Auto-Highlighter: Identifying Salient Sentences in Text

Jessica Zeitz Self
Rebecca Zeitz
Alan Breitler
Chris North
Motivation

Time Wars (CS)
Mark Fisher

Time rather than money is the currency in the recent science fiction film In Time. At the age of 25, the citizens in the future world the film depicts are given only a year more to live. To survive any longer, they must earn extra time. The decadent rich have centuries of empty time available to fritter away, while the poor are only days or hours away from death. In Time is, in effect, the first science fiction film about precarity—a condition that describes an existential predicament as much as it refers to a particular way of organising work.

At the most simple level, precarity is one consequence of the "post-Fordist" restructuring of work that began in the late 1970s: the turn away from fixed, permanent jobs to ways of working that are increasingly casualised. Yet even those within relatively stable forms of employment are not immune from precarity. Workers now have to periodically revalidate their status via systems of "continuous professional development"; almost all work, no matter how menial, involves self-surveillance systems in which the worker is required to assess their own performance. Pay is increasingly correlated to output, albeit an output that is no longer easily measurable in material terms.

For most workers, there is no such thing as the long term. As sociologist Richard Sennett put it in his book The Corrosion of Character: The Personal Consequences of Work in the New Capitalism, the post-Fordist worker "lives in a world marked ... by short-term flexibility and flux ... Corporations break up or join, jobs appear and disappear, as events lacking connection." Throughout history, humans have come to terms with the traumatic upheavals caused by war or natural disasters, but "what's peculiar about uncertainty today," Sennett points out, "is that it exists without any looming historical disaster; instead it is woven into the everyday practices of a vigorous capitalism."

Incacity to even conceive of different ways to work, produce and consume. It's now clear that, from (and with good reason) neoliberalism declared war on this alternative mode of time. It remains tiresome propagation of resentment against those few fugitives who can still escape the treadmill of debt and work, promising to ensure that soon, they too will be condemned to performing interminable, meaningless labour—only to have the solution to the current stagnation lay in more work, rather than an escape from the work. If there is to be any kind of future, it will depend on our winning back the uses of time that neoliberalism has sought to close off and make us forget.
Research Questions

1. Are human highlighted sentences representative of human synthesized summaries? When summarizing, how much do humans rely on sentences they previously highlighted?

2. Which sentences in a document do humans deem salient? Why?

3. Are there differences between experts and non-experts when highlighting and summarizing a document?

4. How closely can simple algorithm heuristics mimic the human selection of salient sentences?
The Experiment

• Independent variables
  – document type
  – participants’ area of expertise
• 40 participants
• 2 documents (technical and non-technical)
• Dependent variables
  – highlights
  – summaries
  – reasons for highlights
  – relation between highlights and summaries
Characterization of Summaries (Q1)

Connecting Highlights and Summaries

<table>
<thead>
<tr>
<th>Reference Location</th>
<th>Go To Statement</th>
<th>Time Wars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-technical</td>
<td>Technical</td>
</tr>
<tr>
<td>Highlighted sentences</td>
<td>51%</td>
<td>50%</td>
</tr>
<tr>
<td>Elsewhere in document</td>
<td>10%</td>
<td>8%</td>
</tr>
<tr>
<td>Synthesized information</td>
<td>39%</td>
<td>42%</td>
</tr>
</tbody>
</table>

Highlights are 50% representative of summaries
## User-defined Categories (Q2)

### Prominent Categories of Reasons for Highlights

<table>
<thead>
<tr>
<th>Category</th>
<th>Go To Statement</th>
<th>Time Wars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-technical</td>
<td>Technical</td>
</tr>
<tr>
<td>Argument/main point</td>
<td>13</td>
<td>27</td>
</tr>
<tr>
<td>Supporting evidence</td>
<td>54</td>
<td>45</td>
</tr>
<tr>
<td>Solution</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Profound statement</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Personally resonated</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Conclusion</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>
Rhetorical Structure (Q2)

- Strong correlation between categories and rhetorical elements
- Higher prevalence of formalized elements
  - main point
  - supporting evidence
- Categories fit order of rhetorical elements
Experts versus Non-experts (Q3)

- Minimal variations
- Correlation = 0.82
- Followed same overall trend
- Employed same methods
User and Algorithm Comparison (Q4)

- Metric based on uniqueness performed poorly
- Small correlation with sentence length
- Metric that found sentences representative of whole text performed well

Correlation R Values between Metrics & Humans

<table>
<thead>
<tr>
<th>Metric</th>
<th>Go To Statement</th>
<th>Time Wars</th>
</tr>
</thead>
<tbody>
<tr>
<td># of n-grams</td>
<td>0.542</td>
<td>0.288</td>
</tr>
<tr>
<td>sentence length</td>
<td>0.335</td>
<td>0.046</td>
</tr>
<tr>
<td>tf-idf</td>
<td>0.612</td>
<td>0.398</td>
</tr>
</tbody>
</table>
Comparison Visual

Algorithm

Highlight Visualizer
An Information Salience Tool

Go To Statement Considered Harmful (Algorithm)
Edsger Dijkstra

For a number of years I have been familiar with the observation that the quality of programmers is a definite function of the density of go to statements in the programs they produce. More recently I discovered we use of the go to statement has such disastrous effects, and I became convinced that the go to statement should be abolished from all “higher level” programming languages (i.e., everything except, perhaps, plain machine code). At that time I did not attach too much importance to this discovery: I now submit my considerations for publication because in very recent discussions in which the subject turned up, I have urged to do so.

My first remark is that, although the programmer’s activity ends when he has constructed a correct program, the process taking place under control of his program is the true subject matter of his activity, for it is the process that has to accomplish the desired effect; it is this process that in its dynamic behavior has to be delegated to the machine.

My second remark is that our intellectual powers are rather geared to master static relations and that our powers to visualize processes evolving in time are relatively poorly developed. For that reason we should (as wise programmers aware of our limitations) our utmost to shorten the conceptual gap between the program and the dynamic process, to make the correspondence between the program (spread out in text space) and the process (spread out in time) as trivial as possible.

Humans

Highlight Visualizer
An Information Salience Tool

Go To Statement Considered Harmful (Humans)
Edsger Dijkstra

For a number of years I have been familiar with the observation that the quality of programmers is a definite function of the density of go to statements in the programs they produce. More recently I discovered we use of the go to statement has such disastrous effects, and I became convinced that the go to statement should be abolished from all “higher level” programming languages (i.e., everything except, perhaps, plain machine code). At that time I did not attach too much importance to this discovery: I now submit my considerations for publication because in very recent discussions in which the subject turned up, I have urged to do so.

My first remark is that, although the programmer’s activity ends when he has constructed a correct program, the process taking place under control of his program is the true subject matter of his activity, for it is the process that has to accomplish the desired effect; it is this process that in its dynamic behavior has to be delegated to the machine.

My second remark is that our intellectual powers are rather geared to master static relations and that our powers to visualize processes evolving in time are relatively poorly developed. For that reason we should (as wise programmers aware of our limitations) our utmost to shorten the conceptual gap between the program and the dynamic process, to make the correspondence between the program (spread out in text space) and the process (spread out in time) as trivial as possible.
In Conclusion

**Highlights**

... are 50% representative of human summaries.

... do fit within broad categories.

... do correlate with rhetorical structure.

... are chosen by experts and non-experts using the same methods.

... chosen by an algorithm using simple text metrics are representative of human summaries.
References


