts through extracting entity, number, and types from NBA sports summaries was developed as the evaluation metrics to automatically evaluate output generation performance [13].

In this paper, we mainly focus on addressing factual issues in NLG task instead of grammatical issues and artificial issues. In the next section, we will discuss our error spotting system with more details.

3 System

3. 1 Data and Model

The data we used in this system is ROTOWIRE [13], one dataset of NBA basketball game summaries, paired with the corresponding *box-score table*, which is a structured summary listing score as well as individual and team achievements in a sports competition. The data set contains 3,398 summaries which are all professionally written by sports editors with an average length of 337 words. The proper summary amount and summary lengths make this data set ideally suited for natural language generation.

The model we used is Puduppully and others' *data-to-text* generation model [9]. This model generates a content plan highlighting which information should be mentioned and in which order and then generates the document while taking the content plan into account. The content plan is a set of game-related data record, which can be represented as $s = \{r_j\}_{j=1}^{J}$. For each content plan item $r \in s$, there are four attributes, which are $r.t$, $r.e$, $r.v$ and $r.s$ presenting a record's type, entity, value, and side, respectively. For example, one content plan might have a record r such that ($r.t$ = PTS, $r.e$ = LeBron James, $r.p$ = 25, $r.s$ = AWAY) where the type of record is Points, the entity is LeBron James, the value is 25 and the side of this record belongs to away team. This innovative method enables the model to achieve great performance improvements both in terms of the number of relevant facts contained in the output text, and the order according to which these data facts are presented.

3. 2 Design Considerations

To design a visual analytics system spotting errors with high flexibility, explorability, and readability, we aim to achieve three primary goals:

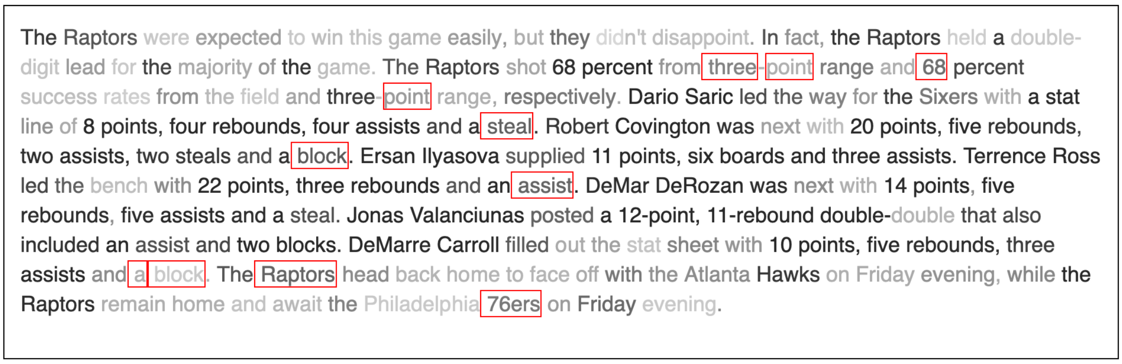
DC1 (Error Hint) Intuitive hints of automatically showing potential erroneous text are of great efficiency in error spotting system so that users could easily pay attention to the targeted word and make further analysis like model evaluation and erroneous text revision.

DC2 (Exploration) As an error spotting system, the ability of visually displaying reliable data facts is desired for users to do data exploration. This feature helps provide references to distinguish between erroneous text and correct text.

DC3 (Overview and Focus) The system should offer the overview of attention distribution. Meanwhile, it is also possible to inspect detailed information on focused word.

3. 3 System Overview

In this study, our system takes advantages of the strength of text heatmap and 2D heatmap - both are commonly used in NLG tasks to lower the learning curve. As shown in Figure 1, AI-generated



The Raptors were expected to win this game easily, but they didn't disappoint. In fact, the Raptors held a double-digit lead for the majority of the game. The Raptors shot 68 percent from three-point range and 68 percent success rates from the field and three-point range, respectively. Dario Saric led the way for the Sixers with a stat line of 8 points, four rebounds, four assists and a steal. Robert Covington was next with 20 points, five rebounds, two assists, two steals and a block. Ersan Ilyasova supplied 11 points, six boards and three assists. Terrence Ross led the bench with 22 points, three rebounds and an assist. DeMar DeRozan was next with 14 points, five rebounds, five assists and a steal. Jonas Valanciunas posted a 12-point, 11-rebound double-double that also included an assist and two blocks. DeMarre Carroll filled out the stat sheet with 10 points, five rebounds, three assists and a block. The Raptors head back home to face off with the Atlanta Hawks on Friday evening, while the Raptors remain home and await the Philadelphia 76ers on Friday evening.

