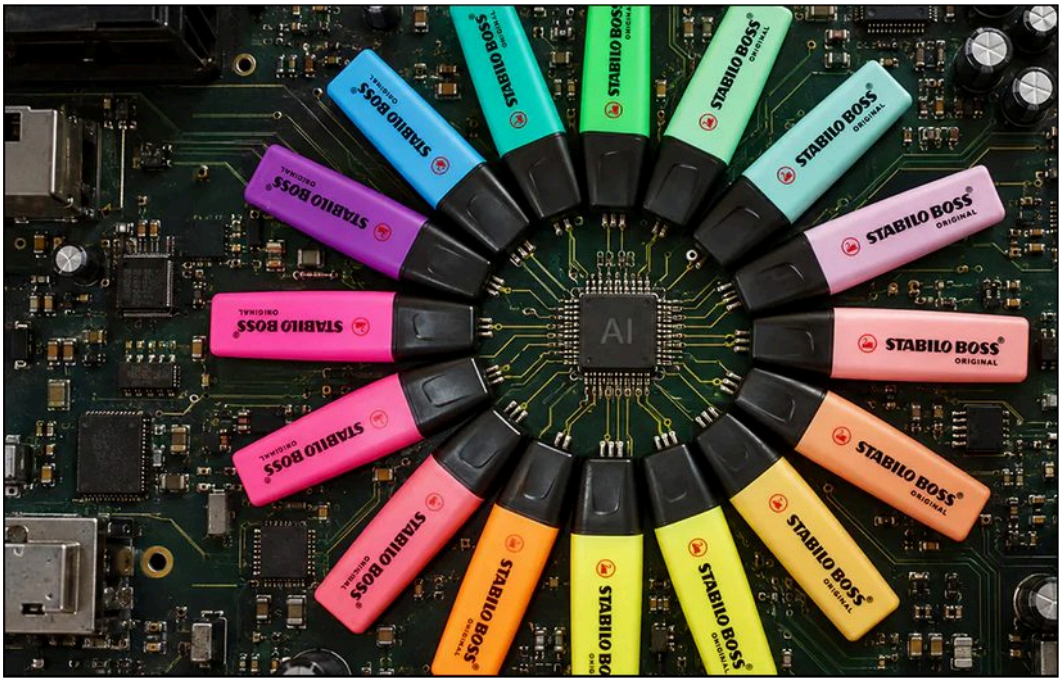
Critically, Enterprise Agents Don't Understand Highlighting

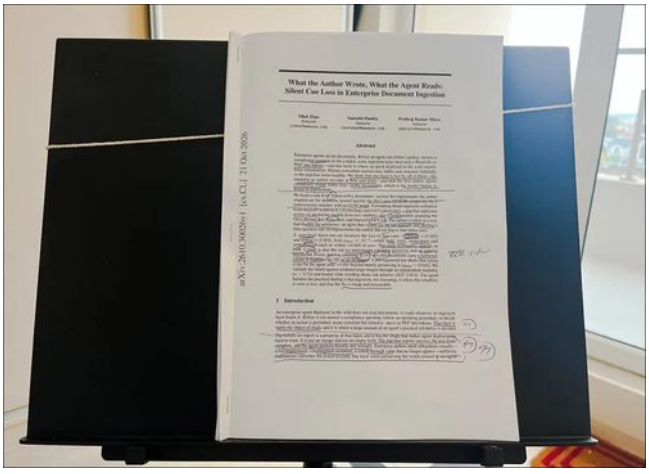
Originally published Fri 30 October at https://www.unite.ai/critically-enterprise-agents-dont-understand-highlighting/



There's a lot of important, unscanned paper in the world, and particularly in the business world. It's estimated that only 1% of the four billion paper documents circulating through global trade each year are digitized, leaving the rest beyond the analytical powers of machine learning.

That's just a rough appraisal of the problem in the *business* sector; governments also produce a huge amount of printed or paper-bound output, and the US National Archive reported this year that only 4.04% of the analog documents it maintains are digitized, with *11.97 billion* yet to go.

