Visualizing Attention in Sequence-to-Sequence Summarization Models

Halden Lin* Tongshuang Wu† Kanit Wongsuphasawat ‡ Yejin Choi § Jeffrey Heer ¶

Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

Attention mechanisms play a critical role in sequence-to-sequence neural networks, where they define context-sensitive focus of input during output generation (e.g., during text summarization). However, when long input or output sequences (i.e., lengthy texts) are involved, existing visualization techniques fail to support tasks crucial to error analysis. We present an interactive visualization of attention in sequence-to-sequence summarization models. Our design covers core error analysis tasks in spite of long sequences by combining and augmenting existing approaches. We discuss the strengths of this design over existing techniques and demonstrate its application in the workflow of natural language processing researchers.

Index Terms: Attention—natural language processing—sequence-to-sequence models—summarization

1 Introduction

In natural language processing, the success of Recurrent Neural Networks (RNNs) has been accompanied by an increasingly difficult challenge in model interpretability [3]. For tasks such as machine-translation [1], question-answering, and summarization [4], recent work in sequence-to-sequence (seq2seq) models—which consume a sequence of text to generate another—enhance the RNN model through *attention* mechanisms. Attention was designed as a way of encoding probabilistic importance (i.e., *weight*) of input tokens, from which an output token may be generated. It also lends itself intuitively to the human notion of a shifting focus, offering a window into these models' behavior. Existing techniques for visualizing attention, including 2D heatmaps [1], flow maps [5], and text heatmaps [2, 6], have given researchers tools for analyzing models. Such techniques are effective under relatively short sequences (i.e., a sentence or two), but result in loss of legibility and interpretability under longer sequences (as found in tasks such as question-answering and summarization). This negatively impacts four core tasks in error analysis of these models: **(1) Text comprehension**: can we read and easily understand the input and output sequences? **(2) Overview**: which spans in the input sequence are most consequential to the generated output? **(3) Pattern identification**: are we seeing sequential or overlapping attention? Can we isolate interesting or strange model decisions of focus? Is attention limited to exact word matching? **(4) Drill-down**: can we inspect attention on a token-by-token or phrasal basis? The performance of existing techniques in each of these tasks is broken down in Table 1. We elaborate in Sect. 2.

In response to these issues, we design an interactive attention visualization for tasks in which attention is mapped between long sequences. We accomplish this by composing and augmenting two existing techniques: text heatmap and flow map. The former enables overview analysis and text comprehension, while the latter supplements pattern identification. Through linking and interaction, we allow users to perform sophisticated drill-down analysis of spans or tokens of interest. We demonstrate our visualization through an example use case. We model this system around summarization, but it can also be extended to other seq2seq tasks. The examples shown in this paper use data from a pre-trained model by See et al. [6].

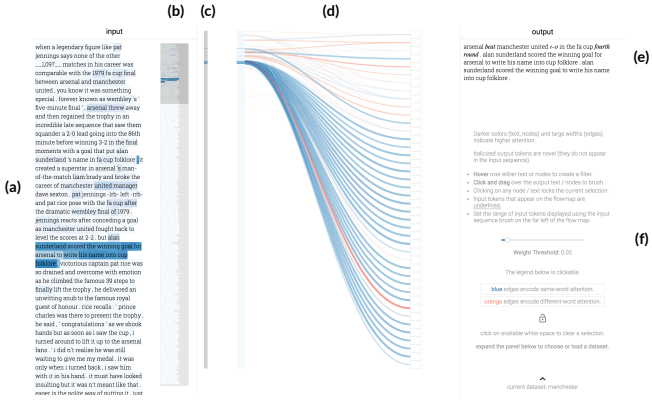

Figure 1: Our visualization system. (a) input text heatmap, (b) input text minimap, (c) flow map input minimap, (d) flow map, (e) output text, and (f) a control panel.

2 Related Work

Attention is typically visualized using one of three methods, as in Table 1. The first two were designed with machine translation in mind [1, 5]: (1) **2D Heatmaps**—correlation matrices with rows being the input, columns being the output, and cells shaded according to the attention weight between their respective tokens [1]; (2) **Flow Maps**—node-link diagrams with edges pairing each input and output token, encoding attention weights via thickness [5]. While widely adopted, these methods are not scalable—increasing the length of sequences quickly squashes text and, with heatmaps, renders cells indistinguishable. Breaking sentences into individual tokens also significantly impacts text comprehension. The third method, **Text Heatmaps**, is often used in longer-sequence domains. In these, one-dimensional heatmaps are superimposed on input text, showing either aggregate [2], or interactive token-at-a-time [6] attention. Both techniques sacrifice detailed information for text comprehension. Aggregation provides an overview of consequential portions of input, while token-at-a-time interaction allows for rudimentary drill-down analysis. However, both fail to enable drill-down analysis over spans (though this can be enabled with modifications, as discussed in Sect. 3) and, crucially, pattern identification (as the structure of the distribution is hidden behind tedious token-by-token interaction). Our work is inspired by text heatmap and flow map techniques.

Table 1: Evaluation of existing visual designs under large sequences.

Task	2D Heatmap	Flow Map	Text Heatmap
Text Comp.	no	no	yes
Overview	no	yes	yes
Pattern Ident.	no	yes	no
Drill-down	no	no	yes, if modif.

