A Web-based Framework for Collecting and Assessing Highlighted Sentences in a Document

Sasha Spala[1], **Franck Dernoncourt**[2], **Walter Chang**[2], **Carl Dockhorn**[1]
[1]Adobe Systems, [2]Adobe Research
{sspala,dernonco,wachang,cdockhor}@adobe.com

Abstract

Automatically highlighting a text aims at identifying key portions that are the most important to a reader. In this paper, we present a web-based framework[1] designed to efficiently and scalably crowdsource two independent but related tasks: collecting highlight annotations, and comparing the performance of automated highlighting systems. The first task is necessary to understand human preferences and train supervised automated highlighting systems. The second task yields a more accurate and fine-grained evaluation than existing automated performance metrics.

1 Introduction

As people have access to an increasingly larger amount of information, technologies may enable them to consume that information more efficiently. Existing technologies have focused on automated summarization techniques. However, summarization techniques are not fully mature: emphasis mistakes are frequent and may cause the reader to miss crucial points in the summarized document. To address this issue, as an alternative to summarization, key portions of a document can instead be highlighted (or made more visible by bold, italic, etc)Highlights appear within their context (unlike a summary), and the impact of ‘bad" highlights is of much lower consequence than ‘bad" summaries.

We believe highlights to be motivated by reading intentions. Thus, we must determine if a difference exists between extractive summary sentences and human highlights. The framework presented in this paper allows users to efficiently and scalably crowdsource two related tasks: collecting highlight annotations, and comparing the performance of automated highlighting systems.

2 Related Work

Highlighting is one of the most common methods of annotation (Baron, 2009), making it a popular content annotation method for increasing comprehension in many reading domains. Passive highlighting, or highlights that already appear in text, has been shown in several studies to be a useful tool for information retention and comprehension (Fowler and Barker, 1974; Lorch Jr., 1989; Lorch Jr et al., 1995).

Rath (1961) asked human annotators to retrieve the "most representative" sentences in a document and failed to find significant human agreement for both human-retrieved and machine-retrieved sentences; Daumé (2004) showed that when instructed to choose the "most important" sentences from a passage, humans still fell short of significant agreement. Though Daumé (2004) had low expectations for human agreement for summarization, we believe that the effect of *inline* content, such as highlights, could significantly increase the efficacy of this task.

We explored several annotation frameworks, but none of them are designed for collecting and assessing highlights. For example, MAE (Stubbs, 2011) allows annotators to select entire spans of text and assign categories and labels to those spans, but did not allow researchers to normalize user input; one must rely on annotators to select the correct length of input and, in our case, define sentence boundaries. Similarly, BRAT (Stenetorp et al., 2012) makes it difficult to select a sentence with exact boundaries without post-processing annotator input.

[1] `https://github.com/Franck-Dernoncourt/sentence-highlighting`

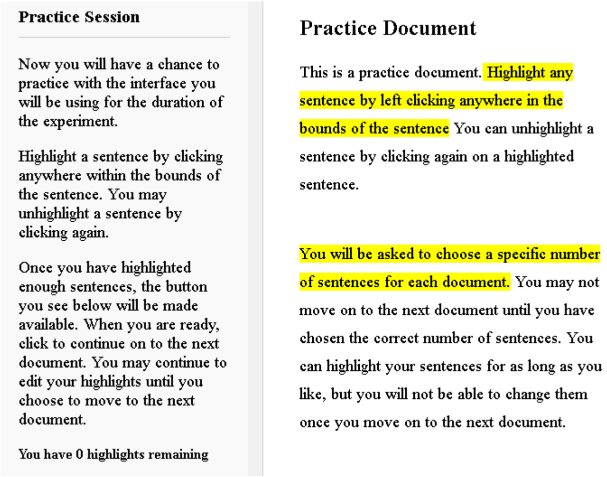
Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 78–81
Santa Fe, New Mexico, USA, August 20-26, 2018.

3 Framework Design

3.1 Overview

We present in the next two sections the interfaces corresponding to the two use cases of our framework: direct highlight annotation, and human evaluation of highlighting systems. For each of these two use cases, the framework collects a wide range of behind-the-scenes data during annotator interactions, including intermediate highlights (versus the final version of highlights an annotator is satisfied with) and the time spent on each section of a document. From the collected data, we can infer a variety of important information , such as how often users adjust their highlights, and whether users scroll across documents to skim the content, or read every word.

Our annotation framework is lightweight, requiring only a Node.js server, which is simple to deploy on Linux, macOS, or Windows.



Practice Session

Now you will have a chance to practice with the interface you will be using for the duration of the experiment.