Therefore it's an ongoing rather than a legacy problem, and I confess that I am part of the problem too:

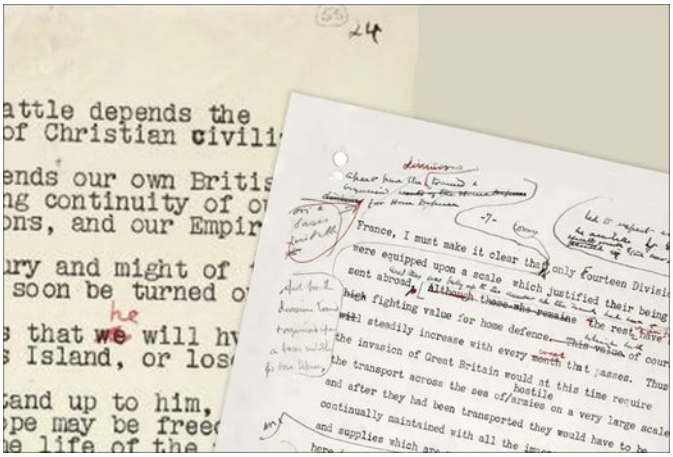


What I sit in front of most days.

In reviewing papers, I print them, because they need a lot of *human* parsing as well as machine summary. Because time is tight, I scrawl large emphases over key sections and potential quotes, so that the shape of the data leaps out at me when it's time to write. Since I'm not quite OCD enough to bother with highlighters, added emphasis is indicated by diverse symbols, larger text, and other very personalized signals of importance.

© Martin Anderson and/or commissioning entities

But a lot of people, including a lot of important people writing important things, do use color highlighters, and alternate text colors, for emphases.



Winston Churchill used different-colored pens for different types of emphases. Here, in the preparations for his 'finest hour' speech, he appears to be using red as a corrective code. Source

According to the very paper depicted in the earlier photo (the one on the black lectern), there may be documents numbering in the hundreds of millions that contain a layer of such emphasis that document-ingestion systems such as Docling are not currently able to see.

These manual emphases are significant, giving weight to text in a way that isn't necessarily reflected in the construction of the language. Therefore reducing the content of such a highlighted page to mere text through OCR (or more advanced methods, such as VLMs) does not convey all of the author's intent, and may even misrepresent the purposes and intentions of the document.

Just the Highlights

The new paper* finds that widely-used document converters routinely discard highlighting and text colors, so that AI agents remain unable to discern important instructions. Of the 6,462 policy documents examined in the study's tests, *nearly a third* contained red text or highlighting, and around one in ten used these markings to draw attention to important requirements.

In regard to enterprise document ingestion agents, the authors† state††:

> *'Mature converters recover text, tables and structure faithfully, so the pipeline looks healthy.*
>
> *'We show that one layer is lost by all of them—the meaning an author encodes in how text* looks*—and that the loss makes agents* confidently wrong *rather than visibly incomplete, which is the harder failure to detect in deployment.'*

In the authors' test, when AI models were asked to identify important requirements in real policy documents, they performed far better when the original highlighting and text colors were preserved – a finding confirmed across six models, including GPT-5.6 Terra, Claude Opus 5, and Claude Sonnet 5.

Effectively, it's possible to extract two signals from highlighting: a) the fact that anything was highlighted at all and b) the *color* it was highlighted in. In an instance where multiple colors were used for manual highlighting, a logical rubric would code red *as in some way negative* (as we saw earlier, in Winston Churchill's corrections to his own speeches):

```
[HEADING level=1 style="Heading 1"]
1. Introduction
[/HEADING]

[PARAGRAPH style="Normal"]
A **probationary period** is the first 90 days of
employment during which the employee's performance
and fit are evaluated. During this period, either
party may terminate employment with 5 days' written
notice.
[/PARAGRAPH]

[PARAGRAPH style="Normal" roles="CRITICAL_WARNING"]
[COLOR:#FF0000]**WARNING: Employees who fail to
submit the I-9 form within 3 business days will be
immediately suspended without pay.**[/COLOR]
[/PARAGRAPH]
```

At the bottom of this JSON output for the authors' first-stage parser, we see that the pipeline has found and codified red highlighting, and, logically, we can see that the words match that codified level of intensity. However, the method is more useful when the emphasis is added to words that do not in themselves escalate the tone that much, and where the highlighting therefore adds a missing emphasis. Source

