Knowledge-Centric Templatic Views of Documents

Anonymous ACL submission

Abstract

Authors seeking to communicate with broader audiences often share their ideas in various document formats, such as slide decks, newsletters, reports, and posters. Prior work on document generation has generally tackled the creation of each separate format to be a different task, leading to fragmented learning processes, redundancy in models and methods, and disjointed evaluation. We consider each of these documents as *templatic views* of the same underlying knowledge/content, and we aim to unify the generation and evaluation of these templatic views. We begin by showing that current LLMs are capable of generating various document formats with little to no supervision. Further, a simple augmentation involving a structured intermediate representation can improve performance, especially for smaller models. We then introduce a novel unified evaluation framework that can be adapted to measuring the quality of document generators for heterogeneous downstream applications. This evaluation is adaptable to a range of user defined criteria and application scenarios, obviating the need for task specific evaluation metrics. Finally, we conduct a human evaluation, which shows that people prefer 82% of the documents generated with our method, while correlating more highly with our unified evaluation framework than prior metrics in the literature.

1 Introduction

Sharing information is vital for communication and discourse across domains, as it allows for knowledge to be disseminated to a wider audience. This is often done by users through documents in multiple formats that nevertheless share some underlying knowledge. A product manager may need to create a requirements spec, a product pitch deck, and an announcement newsletter for the same project. Likewise, a person on the job market may create a resume, a cover letter, and a personal website. We consider these documents to be *templatic views* of the same underlying knowledge.

This is equally true for the scientific domain, in which researchers create documents in multiple formats to effectively communicate and showcase their work, – such as through academic papers, conference talks, social media posts, poster presentations, and non-technical blog posts. Sharing knowledge in multiple formats broadens the audience and can help bridge the information gap between domain experts, researchers in adjacent fields, and even the general public, leading to greater understanding, collaborations and accelerated progress (Bornmann and Mutz, 2014).

Past work on document generation has focused on developing generation and evaluation methods specific to a single document type (Fu et al., 2021; Qiang et al., 2016; Chandrasekaran et al., 2020). Narrow, custom methods tailored to individual document types are, nevertheless, time consuming to engineer and manage over the long term. For example, in an enterprise setting, it's common to have dozens of occupation- and task-specific documents, each with their own template.Additionally, specific trained methods require data that may be expensive to acquire, or even be unavailable entirely. Meanwhile, LLMs have recently shown great success in long document generation (Radford et al., 2019; Brown et al., 2020), indicating that this fragmentation of methods may no longer be necessary. Thus, our goal is to unify methods for both generating and evaluating templatic views of documents, allowing system designers and engineers to manage and adapt to a range of document types and domains easily and efficiently.

We begin by showing that LLMs are capable of diverse, structured document generation, requiring very little instructional guidance to do so effectively. Additionally, a few minor augmentations to the prompt – such as a structured, intermediate representation, and simple stylistic descriptions –

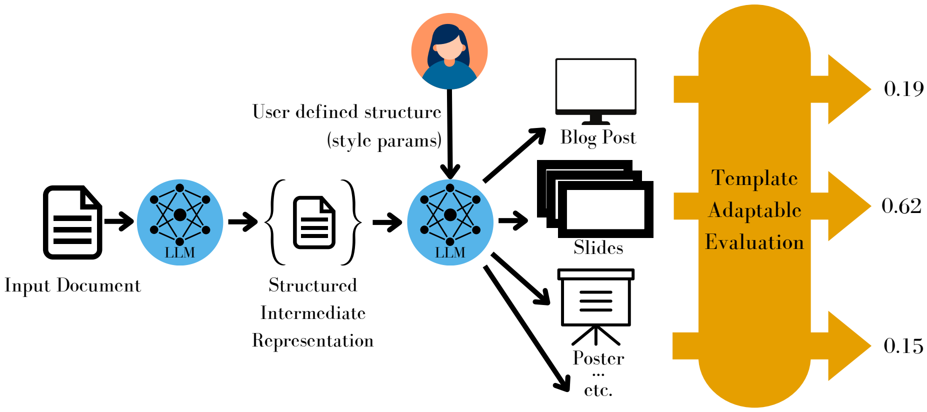


Figure 1: Visualization of our method to unify the generation and evaluation of templatic views of documents. Given an input document, we prompt the LLM to generate an intermediate representation. We can use the representation to prompt the model to generate a templatic view of the input document. We then evaluate the generations using our unified evaluation framework. The LLM represented in the figure is the same model.

can further improve downstream performance, especially for smaller, less resource intensive models. These findings have important implications on the deployment and scaling of unified, real-world AI-assisted document authoring systems.

In similar vein, we then introduce Template Adaptable Evaluation (TAE), departing from prior work's task specific evaluation methods (Zhang* et al., 2020; Qiang et al., 2016; Wang et al., 2015). TAE is a unified precision-recall style framework for automatic evaluation that is highly customizable, allowing users to easily integrate existing text-based metrics from the literature into its formulation and tailor it to their specific use case.Additionally, this framework allows developers to compare performance across document types, without needing to develop an evaluation metric for each individual template.

We evaluate our unified approach for templatic view generation and evaluation on 3 types of documents: slides, posters, and blog posts (Fu et al., 2021; Qiang et al., 2016; Chandrasekaran et al., 2020). Our experiments demonstrate that using a structured intermediate representation leads to improvements in performance across tasks, with greater gains for smaller language language models. In our human evaluation to validate both our unified document generation method and evaluation metric, we show that annotators prefer the output yielded by the structure-aware generation process 82% of the time and that our evaluation metric correlates more highly with human preference than other popular metrics. We release our code[1] to support future research.

[1]`Link suppressed for review.`

2 Related Work

There are several areas of related research in NLP that are relevant to the problems of document transformation and evaluation.

Document summarization has been explored in a number of domains, including news (See et al., 2017), literature (Sciré et al., 2023), law (Deroy et al., 2023), and dialogue (Chen et al., 2021). In the scientific domain, summarization of scientific papers has taken the form of long form summaries (Chandrasekaran et al., 2020), abstract generation (Cohan and Goharian, 2015), conference talks (Lev et al., 2019), and query based summaries (Fok et al., 2023). These summaries can be either extractive (Sefid and Giles, 2022) or abstractive (Chandrasekaran et al., 2020).

Although the tasks of slide and poster generation have generally been considered separate from scientific summarization, they are related in that both tasks require taking an input article, then organizing and abstracting the information to generate a new document. Past work has developed methods for slide generation from papers (Hu and Wan, 2015; Li et al., 2021; Hu and Wan, 2015; Fu et al., 2021), from code (Wang et al., 2023a), or based on a query (Sun et al., 2021). Poster generation has been explored in the form of content extraction for posters (Xu and Wan, 2021), interactive generation (Wang et al., 2015), or full content generation using graphical models (Qiang et al., 2016). To the best of our knowledge, our work is the first to create a unified method capable of generating a diverse range of templatic views of a source document.