Highlight a sentence by clicking anywhere within the bounds of the sentence. You may unhighlight a sentence by clicking again.

Once you have highlighted enough sentences, the button you see below will be made available. When you are ready, click to continue on to the next document. You may continue to edit your highlights until you choose to move to the next document.

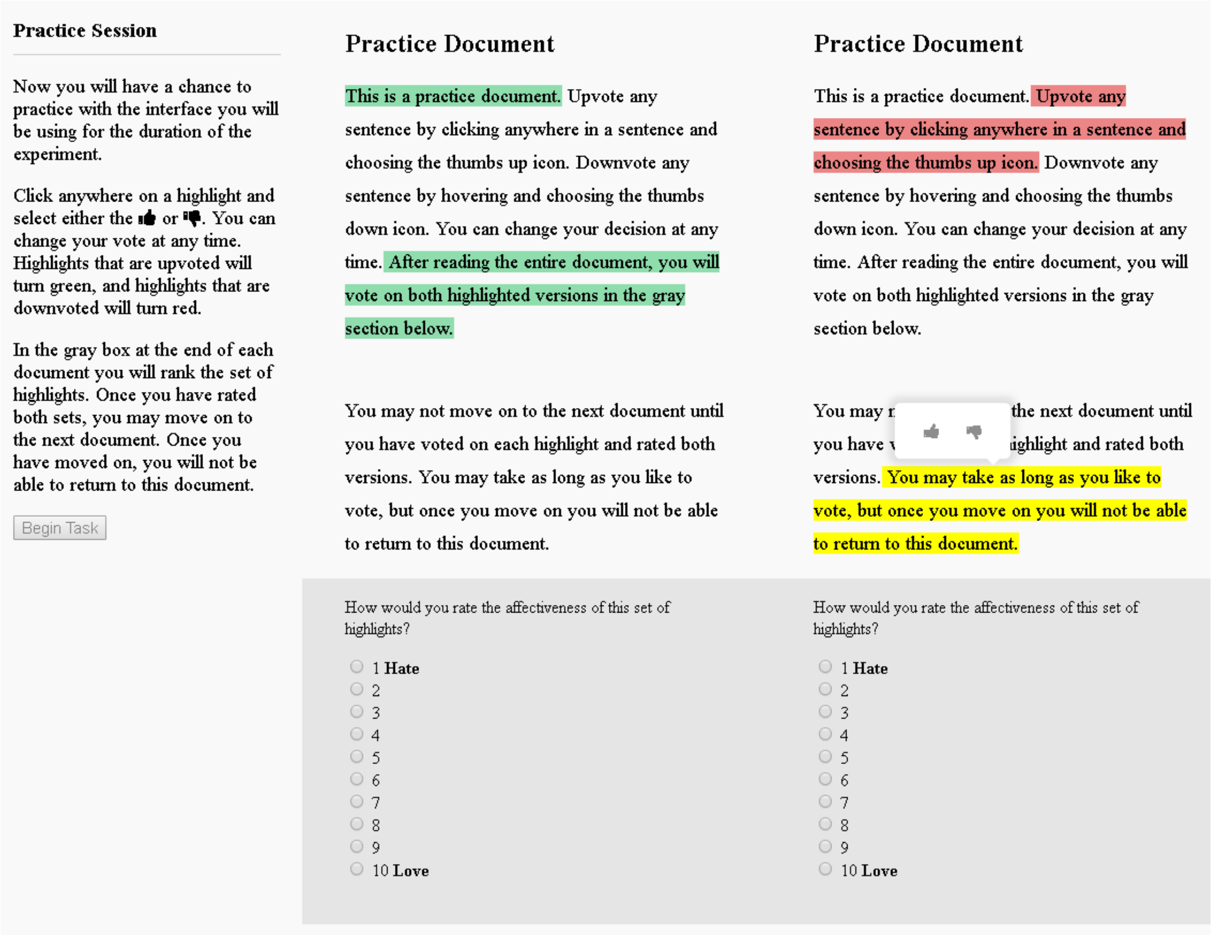
You have 0 highlights remaining

Practice Document

This is a practice document. Highlight any sentence by left clicking anywhere in the bounds of the sentence You can unhighlight a sentence by clicking again on a highlighted sentence.

You will be asked to choose a specific number of sentences for each document. You may not move on to the next document until you have chosen the correct number of sentences. You can highlight your sentences for as long as you like, but you will not be able to change them once you move on to the next document.

Figure 1: Interface for direct highlight annotation. Annotators may highlight or unhighlight any sentence by clicking on it.