図 1 **Text Heatmap of AI generated summary with each word colored in terms of maximum attention value and potential erroneous words wrapped in red box**

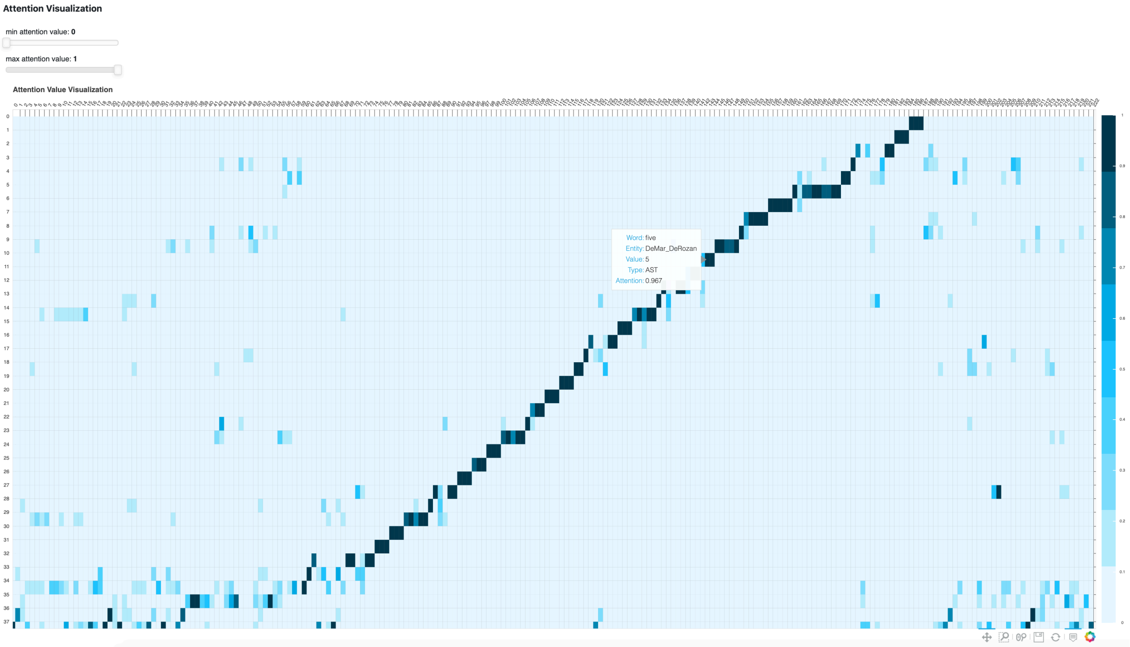


図 2 **2D Heatmap. The x-axis and y-axis of the plot correspond to the words in generated summary and the content plan items, respectively. Each cell $C_{i,j}$shows the attention value of the i-th generated word for the j-th content plan item, in gradient scale from light blue to deep blue.**

game summary is displayed where each word is colored in terms of the maximum attention value among each content plan item. The color of each word is based on a gradient scale from white to black, with deeper color representing more significant attention value. In dealing with error hint (DC1), some words are wrapped by red boxes revealing high possibility of being erroneous texts, where users should pay more attention to. To address exploration issue (DC2), the ErrorSpotter offers game-related statistics exploration function where you can find relevant data facts to verify the erroneous text hints.

In regard to overview and focus (DC3), an interactive 2D heatmap is designed as shown in Figure 2, which provides an intuitive way to inspect correlation between input data and output generated summary by visualizing the attention value between generated words $\{w_i\}_{i=1}^{I}$ and content plan records $\{r_j\}_{j=1}^{J}$. Each column of a matrix in the plot indicates the generated word's attention weights associated with the content plan. From the color variation between each cell, where deeper color indicates higher attention value, we can see which positions in the source data input were considered more important when generating the target word.

3. 4 Error Spotting

The error spotting function is designed based on *attention mechanism*. The attention can be broadly interpreted as a vector of importance weights: in order to predict a word in a sentence, we estimate using the attention vector how strongly it is correlated with other elements and take the sum of their values weighted by the attention vector as the approximation of the target [12]. In other words, one generated word with relatively low attention value to all input elements, which are content plan items in our case, tents to have a high possibility of being predicted by vague references.

However, not every word with relatively low attention value will be regarded as errors by our system. We notice that, in sports summaries, factual issue related erroneous texts tend to appear among attribute-related words, entity-related words, and number-related words. Therefore, we mainly focus on these types of word and analyze its maximum attention value, which is the highest attention value towards all content plan items. If a word has the maximum attention value lower than the threshold value we set, ErrorSpotter will regard this word as potential erroneous text.

3. 5 Interactions

Several types of interaction are attached for users to take advantage of the ErrorSpotter on exploring and investigating AI-generated summaries. The 2D heatmap offers the interactions, namely, hover, filter, save, zoom in, zoom out, and reset. The mouse hover function enables users to check each cell's information in detail, including related word and content plan record through the tooltip, as shown in Figure 2. Besides, through changing the attention value from 0 to 1, users can decide to filter out which part of the figure and pay more attention to their interested part. The filtering feature will be of great help for users to find the connection between the attention value and the distribution of the potential erroneous text.