Large Language Models (LLMs), which are central to our approach, have shown impressive capabilities in a variety of tasks (Radford et al., 2019; Brown et al., 2020). Based on the transformer architecture (Vaswani et al., 2017), LLMs have shown emergent abilities in tasks such as arithmetic and question answering (Wei et al., 2022a). Similar to chain of thought prompting (Wei et al., 2022b) and content planning prompting (Wang et al., 2023b), we show that by generating an intermediate representation of an input document can improve performance over simply prompting the model to generate the final document from the original input.

As past work has tackled generation of templatic views as separate tasks, methods for automatic evaluation of different document types is fragmented. LongSumm, the shared task introduced by Chandrasekaran et al. (2020), uses ROUGE to evaluate model performance (Lin, 2004). Fu et al. (2021) introduced Slide Level ROUGE to evaluate slide generation, a variant that contains a penalty for the number of slides. Qiang et al. (2016) used a trained regressor. For summarization, many automatic evaluation metrics have been introduced such as BERTScore (Zhang* et al., 2020), UniEval (Zhong et al., 2022), BARTScore (Yuan et al., 2021), BLANC (Vasilyev et al., 2020), and MoverScore (Zhao et al., 2019). However, these metrics are intended for a simple input document-summary setup, and do not take into account factors that affect the quality of other types of documents (e.g. structure). Our work is the first to introduce template adaptable evaluation, allowing uniform comparison of performance across template types.

3 Data

We begin by describing the data used in this paper. There is no existing dataset that includes multiple views of a single document. Instead, we evaluate our unified method, described in §4, on 3 existing datasets: DOC2PPT, LongSumm, and Paper-Poster (Fu et al., 2021; Chandrasekaran et al., 2020; Qiang et al., 2016). These datasets are chosen because they each involve generating a different view of a document. Although our method is not specific to the scientific domain, it is one of the few domains with abundantly available public data of multiple templatic views [2]. The three datasets and their associated generation tasks are described below.

Slide Generation. We use use the DOC2PPT dataset (Fu et al., 2021), which contains 5.8K scientific papers in Computer Science and their respective slide decks. As Fu et al. (2021) do not release data splits or code, we randomly sample 1K examples from this dataset for evaluation. The slides are provided as an image for each slide. We use the Azure OCR tool to extract the text from each slide[3].

Blog Generation. We use the LongSumm dataset (Chandrasekaran et al., 2020), which includes blog posts of scientific papers in the Computer Science domain. Since our approach requires no training or supervision, we use the entire training split from Longsumm as our evaluation set. Of the 531 publicly released blog posts in this set, we could only access 505, with the other 26 including broken links or being behind a paywall.

Notably, while Longsumm includes a blind test set of 22 papers, this test set only consists of inputs without their reference outputs, thus making it impossible to compute our custom evaluation metric (see §5). In the interest of completeness and comparison to prior work, we do, however submit runs from our systems to the leader board and report the results of this blind test set in Appendix D.

Poster Generation. We use the Paper-Poster dataset (Qiang et al., 2016), which consists of a dataset of 85 papers in Computer Science and Biology, and their respective scientific posters; two examples containing corrupted PDFs are excluded. Although Qiang et al. (2016) release data splits, they do not release code or results for comparison. Given the small size of the dataset, we use it in its entirety for more robust results. While the authors uses the source files to extract the text of posters for evaluation, they only release the PDF formats. To preprocess the reference posters, we found that automatic tools to extract text from documents did not handle the visual layout of posters well, so we manually extracted the text of the posters in this dataset. Note that this process was only done to obtain evaluation scores, and that our unsupervised generation method is capable of creating target documents without the need for reference data.

For all 3 datasets, we use the Azure Document Layout tool to extract the text of the input papers.[4]

[2]We acknowledge that scientific writing does have structural regularities that may influence unified document generation. Due to the lack of other available datasets we leave exploration of other domains to future work.

[3]https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-ocr

[4]https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/concept-layout

4 Unified LLM-powered Generation of Templatic Views

The most straightforward way to transform documents between templatic views using LLMs, is to simply prompt the system to generate the target view given the input. However, similar to chain of thought prompting (Wei et al., 2022b), we hypothesize that first generating a structured, intermediate representation of an input document and then reasoning over that representation will result in better generations than directly prompting the model. Our goal is to evaluate the capabilites of LLMs to generate long, structured documents, and experiment with how structured prompting can improve performance. We experiment with a simple general two-step process: first generate an intermediate representation, then generate the templatic view. These steps are described in greater detail below, and the process is visualized in Figure 1.

Intermediate Representation Generation. In this work, we set the intermediate representation to be a JSON consisting of a structured layout of the most important parts of the input. We provide the input document to the model along with a template of the representation and prompt it to extract the most important information from the input document, and format it in the given JSON structure. The exact prompts and JSON structure can be found in Appendix §A. While our experiments use a JSON intermediate representation, note that other formats that provide structure to the input text could be employed (e.g. XML or Markdown). Rather than trying to optimize for the best representation format, our goal is to show that this chain of extraction approach along with structured augmentation to prompts can aid the quality of generations from LLMS. We leave exploration of different formats and other prompt optimization to future work.

Templatic View Generation. We then feed the generated representation as input back into the LLM, prompting the model to generate the final output document, represented as a LaTeX document. For each templatic view, the prompt to generate the final LaTeX document takes a short description of the desired output, which we refer to as a style parameter. For example, the style parameter for slide generation is as follows: “Slides should include a title page. Following slides should contain an informative slide title and short, concise bullet points. Longer slides should be broken up into multiple slides.” The use of style parameters makes our method adaptable to new templatic views; the user only needs to write a short description of the template style. Both the generation of the intermediate representations and the final documents require little to no prompt engineering. The prompts and style parameters can be found in Appendix §A.

5 Template Adaptable Evaluation

Prior work on document generation has treated the evaluation of different templatic views as separate tasks. Thus, our goal is to develop a framework of automatic evaluation that is *template adaptable*. This not only allows us to compare performance across diverse datasets, it also removes the requirement of designing and maintaining individual metrics for each template. In order to generalize to multiple templates, we introduce the concept of *panels*. A panel is a unit of organization within a document type, for which the placement and ordering of the panel is important to the overall flow of information in the document.

For example, we consider panels to be each slide in a slide deck and each section on a poster. We consider the entirety of a blog post to be a single panel. Although we test our method on the tasks of slide, blog, and poster generation, the concept of panels is not limited to these document types. For example, each post on a social media thread could be considered a panel, or each page on a website.

We aim to unify the evaluation of templatic views by integrating prior metrics into a template adaptable precision-recall framework, which we refer to as Template-Adaptable Evaluation (TAE). TAE is not a new individual metric, but rather an evaluation framework that allows generalization to new templates. For example, TAE can even be used *with* ROUGE to evaluate poster generation.

The general TAE formulation is as follows:

$$\begin{aligned} \text{Precision} &= Q_P \times O_P \times L \\ \text{Recall} &= Q_R \times O_R \times L \end{aligned} \tag{1}$$

in which Q_P is the precision measure of quality (§5.1), O_P is the precision penalty for order (§5.2), and L is the non-reflexive penalty for length (§5.3). Similarly, Q_R is the recall quality measure and O_R is the recall penalty for order. The precision-recall formulation allows evaluators to decide which measure is most important to them, or calculate an overall F-measure score.