Practice Session

Now you will have a chance to practice with the interface you will be using for the duration of the experiment.

Click anywhere on a highlight and select either the 👍 or 👎. You can change your vote at any time. Highlights that are upvoted will turn green, and highlights that are downvoted will turn red.

In the gray box at the end of each document you will rank the set of highlights. Once you have rated both sets, you may move on to the next document. Once you have moved on, you will not be able to return to this document.

Begin Task

Practice Document

This is a practice document. Upvote any sentence by clicking anywhere in a sentence and choosing the thumbs up icon. Downvote any sentence by hovering and choosing the thumbs down icon. You can change your decision at any time. After reading the entire document, you will vote on both highlighted versions in the gray section below.

You may not move on to the next document until you have voted on each highlight and rated both versions. You may take as long as you like to vote, but once you move on you will not be able to return to this document.

How would you rate the affectiveness of this set of highlights?

○ 1 Hate
○ 2
○ 3
○ 4
○ 5
○ 6
○ 7
○ 8
○ 9
○ 10 Love

Practice Document

This is a practice document. Upvote any sentence by clicking anywhere in a sentence and choosing the thumbs up icon. Downvote any sentence by hovering and choosing the thumbs down icon. You can change your decision at any time. After reading the entire document, you will vote on both highlighted versions in the gray section below.

You may r 👍 👎 the next document until you have v ighlight and rated both versions. You may take as long as you like to vote, but once you move on you will not be able to return to this document.

How would you rate the affectiveness of this set of highlights?

○ 1 Hate
○ 2
○ 3
○ 4
○ 5
○ 6
○ 7
○ 8
○ 9
○ 10 Love

Figure 2: Interface for highlighting system assessment. Participants are presented with two versions of the same text with different highlights. Participants must upvote (green) or downvote (red) each highlight, then give a global grade between 1 and 10 for each of the two versions.

3.2 Use Case 1: Highlight Annotation Collection

Figure 1 presents the interface used to collect provided highlight annotations. Annotators are asked to highlight sentences that would make document comprehension easier and faster for another naive reader. A counter in the left column updates the number of highlights remaining as annotators work through each document.

Clicking anywhere within the boundaries of a sentence highlights the entire sentence in yellow. Annotators are allowed to highlight and unhighlight as many times as desired, but are not able to revisit the same document after moving to the next document. All annotators are required to complete a brief tutorial session before beginning that demonstrates the interface controls.

This highlight collection phase attempts to simplify user interaction; Highlighting and unhighlighting can be done with a single left click. There are no color variations; the text size for the left panel and the document title and content stay consistent throughout the task.

3.3 Use Case 2: Highlight System Assessment

Figure 2 presents the interface where participants evaluate highlighting systems. Participants are instructed to "upvote" and "downvote" individual highlights that they believe will help identify the main point(s) of the document. Participants are shown two different highlighted versions, generated from two highlighting systems. Systems are randomized and anonymized, both in location (e.g., left or right side of the content frame) and pairing.

To handle annotation of positive and negative votes on individual highlights, we introduced the "thumbs up" and "thumbs down" buttons, displayed after left clicking anywhere within the boundaries of a highlighted sentence. Participants must vote on every highlight displayed on the document. Once they reach the end of the document, they must rate the two versions of the highlights on a one-to-ten scale before moving to the next document.

4 Analysis Reporting

To help researchers analyze the results, our framework provides analysis scripts, written in Python 3. In this section, we present some of these analyses.

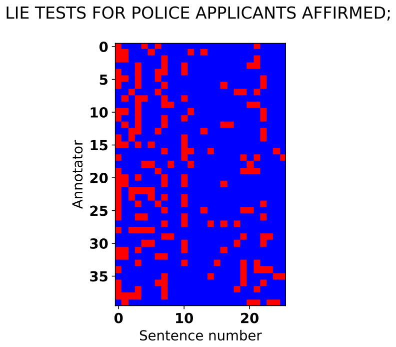


Figure 3: Binary heatmap showing annotator highlights. Red and blue cells correspond to highlighted and non-highlighted sentences, respectively. Each row represents an annotator, each column a sentence in the document.

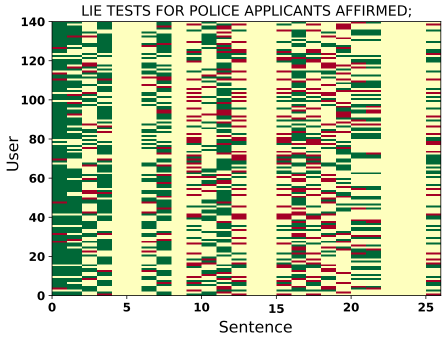


Figure 4: Representation of highlight votes, where green and red cells represent up- and down votes, respectively, and cream reflects that the model shown to the participant did not highlight that sentence.