*e-mail: haldenl@uw.edu
†e-mail: wtshuang@uw.edu
‡e-mail: kanitw@uw.edu
§e-mail: yejin@uw.edu
¶e-mail: jheer@uw.edu

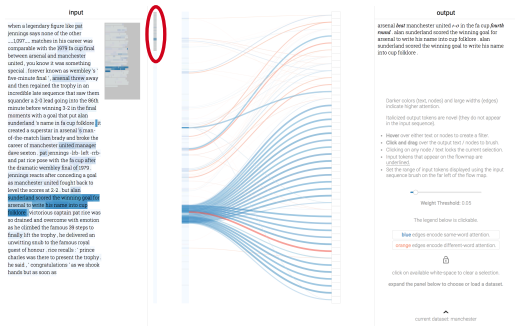

Figure 2: Adjusting the brush over the flow map's input minimap allows users to zoom in on areas of interest. Compare with Fig. 1.

3 Visualization Design

Our design takes the advantage of **flow map** and **text heatmap** techniques. As in Fig. 1, we lay out the input article (a) and the output summary (e) side-by-side in conventional paragraph form (showing aggregate attention at rest), allowing for concurrent comprehension. We also offer a minimap (b) over the scrollable input text such that users may quickly identify regions of high consequence.

To complement the text comprehension and overview analysis enabled by the text component, we employ a flow map (Fig. 1d) to convey high-level attention distribution and structure, enabling pattern identification. Edges encode attention weight between the tokens they connect through width and intensity. We expand on the existing technique [5] in a few ways. First, we color input nodes as a one-dimensional heatmap according to aggregate attention, giving users context and a mapping between text and flow map. Second, output tokens that attend with high weight to matching input tokens represent a copying of the input token. With summarization, this is of particular interest, as it indicates *extraction* (copying of source text), rather than *abstraction* (paraphrasing). To highlight this phenomenon, a blue edge encodes a match in the tokens it connects, while an orange edge encodes a difference.

We link flow map and text components with interaction and filtering to enable drill-down analysis. We expand on existing interactive one-shot highlighting of text heatmaps [6] to enable brushing over multiple tokens. This action results in a normalized aggregation of weights and updating of input heatmaps. We also allow users to zoom in on portions of the input sequence by adjusting a brush over a minimap of the flow map's input heatmap (Fig. 1c).

4 Example Use Case

Suppose we are assessing the quality of the generated summary in Fig. 1. First, by skimming the input pane, we find that the article describes the 1979 FA Cup Final between Arsenal and Manchester United. Next, the heatmap over the text shows us that the generated summary drew primarily from the beginning of the article. We resize the brush over our input sequence (Fig. 1c) to zoom in on this section (Fig. 2). It then becomes apparent that there are two distinct portions of the summary. One portion (the first sentence) seems to be largely abstractive, as hinted at by its orange edges and italicized (novel) tokens. Meanwhile, the rest of the summary looks to be extractive, as hinted at by its blue edges, sequential attention structure, and non-italicized (derivative) tokens. What's more, the overlapping edges in this second half explain the repetition in the generated summary: sentences two and three attend to the same source. This also explains the high aggregated attention in that article section.

Brushing over the first sentence of the summary re-aggregates input heatmaps and gives us a better idea of the input tokens most influential in this sentence's generation (Fig. 3). Another scan of flow map and input pane confirms this sentence as an abstraction—attention is fairly spread out, and the sentence does not appear in the attended-to portions of the input sequence (even in structure).

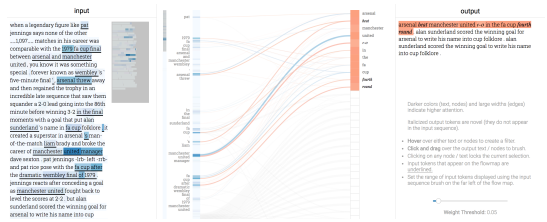

Figure 3: Brushing over output tokens results in normalized aggregation of attention weights over the input.

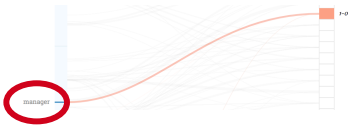

Figure 4: Single token selection can reveal critical errors in the attention mechanism.

While abstraction is desired, a closer look at the article and summary reveals that this summary is factually incorrect—among other technicalities, the match score was 3-2, not 1-0. A selection of the generated token '1-0' (Fig. 4) quickly tells us that attention incorrectly pointed the model towards 'manager' instead of '3-2'. One hypothesis for this inaccuracy could be that the model is overfitting to sports articles, which may potentially describe '1-0' scorelines more often than not.

5 Conclusion

We presented an interactive visualization for attention in seq2seq summarization models and described its strengths over existing visualization techniques[1]. In particular, we pointed out its strong support for text comprehension, pattern identification, and overview & drill-down analysis under long input sequences. We provided an example use case to demonstrate how our implementation can provide initial insights into the behavior of an underlying model. Moving beyond, this visualization may serve as a plug-in for larger model tuning systems, where it can help researchers generalize from specific examples to broader deficiencies. For example, our use case in Sect. 4 can motivate researchers to inspect the distribution of scorelines in the training dataset and subsequently tune the model or introduce novel mechanisms. We also hope to extend this visualization design to other lengthy seq2seq tasks, such as question-answering, machine-translation, and even conversation.

Acknowledgments

We wish to thank Ari Holtzman and Nelson Liu for their guidance in shaping this work. We would also like to thank Dominik Moritz, Aishwarya Nirmal, Alan Tan, and Shobhit Hathi for their feedback.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, 2014.

[2] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

[3] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *CoRR*, 2015.

[4] R. Nallapati, B. Zhou, C. N. dos Santos, aglar Gülehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, 2016.

[5] M. Rikters, M. Fishel, and O. Bojar. Visualizing neural machine translation attention and confidence. 2017.

[6] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL*, 2017.

[1] We have open-sourced this project at `https://bit.ly/2LStwYt`