5.1 Quality Measure

For the TAE precision score, we calculate the average similarity between the generated panels and their most similar reference panel as follows:

$$Q_P = \frac{1}{|\tilde{S}|} \sum_{\tilde{S}} \max_{\text{sim}}(S, \tilde{S}_i) \tag{2}$$

in which S is the set of reference panels and $\tilde{S}$ is the set of generated panels. For the similarity metric, the user can choose a metric that best matches their use case, such as ROUGE, BERTScore, or a custom trained regressor (Lin, 2004; Zhang* et al., 2020). For example, a user might choose ROUGE if they want a similarity metric that focuses on exact word overlap, or BERTScore to measure broader semantic similarity.

Similar to precision, the TAE recall score is calculated as the average similarity between the reference panels and their most similar generated panel:

$$Q_R = \frac{1}{|S|} \sum_{S} \max_{\text{sim}}(\tilde{S}, S_i) \tag{3}$$

By splitting the evaluation of quality into precision and recall, we can evaluate both the content of the slides that were generated as well as the coverage of this content against some reference.

5.2 Order Penalty

Broadly, the goal of the ordering penalty is to measure the similarity of the *order* of information in reference and generated panels, independent of other factors. Unfortunately, because the cardinality of panels in the two outputs is not necessarily the same, a direct one-to-one mapping to compare ordering is not feasible. Instead, a panel in one set can align to multiple references in the other, or none at all – as depicted in Figure 2. Intuitively, our solution is to virtually replicate (resp. drop) panels that have multiple (resp. zero) alignments in the reference set so that a one-to-one mapping of ordering, can in fact be computed.

Formally, assume S and $\tilde{S}$ are sequences of reference and generated panels respectively. We use the maximum similarity scores calculated in §5.1 to align the panels across sets.

For the precision ordering penalty, we define the following operation $\lambda_P(s) = \sum_{\tilde{s}} \delta_P(s, \tilde{s})$, where

$$\delta_P(s, \tilde{s}) = \begin{cases} 1, & \text{iff } s \to \tilde{s} \\ 0, & \text{otherwise} \end{cases}$$

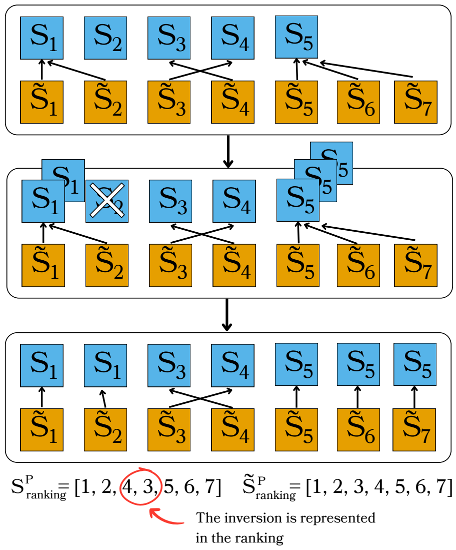


Figure 2: Example of the process of obtaining the rankings for the precision ordering penalty. We first use the similarity measure to map each generated panel to its most similar reference document. This mapping is used to calculate the precision quality score Q_P. We then use the mappings to create a one-to-one alignment from the generated to the reference panels, which we use to calculate the precision ordering penalty (O_P). By creating a one-to-one alignment, we are able to represent inversions in the ordering. This process is reflexive, and panels not accounted for in the precision ordering penalty are accounted for in the recall ordering penalty.

Intuitively, this captures the cardinality of the alignment of a panel in S with panels in $\tilde{S}$. Then, using this operation we can replace every $s \in S$ with $\lambda_P(s)$ copies, leading to an identical cardinality for both S and $\tilde{S}$, and subsequent one-to-one mapping between their corresponding panels.

Then, to operationalize a penalty score for the two sets of ordered panels we associate them with ranks in both sets and use a rank correlation metric to compute the degree of agreement. Specifically, rank assignment is done as follows: panels in $\tilde{S}$ are simply assigned ranks in order of appearance 1 through N – we call this $\tilde{S}^P_{ranking}$; meanwhile panels in S are assigned the identical rank to their one-to-one aligned panel in $\tilde{S}$ and $\tilde{S}^P_{ranking}$ – we refer to these rankings as $S^P_{ranking}$. An example of this process can be found in Figure 2. The final ordering penalty is computed using Spearman's rank correlation (Szmidt and Kacprzyk, 2010):

$$O_P = \frac{\text{Spearman}(S^P_{ranking}, \tilde{S}^P_{ranking}) + 1}{2} \tag{4}$$

where we perform a linear transformation to map the original range of the correlation coefficient [-1, 1] to the desired range [0, 1].

Similarly, for the recall ordering penalty, we map the reference panels to the generated panels, calculated as $\lambda_R(s) = \sum_s \delta_R(\tilde{s}, s)$. O_R is calculated similar to O_P, using the recall rankings.

5.3 Length Penalty

Finally, we compute a length penalty for both the recall and precision scores. Similar to Fu et al. (2021), this is done as follows:

$$L = e^{\frac{-\mathrm{abs}(|S|-|\tilde{S}|)}{|S|}} \tag{5}$$

We chose to keep L non-reflexive, because in the reverse case – as $|\tilde{S}| \to \infty$, $L \to 1$ – the metric could be cheated by over-generating.

6 Results

As mentioned in §3, past work on Doc2PPT and Paper-Posters do not release code, making it difficult to do a direct comparison. They also do not report any baselines to compare against. Meanwhile, Longsumm's blind test does not allow us to compute our custom metric, although we do report the leaderboard results in Appendix D. Notably, with almost no prompt engineering our LLM-based system places second on this leaderboard. We argue that for the investigation in this paper, direct comparison to prior non-LLM baselines is not only unfair to those approaches, but not particularly insightful. Therefore, similar to Wei et al. (2022b), we focus on variants of our LLM-based method and treat them as baselines. Example outputs of each template type can be found in Appendix E.

We conduct experiments with the following settings: (1) No Representation – this is the default setting of going directly from the source document to the target document. We skip the intermediate generation step, passing the full paper as input. We experiment both with and without the style parameters. (2) Own Representation – we do not pass a JSON structure to the intermediate generation step, and allow the model to choose its own structure. (3) Text Representation – we extract the text from the intermediate representation, discarding the JSON structure. (4) JSON Representation – this is the full JSON structure for the intermediate generation step. We experiment both with and without the style parameters.