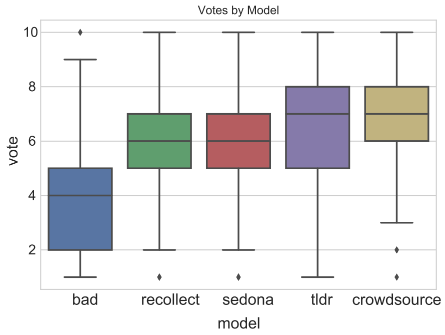


Figure 5: Vote distribution by model. "Bad" is an a model that selects intentionally bad highlights; "crowdsource" displays the highlights most often chosen by annotators during highlight collection; "sedona" (Elhoseiny et al., 2016), "recollect" (Modani et al., 2015; Modani et al., 2016) and "tldr" (`smmry.com`) are extractive summarization models that are used to select which sentences to highlight.

4.1 Use Case 1: Highlight Annotation Results

After annotators highlight sentences in a document, the annotations may be viewed as a binary heatmap, as shown in Figure 3. The heatmap may be used to identify highlight clusters (e.g., if highlights tend to be located at the beginning of the document) as well as an approximate overview of the inter-annotator agreement. The Krippendorff Alpha score (Krippendorff, 2011) is computed, which indicates the overall agreement across all annotators.

4.2 Use Case 2: Highlight Assessment Results

Figure 4 displays up- and down-votes on all sentences in a document, for all automated highlighting models. It can be used to visually determine the consistency of the annotators. E.g., ideally if a sentence is worth being highlighted, it should be upvoted across all annotators, regardless of the model that highlighted it.

Figure 5 contains one boxplot for each model. Specifically, each boxplot represents the distribution of participants' votes that they cast on a document that was highlighted by the model corresponding to the boxplot.

5 Conclusion

In this paper, we have presented a web-based framework designed to efficiently and scalably crowdsource the collection of highlight annotations as well as the human comparison of the performance of automated highlighting systems. The interface is highly customizable, easy to tune, and, based on our experience the framework with Amazon Mechanical Turk, easily understood by annotators. The framework as well as its source code is freely available. We hope it will help foster research in the field of automated highlighting.

References

[Baron2009] Dennis Baron. 2009. *A better pencil: Readers, writers, and the digital revolution*. Oxford Uni. Press.

[Daumé III and Marcu2004] Hal Daumé III and Daniel Marcu. 2004. Generic sentence fusion is an ill-defined summarization task. In *Text Summarization Branches Out Workshop at ACL*.

[Elhoseiny et al.2016] Mohamed Elhoseiny, Scott Cohen, Walter Chang, Brian Price, and Ahmed Elgammal. 2016. Automatic annotation of structured facts in images. In *5th Workshop on Vision and Language*.

[Fowler and Barker1974] Robert L. Fowler and Anne S. Barker. 1974. Effectiveness of highlighting for retention of text material. *Journal of Applied Psychology*, 59(3):358–364.

[Krippendorff2011] Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

[Lorch Jr et al.1995] R.F. Lorch Jr, E. Pugzles Lorch, and M.A. Klusewitz. 1995. Effects of typographical cues on reading and recall of text. *Contemporary Educational Psychology*, 20(1):51–64.

[Lorch Jr.1989] R. F. Lorch Jr. 1989. Text-signaling devices and their effects on reading and memory processes. *Educational Psychology Review*, 1(3):209–234.

[Modani et al.2015] Natwar Modani, Elham Khabiri, Harini Srinivasan, and James Caverlee. 2015. Creating diverse product review summaries: a graph approach. In *ICWISE*, pages 169–184. Springer.

[Modani et al.2016] Natwar Modani, Balaji Vasan Srinivasan, and Harsh Jhamtani. 2016. Generating multiple diverse summaries. *International Conference on Web Information Systems Engineering*.

[Rath et al.1961] G.J. Rath, A. Resnick, and T.R. Savage. 1961. The formation of abstracts by the selection of sentences. part i. sentence selection by men and machines. *JAIST*, 12(2):139–141.

[Stenetorp et al.2012] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *ACL Demonstrations*.

[Stubbs2011] Amber Stubbs. 2011. MAE and MAI: lightweight annotation and adjudication tools. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 129–133. Association for Computational Linguistics.