© Martin Anderson and/or commissioning entities

The authors suggest that existing pipelines can solve the problem quite simply, by preserving the original highlighting and text colors during document conversion, and adding labels that explain their significance to the AI model. While this requires no additional AI processing, it does roughly double the number of tokens needed to represent each document:

> *'Recovering these cues is deterministic, code-only, and costs no LLM calls; the entire effect reported here comes from parsing alone. It does cost tokens—roughly twice plain text—so it is a deliberate trade of context budget for factuality rather than a free win.'*

Bold Indifference

One interesting facet of the study is the extent to which the existing means of emphasis in typed digital documents is, apparently, ineffective through overuse[††]:

> *'The entire effect [i.e., of the new pipeline] is carried by* ***highlight*** *[and] [****colour****]. Bold, italic, monospace and strikethrough contribute nothing [measurable], with confidence intervals straddling it.*
>
> *'This is not because those cues are absent: a single document carries over a thousand bold markers. It is because they are too common to be informative.*
>
> *'An author who bolds every heading and defined term has not thereby singled anything out, whereas colouring or highlighting a requirement is a deliberate, sparing act. Cue typing pays only for cues rare enough to signal intent.'*

Tasks and Tests

To establish the system, the authors scanned 6,462 real policy documents from an unspecified public corpus in .docx format, checking the documents for red text and highlighting, without making any guesses based on the wording. The models were then asked to pick out the requirements that had been marked as especially important, and leave the unmarked ones alone, with the core prompt:

> *'List every requirement in this document that the author has singled out for special attention or elevated priority.'*

For the main test, 84 documents were selected, containing 332 marked requirements and 4,192 unmarked ones (i.e., requirements without red text or colored highlighting). Each document had to contain at least one marked requirement and three unmarked ones, so the AI couldn't simply list everything.

The ground truth was established directly from the documents' formatting, rather than through human judgment or another AI model. Performance was measured using precision, recall and F1 scores, taking account of both correctly identified requirements and false positives, where unmarked requirements were mistakenly included.

Scoring was programmatic, with no LLM judge involved. To count as a match, an answer had to contain at least four consecutive content words from the original requirement, helping to prevent similar but different requirements from being counted as correct.

Ablating Highlights

A further test was carried out to see whether important requirements could be identified from their wording alone, without highlighting or colors. A text-only classifier trained on examples achieved an ROC AUC score of 0.664 (where 0.5 represents random guessing).

Simpler methods performed no better than chance: a score of 0.508 was recorded when words suggesting urgency were used as clues, while sentence length produced a score of 0.492. These results suggest that very little can be learned about what the highlighter-wielding author *intended*, just from the wording alone.

This makes sense: presumably, if the language had been adequately forceful, the author would not have felt the need for any *post facto* emphasis.

Contenders

The approaches compared comprised the authors' own parser, *OURS*; *OURS with salience roles*; *OURS with untyped markup*; *docx2python*; Docling; and plain-text conversion.

© Martin Anderson and/or commissioning entities

The same documents and questions were used for each approach, with only the *way* the documents were converted being changed. The principle comparison covered 76 documents, with each test repeated three times:

Arm	micro F1	prec.	FP rate	Δ macro F1	p_{Holm}
OURS	**0.690**	0.824	0.011	—	—
OURS, salience roles	0.687	0.834	0.010	−0.005	1.00
OURS, untyped markup	0.605	0.811	0.010	+0.092	**0.025**
docx2python	0.586	0.797	0.010	+0.110	**0.022**
docling	0.150	0.196	0.043	+0.578	$< 10^{-4}$
plain text	0.075	0.437	0.005	+0.689	$< 10^{-4}$
Pre-registered confirmatory test (79 docs, 6 repeats)					
OURS, salience roles	**0.667**	0.836	0.010	—	—
OURS, untyped markup	0.597	0.818	0.009	+0.085	**0.049**
docx2python	0.570	0.799	0.010	+0.107	**0.014**
H1 typed > untyped: 13/4/62, CI [+.035, +.145]					
H2 typed > docx2python: 17/4/58, CI [+.047, +.174]					

Test results comparing six methods of identifying requirements marked as 'important' in Word documents. The upper section shows results from 76 documents, tested three times. The lower section shows a separate confirmation test using 79 documents and six repetitions. Higher F1 scores indicate better performance; lower false-positive rates indicate fewer unmarked requirements incorrectly selected. Statistically significant differences are shown in bold.