We use `gpt35-16k` in our main set of experiments. We truncate text that is too long for the input window and use a temperature of 0.0 as standard.[5]

[5]A detailed evaluation of the temperature hyper-parameter is included in Appendix §C

			Similarity Measure			
	Rep.	Style	R-L	M	B	BERTS
Slides	None	×	5.0	6.4	0.3	31.6
	None	✓	5.1	6.0	0.4	31.7
	Own	✓	6.5	7.1	1.2	36.1
	Text	✓	7.3	8.0	1.4	36.4
	JSON	×	4.2	6.0	0.3	31.4
	JSON	✓	**7.4**	**8.4**	**1.5**	**36.9**
Blogs	None	×	26.6	19.6	3.0	82.5
	None	✓	25.1	17.7	2.3	**82.8**
	Own	✓	23.9	19.2	2.3	82.2
	Text	✓	25.4	19.3	2.5	82.5
	JSON	×	**28.3**	**25.3**	**5.0**	82.3
	JSON	✓	25.4	19.6	2.8	82.4
Posters	None	×	8.1	10.3	1.0	35.6
	None	✓	10.1	11.6	1.9	39.5
	Own	✓	12.8	12.6	2.9	52.8
	Text	✓	11.3	11.7	2.1	45.9
	JSON	×	14.2	**16.8**	4.0	52.8
	JSON	✓	**15.5**	14.5	**15.3**	**53.3**

Table 1: Evaluation results using GPT3.5 (`gpt35-16k`). For each template, we experiment with different representations (Rep) and whether or not we include the style parameters (Style). We report the TAE F1 scores as calculated in §5, using ROUGE-L (R-L), METEOR (M), BLEU (B), and BERTScore (BERTS) as the similarity metrics.

6.1 Results of automatic evaluation

In Table 1, we report the TAE F1 scores as described in §5, using ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), BLEU (Papineni et al., 2002), and BERTScore (Zhang* et al., 2020) for the similarity measure. As seen in the results, by most measures, generating a JSON intermediate representation yields the best performance.

We see that using the text representation generally degrades the performance over providing the structured JSON representation, indicating that structure is important for downstream performance in addition to abstractive filtering of information. Additionally, the text representation performs better than skipping the intermediate step altogether for both the poster and slide generation task, but not the blog generation task. This is likely because posters and slides have more inherent structure than blog posts, which can be relatively free-form.

Finally, we see that allowing the model to choose its own representation format degrades performance over providing our pre-defined JSON structure. However, we see that in most cases, providing a representation generated without a JSON structure still performs better than skipping the intermediate generation step altogether (while maintaining the same style parameter setting). This indicates that even without a pre-defined structure, the intermediate step is still valuable for performance.

	Model	Rep.	R-L	M	B	BERTS
			Similarity Measure			
Slides	MS	×	0.6	0.4	0.0	28.6
		✓	**4.4**	**4.6**	**0.4**	**30.5**
	MX	×	4.8	7.3	0.5	31.8
		✓	**6.7**	**7.9**	**1.0**	**34.0**
	GPT4	×	8.3	**9.6**	1.7	36.2
		✓	**8.4**	9.1	**2.0**	**38.1**
Blog	MS	×	2.7	1.7	0.1	73.5
		✓	**21.7**	**16.2**	**1.7**	**81.3**
	MX	×	22.8	15.7	2.1	**82.6**
		✓	**25.6**	**20.5**	**3.2**	82.5
	GPT4	×	25.7	19.9	2.6	**82.8**
		✓	**25.8**	**20.2**	**3.1**	82.7
Poster	MS	×	3.2	1.8	0.2	32.2
		✓	**6.0**	**6.5**	**1.2**	**38.1**
	MX	×	**10.5**	**11.4**	**1.7**	40.9
		✓	10.4	11.0	1.5	**50.7**
	GPT4	×	**16.4**	**18.2**	**4.5**	**59.8**
		✓	14.6	15.3	3.7	57.2

Table 2: TAE F1 scores using Mistral-7b (MS), Mixtral (MX), and GPT4. We use ROUGE-L (R-L), METEOR (M), BLEU (B) and BERTScore (BERTS) as our similarity measures. For each template, we compare a JSON representation versus skipping the intermediate generation step (Rep), maintaining the same style parameters in both settings.

Do results generalize to other models? We conduct a subset of our experiments on Mistral-7B (Jiang et al., 2023), Mixtral (Jiang et al., 2024), and GPT4 (`gpt4-32k`), comparing the JSON representation to skipping the intermediate step. We maintain the same style parameters in both settings. In Table 2, we can see that by most measures, the documents generated with the intermediate representation score higher than the documents generated without, particularly for blog posts and slides. The difference in performance is larger for Mistral than Mixtral and GPT4, indicating that our method particularly improves the performance of smaller models. Smaller models are generally cheaper, less resource intensive, and faster, but often operate at the cost of performance. The results indicate that for applications that are sensitive to cost or latency, this trade-off can be mitigated with a structured intermediate representation. The only experiment in which the documents generated with the representation do not strictly score higher on most measures is the posters generated with Mixtral and GPT4. Upon closer inspection, the references in this dataset are very verbose, averaging 391 tokens. Our method produces generally less verbose posters, averaging 265 total tokens compared to 345 tokens produced by the baseline. We hypothesize that by editing the style parameters to include information about verbosity and length, we can improve performance on posters in the future.

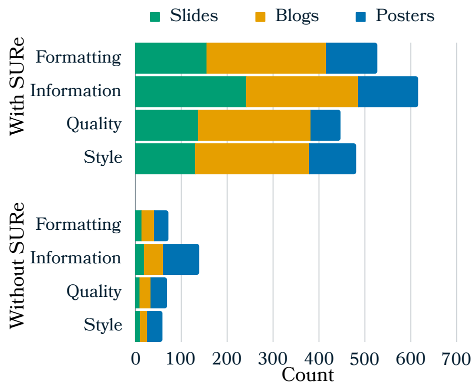


Figure 3: Reasons annotators preferred each document. While annotators largely preferred documents generated with an intermediate representation, the most common reasons for preference are better formatting and information content. We exclude the “Other” count as it was only selected once.

6.2 Human evaluation

After showing that LLMs benefit from intermediate structured representations in document transformations, we investigate whether our proposed evaluation framework aligns better with human judgment than previously proposed metrics. We sample 100 documents each from DOC2PPT and LongSumm, and use the entirety of the Paper-Poster dataset in this study. We present annotators with 2 versions of each document, one generated with the intermediate representation and one without. Both versions use `gpt4-32k`, as the best performing model.

The annotators are provided with the original paper and the intended document type (blog, slide deck, or poster), and are asked the following questions: (1) Which document do you prefer? (2) On a scale of 1-3, to what degree do you prefer your selection? (3) Why do you prefer your selection? For question 3, annotators are also provided with a multi-select checklist of reasons for their preference: (1) quality of the content, (2) formatting, (3) document style matching the intended document types, (4) information represented in the document, and (5) other (along with a free text box). The full instructions, including the reasons provided and examples, can be found in Appendix §B.

If the models do not produce LaTeX and instead produce only text, we wrap the text with `\begin{document}` and `\end{document}`. We force the compilation of the outputs with the command: `pdflatex --interaction=nonstopmode <filename.tex>`. Occasionally, this forced compilation leads to oddly formatted documents, but we consider this to be a part of the performance of the method and present the documents with no further changes. Each document is annotated by 3 different annotators. We employed 4 annotators from India, sourced via a third-party agency, to carry out the human evaluation of our task based on a guideline document containing task-specific instructions, guidance, and annotated examples. They were compensated at a rate of $11.98 USD per hour for the total time spent working on the task, including a training round of annotation.