To implement statistics exploration function, one entity select pull-down list is attached to the system. Through selecting a specific entity, relevant game statistics of the selected entity will be displayed visually in tabular format, which can be taken as reference for error checking. Moreover, there is a game selecting panel at the top of the interface where users could explore the data of different games. Once the target game is changed, corresponding content of the text heatmap and the 2D heatmap will also be changed simultaneously.

3. 6 Implementation Details

Visualizations of the ErrorSpotter are created using Bokeh[†1] version 2.1.1, a Python library for interactive visualization that targets web browsers for representation. Running a Bokeh server, we can create an interactive web application that can connect HTML-based front-end UI events to our visual system.

The system is not open to the public yet since we are still trying hard to improve some function-

[†1] Bokeh Documentation: `https://docs.bokeh.org`

alities. When it is ready, we will share the system to public.

4 Summary

Our study provides an interactive and intuitive visualizer for spotting erroneous text for sports summary related NLG tasks. With the attention mechanism integrated, we achieve potential error spotting function. Through combining multiple heatmap visualizations and interactions, the ErrorSpotter offers in-depth explorations on targeted tokens, which help analysts flexibly explore and analyze the errors that occurred in generated text. Below, we discuss some limitations of our proposed visualization, as well as some promising future extensions.

4. 1 Limitations

As we mentioned in subsection 2. 3, in NLG tasks, there are three issue types we mainly focus on. It is efficient to use the ErrorSpotter to detect factual issue related errors. However, it is difficult to tackle the grammatical issues and the artificial issues, only relying on our system. It is desired to combine other efficient techniques with the ErrorSpotter to generate a comprehensive hybrid error spotting system.

Moreover, in order to use the ErrorSpotter, attention-based NLG model is necessary. On that premise, attention-augment visualizer is created to help users gain initial understandings about the correlation between input data and generated text. Then, with the help of attention mechanism, the system can spot potential erroneous text. Therefore, the ErrorSpotter is unable to be used upon non-attention based model generated text.

4. 2 Future Work

The ErrorSpotter is still under development. Some work is planned to be done in the future.

Currently, the ErrorSpotter offers an error spotting function. However, we consider that it will be more convenient and practical to enable users to personally revise the generated text. Therefore, We would like to add editing function into our system where human editors are able to recreate one comprehensive and precise game summary with the support of the ErrorSpotter.

Besides, we are not satisfied with current game data exploration function. Even though users are able to explore game related box-score data through choosing the entity from pull-down list, the exploration function is not interactively linked with the generated text. Users are unable to do exploration in terms of each word appeared in the generated text. Therefore, we plan to make improvement on data exploration part by adding interactive entity choosing function in generated text panel.

Finally, to demonstrate the usefulness of our visualization design, the case study on evaluating the ErrorSpotter is desired. The precision and recall will be tested to verify the accuracy of potential error spotting function. Proper adjustments to our system are also being considered to apply ErrorSpotter in other domains and analysis tasks.

参考文献

[1] Bahdanau, D., Cho, K., and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, 2014.

[2] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014.

[3] Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W.: Unified Language Model Pre-training for Natural Language Understanding and Generation, *Advances in Neural Information Processing Systems 32*, Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R.(eds.), Curran Associates, Inc., 2019, pp. 13063–13075.

[4] Dong, Z., Wu, T., Song, S., and Zhang, M.: Interactive Attention Model Explorer for Natural

Language Processing Tasks with Unbalanced Data Sizes, *2020 IEEE Pacific Visualization Symposium (PacificVis)*, IEEE, June 2020.

[5] Gehrmann, S., Strobelt, H., and Rush, A.: GLTR: Statistical Detection and Visualization of Generated Text, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy, Association for Computational Linguistics, July 2019, pp. 111–116.

[6] Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., and Bengio, Y.: A Structured Self-attentive Sentence Embedding, 2017.

[7] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C.: The CoNLL-2014 Shared Task on Grammatical Error Correction, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, Baltimore, Maryland, Association for Computational Linguistics, June 2014, pp. 1–14.

[8] Omelianchuk, K., Atrasevych, V., Chernodub, A., and Skurzhanskyi, O.: GECToR – Grammatical Error Correction: Tag, Not Rewrite, *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Seattle, WA, USA, Association for Computational Linguistics, July 2020, pp. 163–170.

[9] Puduppully, R., Dong, L., and Lapata, M.: Data-to-Text Generation with Content Selection and Planning, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33(2019), pp. 6908–6915.

[10] Ritschel, H. and André, E.: Shaping a social robot's humor with Natural Language Generation and socially-aware reinforcement learning, *Proceedings of the Workshop on NLG for Human–Robot Interaction*, Tilburg, The Netherlands, Association for Computational Linguistics, November 2018, pp. 12–16.

[11] Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems 27*, Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q.(eds.), Curran Associates, Inc., 2014, pp. 3104–3112.

[12] Weng, L.: Attention? Attention!, 2018. https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html.

[13] Wiseman, S., Shieber, S., and Rush, A.: Challenges in Data-to-Document Generation, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Association for Computational Linguistics, September 2017, pp. 2253–2263.