As shown above, the best overall F1 score of 0.690 was achieved by the authors' system, compared with 0.586 for docx2python; 0.150 for Docling; and just 0.075 for plain text (F1 combines precision and recall into a single measure of accuracy, where 1.0 represents a perfect score).

The importance of preserving formatting was also demonstrated by the false-positive results: with Docling, unmarked requirements were incorrectly selected at nearly four times the rate recorded for the authors' system (4.3% versus 1.1%).

A separate test was used to examine whether AI models benefited from being told what the formatting meant, rather than simply being shown the original colors and highlighting, and a small but statistically significant improvement was found when these extra labels were provided. However, most of the benefit had already been obtained by preserving the formatting itself.

External Pressure

The researchers conducted an additional assessment using Microsoft's Office Comprehension Benchmark (OCB), which measures how accurately information can be recovered from Office documents.

326 questions were extracted from 59 Word documents, covering formatting, tables, hyperlinks, comments and footnotes. Both the questions and the scoring criteria were supplied by Microsoft, independently of the study:

Feature	n	plain	docling	unstr.	dx2py	ours
Table	143	0.916	0.939	0.932	**0.986**	**0.986**
Text	59	0.977	0.977	0.977	0.977	**1.000**
Font	24	0.083	0.312	0.292	0.604	**0.750**
Hyperlink	16	0.062	**1.000**	0.062	**1.000**	**1.000**
Layout	14	0.357	0.196	0.071	0.714	**0.821**
Metadata	14	0.000	0.000	0.000	0.000	**1.000**
Comment	11	0.091	0.091	0.091	0.091	**1.000**
Highlight	9	0.333	0.333	0.333	**1.000**	0.889
Footnote	8	0.000	0.000	0.000	0.000	**1.000**
other (5 features)	28	0.214	0.446	0.500	0.536	**0.929**
formatting	38	0.184	0.408	0.263	0.750	**0.816**
non-formatting	288	0.693	0.761	0.722	0.818	**0.977**
all items	326	0.634	0.720	0.668	0.810	**0.959**
Δ vs. ours		+.325	+.239	+.290	+.148	—
p_{Holm}			all $< 10^{-6}$			—

Results comparing five document-conversion methods on Microsoft's Office Comprehension Benchmark, covering 326 questions across 59 Word documents. Scores range from 0 to 1, with higher scores indicating better performance, with best score in each category in bold. Bottom rows summarize overall performance and statistical significance. Here, the authors' system achieved the highest overall score, though docx2python performed better on highlighting.

As shown above, an excellent overall score of 0.959 was achieved by the authors' system, compared with a good score of 0.810 for docx2python, a moderate 0.720 for Docling, and a relatively poor 0.634 for plain text.

© Martin Anderson and/or commissioning entities

Perfect scores were achieved by the new system for metadata, comments and footnotes, while all competing methods scored zero – though these improvements weren't solely due to preserving highlighting and text colors; and the authors' system also proved better at recovering information from tables, document layouts, fonts and hyperlinks.

Conclusion

I tried with little success recently to see if my own manual emphasis of the tables and figures in a PDF science paper could be automated via LLMs and/or Python, and the results were…well, hilarious. Stratifying, quantifying and parsing science papers is now a well-subscribed strand in the literature, and there's not much room – or hope – for the casual amateur.

As regards to this paper, the authors' victory is to have identified a significant emphasis indicator that was being lost in translation. However, the doubling of token spend required by their solution is no joke, at the kind of scale that OCR and document ingestion layers need to operate at. Therefore, the prize here is to have *identified* the issue, I think.

* *Titled 'What the Author Wrote, What the Agent Reads: Silent Cue Loss in Enterprise Document Ingestion'.*

† *All from Amazon.*

†† *Authors' emphases, not mine, though I do routinely convert authors' inline citations to hyperlinks.*

First published Friday, October 30, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Website: martinanderson.ai
Contact: martin@martinanderson.ai