Which method do humans prefer? The documents generated with an intermediate representation were preferred by 82%, based on majority vote (71% unanimously). The annotator agreement score was 0.51 with Krippendorff's alpha, indicating that while this is a subjective and specialized task, even non-expert annotators agree to a moderate degree. A visualization of the reasons the annotators preferred their selection can be found in Figure 3. It can be seen that while annotators largely preferred the documents generated with an intermediate representation, the most common reasons for preference are better formatting and better information content. This indicates that the structure provided by the intermediate representation makes it easier for the model to format the final document well. Additionally, the intermediate representation only includes the most salient information from the original text, resulting in higher quality of information content. Finally, we see a fairly even distribution across different templatic views for the reasons of preference, indicating humans prefer the documents generated with the intermediate representation across different document types.

Which metric correlates better with humans? We test whether our metric, as described in §5, correlates better with human preference compared to prior evaluation metrics in the literature. For each annotation, given the degree of the preference d (Appendix §B Q. 2) we convert value to a score $P(d) \rightarrow [1, 2, 3]$ if d is slight, moderate, or strong, respectively. If the annotator prefers the document generated without an intermediate representation, we take $-P(d)$ instead. This allows us to measure if the metric captures directionality of preference along with degree. in parallel, we compute the automatic score m for each metric, then calculate $S = m(\text{with rep}) - m(\text{skip rep})$ where m is the metric we are evaluating (e.g ROUGE). If a human annotator prefers a document generated without the intermediate step, we'd expect a good metric to assign a higher score to that document as well, resulting in both S and $P(d)$ being negative (and positive in the opposite case). Using this intuition we assign an affinity score of a metric with respect to human evaluation as the Pearson correlation (Freedman et al., 2007) of S and $P(d)$.

Since prior metrics are not designed to account for the structure of documents, we compute them by extracting only the text of both the generated and reference documents. The correlations with human judgement for each metric to its respective TAE variants can be found in Table 3. As we can see from the results, evaluations using our template adaptable framework correlate more highly with human judgement, except in the case of BERTScore. In the latter case the results are not statistically significant, and we hypothesize that the open-domain nature of BERT embeddings are poorly suited to represent the semantic similarity of scientific text.

Metric	PearsonR
ROUGE-L	14.5
TAE ROUGE-L	**19.7**
METEOR	24.6
TAE METEOR	**25.2**
BLEU	13.6
TAE BLEU	**13.8**
BERTScore	**10.6***
TAE BERTScore	5.4*

Table 3: Correlation of evaluation metrics with human judgement. We compare each metric computed using the TAE framework versus the standard computation. *Indicates the correlation is not statistically significant ($p > 0.01$).

7 Conclusion

In many domains, people choose to disseminate information across different modalities and formats for better communication to broader audiences. We proposed a unified view of document transformation and evaluation. We showed that LLMs are capable of templatic document generation with minimal supervision, and that a structured, intermediate representation can improve performance, particularly for smaller models. We also introduced a flexible precision-recall framework for automatic evaluation that easily integrates existing evaluation metrics into a unified system and allows for comparison across diverse datasets without additional task specific metric design. Finally, we conducted a human evaluation and showed that annotators prefer the documents generated using the intermediate representation 82% of the time and that our evaluation framework correlates better with human preference than standard evaluation metrics.

8 Limitations

Although our methods are not domain specific, we only evaluated them in the scientific domain, due to the availability of public data. Additionally, our framework is limited to textual content. In future work we plan to explore the application of our unified framework for generation and evaluation on document views in other domains, as well as incorporating multi-modal models and content generation. Finally, it is possible that some of our test data has leaked into the training data of the models with which we experimented. This limitation is not unique to our work and exists for our baselines in addition to our methods.

9 Ethics

The potential risks of our work are similar to those of other work in downstream applications of LLMs. LLM generated documents can potential generate copy-righted material (Carlini et al., 2020), personally-identifiable information (Lukas et al., 2023), or factually incorrect text (Manakul et al., 2023). The use of LLMs to generate documents may violate some academic dishonesty policies (Zdravkova et al., 2023). Our system is intended to be used in collaboration with human writers. Users should edit the generations, checking for factual inconsistencies and other potential errors. Our work is intended to save users time that might be spent repeating information across multiple documents, so they can focus on content creation. Therefore, we believe the benefits of our work outweigh the potential risks.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

Lutz Bornmann and Rüdiger Mutz. 2014. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. In *USENIX Security Symposium*.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *Conference on Empirical Methods in Natural Language Processing*.

Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2023. How ready are pre-trained abstractive models and llms for legal case judgement summarization? *ArXiv*, abs/2306.01248.

Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. 2023. Qlarify: Bridging scholarly abstracts and papers with recursively expandable summaries. *ArXiv*, abs/2310.07581.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Tsu-Jui Fu, William Yang Wang, Daniel J. McDuff, and Yale Song. 2021. Doc2ppt: Automatic presentation slides generation from scientific documents. In *AAAI Conference on Artificial Intelligence*.

Yue Hu and Xiaojun Wan. 2015. Ppsgen: Learning-based presentation slides generation for academic papers. *IEEE Transactions on Knowledge and Data Engineering*, 27:1085–1097.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L'elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2125–2131, Florence, Italy. Association for Computational Linguistics.

Da-Wei Li, Danqing Huang, Tingting Ma, and Chin-Yew Lin. 2021. Towards topic-aware slide generation for academic papers with unsupervised mutual learning. In *AAAI Conference on Artificial Intelligence*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Nils Lukas, A. Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-B'eguelin. 2023. Analyzing leakage of personally identifiable information in language models. *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363.

Potsawee Manakul, Adian Liusie, and Mark John Francis Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv*, abs/2303.08896.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Yuting Qiang, Yanwei Fu, Yanwen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2016. Learning to generate posters of scientific papers. *ArXiv*, abs/1604.01219.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alessandro Sciré, Simone Conia, Simone Ciciliano, and Roberto Navigli. 2023. Echoes from alexandria: A large resource for multilingual book summarization. In *Annual Meeting of the Association for Computational Linguistics*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Athar Sefid and C. Lee Giles. 2022. Scibertsum: Extractive summarization for scientific documents. In *Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, May 22–25, 2022, Proceedings*, page 688–701, Berlin, Heidelberg. Springer-Verlag.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. 2021. D2S: Document-to-slide generation via query-based text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1418, Online. Association for Computational Linguistics.

Eulalia Szmidt and Janusz Kacprzyk. 2010. The spearman rank correlation coefficient between intuitionistic fuzzy sets. In *2010 5th IEEE International Conference Intelligent Systems*, pages 276–280.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Fengjie Wang, Xuye Liu, Oujing Liu, Ali Neshati, Tengfei Ma, Min Zhu, and J. Zhao. 2023a. Slide4n: Creating presentation slides from computational notebooks with human-ai collaboration. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yuanyuan Wang, Yukiko Kawai, and Kazutoshi Sumiya. 2015. iposter: Interactive poster generation based on topic structure and slide presentation. *Transactions of The Japanese Society for Artificial Intelligence*, 30:112–123.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H.

Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Sheng Xu and Xiaojun Wan. 2021. Neural content extraction for poster generation of scientific papers. *ArXiv*, abs/2112.08550.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Katerina Zdravkova, Fisnik Dalipi, and Fredrik Ahlgren. 2023. Integration of large language models into higher education: A perspective from learners. *2023 International Symposium on Computers in Education (SIIE)*, pages 1–6.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Conference on Empirical Methods in Natural Language Processing*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Peng Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Conference on Empirical Methods in Natural Language Processing*.

A Prompt details

The prompts and intermediate representation template used can be found in Table 4 and Figure 4, respectively. We note that the specific structure provided to the prompt is not inherent to our method, and a different structure could be provided depending on the input document and domain. For the tasks we evaluate in this paper, we use the following style parameters:

- **Slides:** "Slides should include a title page. Following slides should contain an informative slide title and short, concise bullet points. Longer slides should be broken up into multiple slides."
- **Posters:** "Posters should include a title section at the top. Each panel should include a heading and short, concise bullet points of the most important take-aways from that section."
- **Blogs:** "Blogs should include paragraphs introducing the topic, a summary of the input document, and important takeaways. The blog should be more readable to a general audience than the input document."

B Annotation Instructions

B.1 Questions

Question 1 – Which document do you prefer?
In this question, you are asked to choose which document version you prefer. Some examples of qualities you may use to decide your preference include:

- The quality of the content – The text is grammatical and understandable. E.g. Document A contains major grammatical errors while Document B only contains minor errors.
- The formatting – The formatting is reasonable and matches the formatting of the intended document type. E.g. A poster contains panels and each panel contains a header and body text.
- The style – The document matches the style of the intended document type. E.g. Shorter, bulleted sentences in a slide deck.
- Information represented in the document – The document contains sufficient information to represent the input document. E.g. A blog post represents the most important sections from the input document.

```
{
“Document Title": "TITLE",
"Document Authors: [
                    "AUTHOR 1",
                    "AUTHOR2",
                    ...
                    "AUTHOR N"
                   ],
"SECTION TITLE 1": {
                    "Content": [
                                "SENTENCE 1",
                                "SENTENCE 2",
                                ...
                                "SENTENCE N"
                                ]
                    },
"SECTION TITLE 2": {
                    "Content": [
                                "SENTENCE 1",
                                "SENTENCE 2",
                                ...
                                "SENTENCE N"
                                ]
                    },
...
"SECTION TITLE N": {
                    "Content": [
                                "SENTENCE 1",
                                "SENTENCE 2",
                                ...
                                "SENTENCE N"
                                ]
                    }
}
```

Figure 4: Template of the intermediate representation provided to the prompts in Table 4.

The above criteria are non-exhaustive. Not all criteria must be met, and you may use other relevant criteria to make your decision. You are not rating the document for factual correctness,[6] and only need to refer to the corresponding scientific article if it will aid in making your preference. You can answer this question with either Document A or Document B.

Question 2 – On a scale of 1-3, to what degree do you prefer your selection?
In this question you will rate the degree to which you prefer your selection, on the following scale:

1. Small preference – The documents are similar in quality and only contain minor differences that affect my preference.

[6]The annotators are non-experts and do not have the background to determine factual correctness of scientific information. Instead, they are encouraged to use the original paper to understand if the information presented in the documents represent the information in the paper, to the best of their understanding.

Prompt Function	Prompt
Generate the intermediate representation	"Given the input text, extract the document title and authors. For each section in the given input text, extract the most important sentences. Format the output using the following JSON template:\n <SURe STRUCTURE>\n\n Input: <INPUT DOCUMENT>\n Output:"
Generate LaTeX document	"Summarize the following input in a <TEMPLATE TYPE>style. Style parameters: <STYLE PARAMETERS> Format the output document as a latex file:\n Input: <INPUT DOCUMENT>\n\n Output:"

Table 4: Prompts used to generate the intermediate representation and final LaTeX document. The JSON structure is pictured in Figure 4.

2. Moderate preference – I clearly prefer one document but the differences are not major.

3. Strong preference – I have a strong preference for one document and the differences between the documents are major.

Question 3 – Why do you prefer your selection? (You may select more than one property)

☐ Formatting

☐ Information

☐ Quality

☐ Style

☐ Other (free text)

		Simularity Measure			
	Temp	R-L	M	B	BERTS
Slides	0.0	7.3	8.2	**1.6**	35.1
	0.25	7.2	**8.3**	1.5	35.4
	0.5	7.0	**8.3**	1.5	**36.4**
	0.75	7.0	8.0	1.2	35.5
	1.0	**7.4**	8.2	1.4	35.8
Blogs	0.0	25.3	19.9	2.7	82.7
	0.25	25.5	19.8	2.7	82.6
	0.5	**26.2**	**20.9**	**3.2**	**82.8**
	0.75	25.4	19.9	2.7	82.6
	1.0	24.8	19.9	2.6	82.7
Posters	0.0	**13.5**	**15.3**	**3.4**	**53.5**
	0.25	13.0	14.8	**3.4**	53.1
	0.5	12.5	14.0	2.7	52.2
	0.75	12.0	13.9	3.0	50.8
	1.0	11.6	11.9	2.4	50.3

Table 5: Results of the temperature hyperparameter experiments. We use ROUGE-L (R-L), METEOR (M), BLEU (B) and BERTScore (BERTS) as our similarity measures.

B.2 Edge cases

For most edge cases, it is up to your discretion on how to best handle the case. However, below are a few examples of how you could consider certain edge cases:

Example 1: Slides 1-5 of Document A are higher quality but slides 6-10 of Document B are higher quality. You could reason that the first slides represent the most important information, and choose Document A. However, since Document B contained higher quality slides for another portion of the document, you could rate your degree of preference as "Small preference."

Example 2: Document A more closely matches the style of the intended document type, but Document B contains more relevant information to the source document. You could consider if Document A contains sufficient information to represent the input document, such as representing the most important sections. If yes, then you could prefer Document A. If not, then you could reason that information content is more important than style, and prefer Document B.

Example 3: Document A contains more relevant information than Document B, but also contains major formatting errors, such as text being cut off from the document.

You could reason that although Document A contains more relevant information, the major formatting errors are significant enough to prefer Document B.

Example 4: Neither document matches the style or formatting of the intended document type. Since neither document matches the style or formatting of the intended document type, you could consider other criteria, such as quality of content or the information represented.

C Temperature Experiments

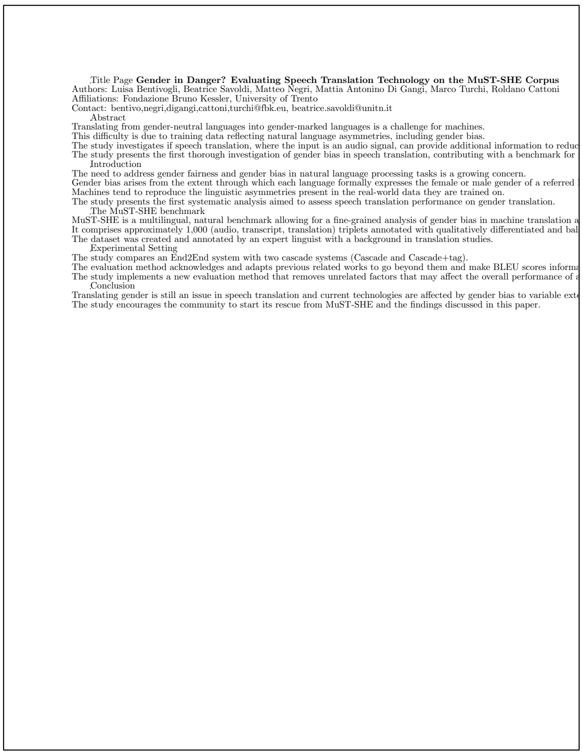
We experiment with the temperature of the generations to see how temperature affects performance. We randomly sample 100 documents each from LongSumm and Doc2PPT for the blog and poster generation tasks, respectively. We use the entirety of the Paper-Poster dataset, since it contains less than 100 examples. We use `gpt35-16k` and experiment with the temperatures $[0.0, 0.25, 0.5, 0.75, 1.0]$. The results of this experiment can be found in Table 5. As we can see from the results, there seems to be little consistency across the different types in which temperature performs the best.

D Longsumm Blind Test Set Results

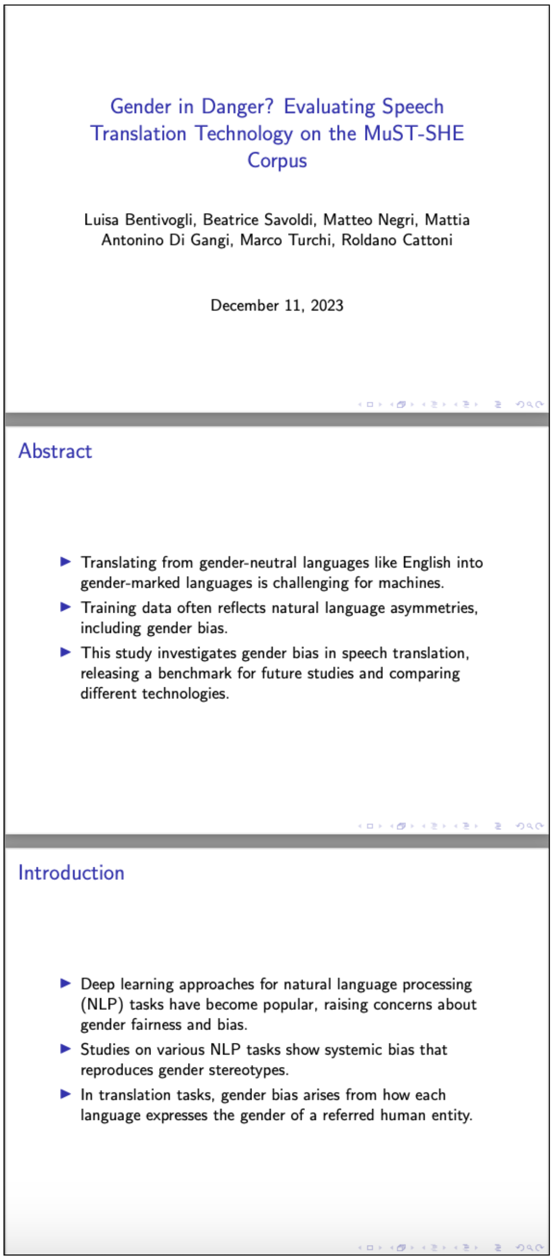
We submit the final documents from GPT4, the best performing model overall, to the Longsumm blind test set evaluation. We compare the documents generated with and without the intermediate step. We see that without the intermediate representation we get a Rouge-1 score of 46.8 while the results generated without the intermediate representation received a Rouge-1 score of 46.4. We note that this blind test set of 22 papers is significantly smaller than the evaluation data (505 papers) we used in the main body of this paper. Despite not designing a task specific method, we place second on the leaderboard, showing the powerful capabilities of LLMs in long document generation.

E Example Outputs

We provide examples of the outputs generated with and without the intermediate representation below. The documents in all examples are generated with GPT4 (`gpt4-32k`). Figure 5 includes example slide generations, Figure 6 includes example blog generations, and Figure 7 includes example poster generations.

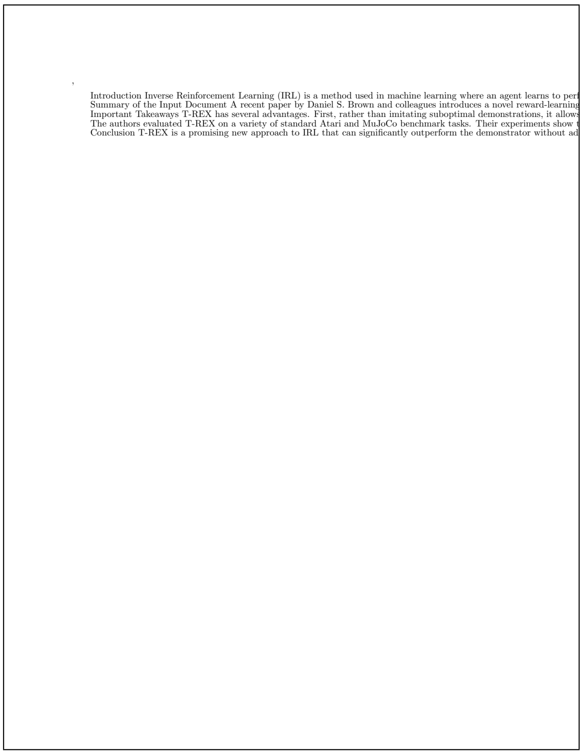


(a) Document generated without intermediate representation. This example is not cropped.



(b) Document generated with intermediate representation. This example is cropped for space and includes an additional 4 slides that are not included for space.

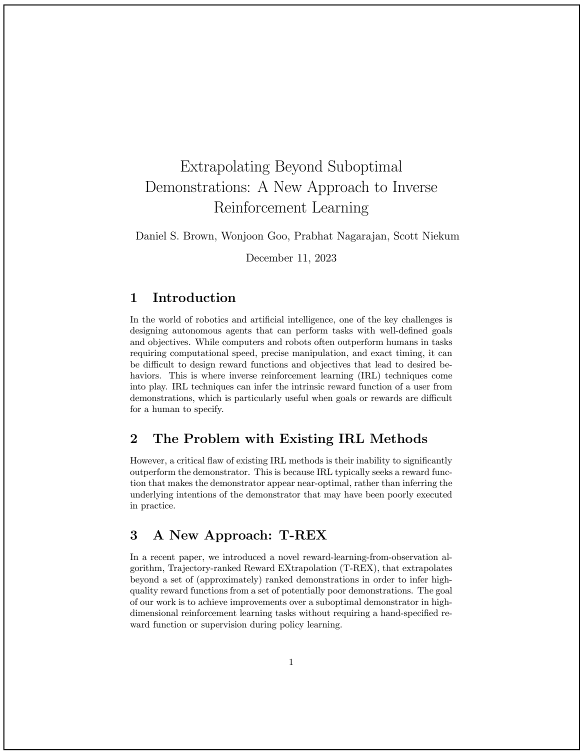
Figure 5: The above documents are example slides generated by GPT4 (`gpt4-32k`) with and without the intermediate representation. We can see that without the intermediate step, the model did not generate a true slide deck.



,

Introduction Inverse Reinforcement Learning (IRL) is a method used in machine learning where an agent learns to perf
Summary of the Input Document A recent paper by Daniel S. Brown and colleagues introduces a novel reward-learning
Important Takeaways T-REX has several advantages. First, rather than imitating suboptimal demonstrations, it allows
The authors evaluated T-REX on a variety of standard Atari and MuJoCo benchmark tasks. Their experiments show t
Conclusion T-REX is a promising new approach to IRL that can significantly outperform the demonstrator without ad

(a) Document generated without the intermediate representation. This example is not cropped.



Extrapolating Beyond Suboptimal Demonstrations: A New Approach to Inverse Reinforcement Learning

Daniel S. Brown, Wonjoon Goo, Prabhat Nagarajan, Scott Niekum

December 11, 2023

1 Introduction

In the world of robotics and artificial intelligence, one of the key challenges is designing autonomous agents that can perform tasks with well-defined goals and objectives. While computers and robots often outperform humans in tasks requiring computational speed, precise manipulation, and exact timing, it can be difficult to design reward functions and objectives that lead to desired behaviors. This is where inverse reinforcement learning (IRL) techniques come into play. IRL techniques can infer the intrinsic reward function of a user from demonstrations, which is particularly useful when goals or rewards are difficult for a human to specify.

2 The Problem with Existing IRL Methods

However, a critical flaw of existing IRL methods is their inability to significantly outperform the demonstrator. This is because IRL typically seeks a reward function that makes the demonstrator appear near-optimal, rather than inferring the underlying intentions of the demonstrator that may have been poorly executed in practice.

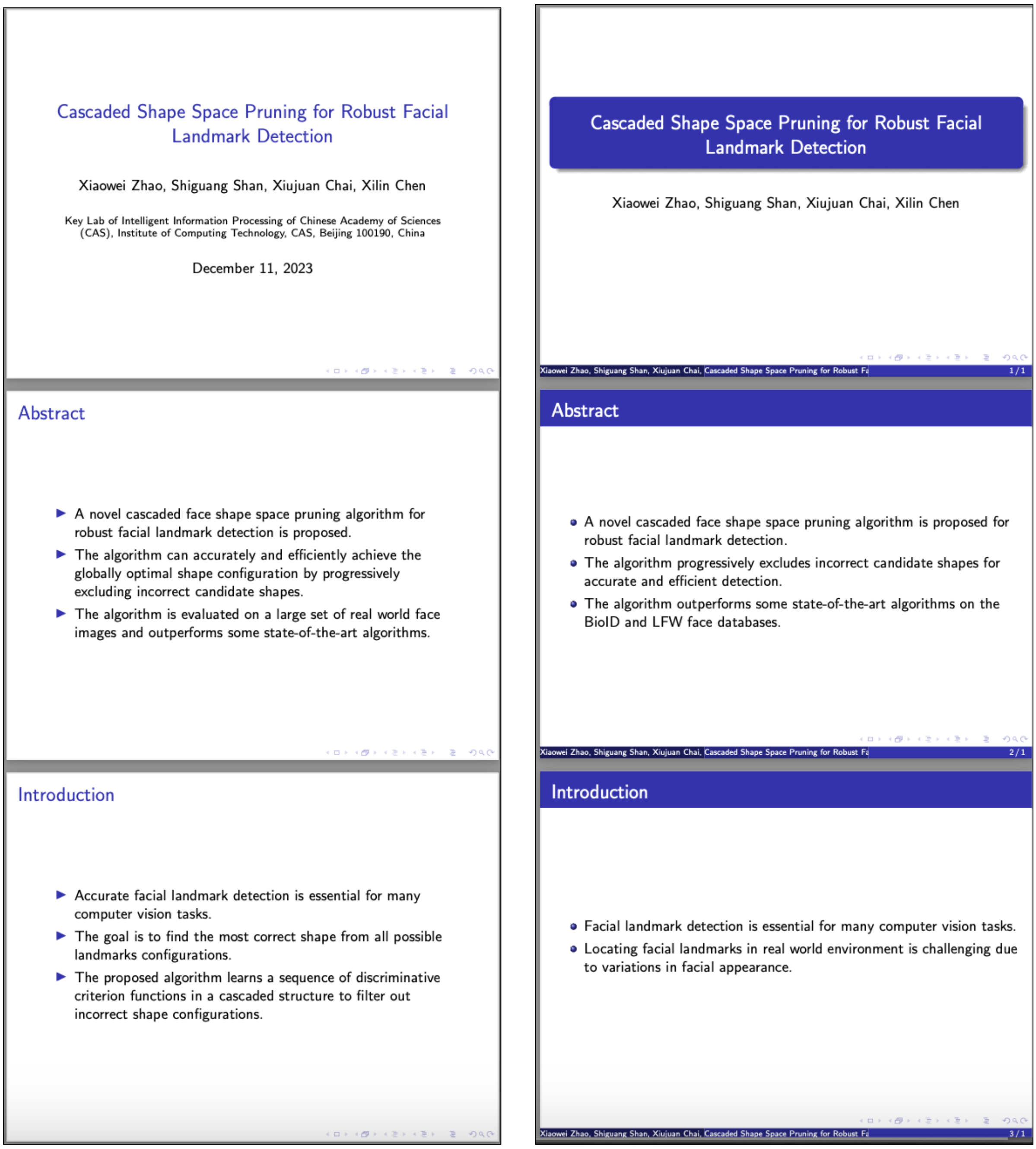
3 A New Approach: T-REX

In a recent paper, we introduced a novel reward-learning-from-observation algorithm, Trajectory-ranked Reward EXtrapolation (T-REX), that extrapolates beyond a set of (approximately) ranked demonstrations in order to infer high-quality reward functions from a set of potentially poor demonstrations. The goal of our work is to achieve improvements over a suboptimal demonstrator in high-dimensional reinforcement learning tasks without requiring a hand-specified reward function or supervision during policy learning.

1

(b) Document generated with the intermediate representation. This example is cropped for space and includes an additional page of text.

Figure 6: The above documents are example blog posts generated by GPT4 (`gpt4-32k`) with and without the intermediate representation. We can see that without the intermediate representation, the model did not properly format the LaTeX file for compilation.



(a) Document generated without the intermediate representation. This example is cropped for space and includes an additional 3 slides.

(b) Document generated with the intermediate representation. This example is cropped for space and includes an additional 4 slides.

Figure 7: The above documents are example posters generated by GPT4 (`gpt4-32k`) with and without the intermediate representation. We found that GPT4 often generates slide decks in place of posters. We can see that the document generated without the intermediate representation contains more verbose panels and includes less formatting.